

Springer Tracts in Advanced Robotics

Volume 25

Editors: Bruno Siciliano · Oussama Khatib · Frans Groen

Springer Tracts in Advanced Robotics

Edited by B. Siciliano, O. Khatib, and F. Groen

- Vol. 24:** Yuta, S.; Asama, H.; Thrun, S.; Prassler, E.; Tsubouchi, T. (Eds.)
Field and Service Robotics
550 p. 2006 [3-540-32801-7]
- Vol. 23:** Andrade-Cetto, J.; Sanfeliu, A.
Environment Learning for Indoor Mobile Robots
130 p. 2006 [3-540-32795-9]
- Vol. 22:** Christensen, H.I. (Ed.)
European Robotics Symposium 2006
209 p. 2006 [3-540-32688-X]
- Vol. 21:** Ang Jr., H.; Khatib, O. (Eds.)
Experimental Robotics IX
618 p. 2006 [3-540-28816-3]
- Vol. 20:** Xu, Y.; Ou, Y.
Control of Single Wheel Robots
188 p. 2005 [3-540-28184-3]
- Vol. 19:** Lefebvre, T.; Bruyninckx, H.; De Schutter, J.
Nonlinear Kalman Filtering for Force-Controlled Robot Tasks
280 p. 2005 [3-540-28023-5]
- Vol. 18:** Barbagli, F.; Prattichizzo, D.; Salisbury, K. (Eds.)
Multi-point Interaction with Real and Virtual Objects
281 p. 2005 [3-540-26036-6]
- Vol. 17:** Erdmann, M.; Hsu, D.; Overmars, M.; van der Stappen, F.A (Eds.)
Algorithmic Foundations of Robotics VI
472 p. 2005 [3-540-25728-4]
- Vol. 16:** Cuesta, F.; Ollero, A.
Intelligent Mobile Robot Navigation
224 p. 2005 [3-540-23956-1]
- Vol. 15:** Dario, P.; Chatila R. (Eds.)
Robotics Research – The Eleventh International Symposium
595 p. 2005 [3-540-23214-1]
- Vol. 14:** Prassler, E.; Lawitzky, G.; Stopp, A.; Grunwald, G.; Hägele, M.; Dillmann, R.; Iossifidis, I. (Eds.)
Advances in Human-Robot Interaction
414 p. 2005 [3-540-23211-7]
- Vol. 13:** Chung, W.
Nonholonomic Manipulators
115 p. 2004 [3-540-22108-5]
- Vol. 12:** Iagnemma K.; Dubowsky, S.
Mobile Robots in Rough Terrain – Estimation, Motion Planning, and Control with Application to Planetary Rovers
123 p. 2004 [3-540-21968-4]
- Vol. 11:** Kim, J.-H.; Kim, D.-H.; Kim, Y.-J.; Seow, K.-T.
Soccer Robotics
353 p. 2004 [3-540-21859-9]
- Vol. 10:** Siciliano, B.; De Luca, A.; Melchiorri, C.; Casalino, G. (Eds.)
Advances in Control of Articulated and Mobile Robots
259 p. 2004 [3-540-20783-X]
- Vol. 9:** Yamane, K.
Simulating and Generating Motions of Human Figures
176 p. 2004 [3-540-20317-6]
- Vol. 8:** Baeten, J.; De Schutter, J.
Integrated Visual Servoing and Force Control
198 p. 2004 [3-540-40475-9]
- Vol. 7:** Boissonnat, J.-D.; Burdick, J.; Goldberg, K.; Hutchinson, S. (Eds.)
Algorithmic Foundations of Robotics V
577 p. 2004 [3-540-40476-7]
- Vol. 6:** Jarvis, R.A.; Zelinsky, A. (Eds.)
Robotics Research – The Tenth International Symposium
580 p. 2003 [3-540-00550-1]
- Vol. 5:** Siciliano, B.; Dario, P. (Eds.)
Experimental Robotics VIII
685 p. 2003 [3-540-00305-3]
- Vol. 4:** Bicchi, A.; Christensen, H.I.; Prattichizzo, D. (Eds.)
Control Problems in Robotics
296 p. 2003 [3-540-00251-0]
- Vol. 3:** Natale, C.
Interaction Control of Robot Manipulators – Six-degrees-of-freedom Tasks
120 p. 2003 [3-540-00159-X]
- Vol. 2:** Antonelli, G.
Underwater Robots – Motion and Force Control of Vehicle-Manipulator Systems
209 p. 2003 [3-540-00054-2]
- Vol. 1:** Caccavale, F.; Villani, L. (Eds.)
Fault Diagnosis and Fault Tolerance for Mechatronic Systems – Recent Advances
191 p. 2002 [3-540-44159-X]

Peter Corke · Salah Sukkarieh (Eds.)

Field and Service Robotics

Results of the 5th International Conference

With 395 Figures

Professor Bruno Siciliano, Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Napoli, Italy, email: siciliano@unina.it

Professor Oussama Khatib, Robotics Laboratory, Department of Computer Science, Stanford University, Stanford, CA 94305-9010, USA, email: khatib@cs.stanford.edu

Professor Frans Groen, Department of Computer Science, Universiteit van Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands, email: groen@science.uva.nl

Editors

Dr. Peter Corke
CSIRO ICT Centre
Autonomous Systems Laboratory
P.O. Box 883
4069 Kenmore
Australia
peter.corke@csiro.au

Dr. Salah Sukkariah
Australian Centre for Field Robotics
The Rose Street Building J04
The University of Sydney, NSW
Australia
salah@acfr.usyd.edu.au

ISSN print edition: 1610-7438

ISSN electronic edition: 1610-742X

ISBN-10 3-540-33452-1 **Springer Berlin Heidelberg New York**

ISBN-13 978-3-540-33452-1 **Springer Berlin Heidelberg New York**

Library of Congress Control Number: 2006923558

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in other ways, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2006

Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Digital data supplied by editors.

Data-conversion and production: PTP-Berlin Protago- \TeX -Production GmbH, Germany (www.ptp-berlin.com)

Cover-Design: design & production GmbH, Heidelberg

Printed on acid-free paper 89/3141/Yu - 5 4 3 2 1 0

Editorial Advisory Board

EUROPE

Herman Bruyninckx, KU Leuven, Belgium
Raja Chatila, LAAS, France
Henrik Christensen, KTH, Sweden
Paolo Dario, Scuola Superiore Sant'Anna Pisa, Italy
Rüdiger Dillmann, Universität Karlsruhe, Germany

AMERICA

Ken Goldberg, UC Berkeley, USA
John Hollerbach, University of Utah, USA
Lydia Kavraki, Rice University, USA
Tim Salcudean, University of British Columbia, Canada
Sebastian Thrun, Stanford University, USA

ASIA/OCEANIA

Peter Corke, CSIRO, Australia
Makoto Kaneko, Hiroshima University, Japan
Sukhan Lee, Sungkyunkwan University, Korea
Yangsheng Xu, Chinese University of Hong Kong, PRC
Shin'ichi Yuta, Tsukuba University, Japan

*This book is dedicated to the memory of
Dr Michael N. Wendt (1968 – 2005)
who was an enthusiastic presenter at FSR2005*

Foreword

At the dawn of the new millennium, robotics is undergoing a major transformation in scope and dimension. From a largely dominant industrial focus, robotics is rapidly expanding into the challenges of unstructured environments. Interacting with, assisting, serving, and exploring with humans, the emerging robots will increasingly touch people and their lives.

The goal of the new series of Springer Tracts in Advanced Robotics (STAR) is to bring, in a timely fashion, the latest advances and developments in robotics on the basis of their significance and quality. It is our hope that the wider dissemination of research developments will stimulate more exchanges and collaborations among the research community and contribute to further advancement of this rapidly growing field.

Since its inception in 1996, FSR, the International Conference on Field and Service Robotics has published archival volumes of high reference value. Starting its past edition in 2003, FSR has found its natural home within STAR, together with other thematic symposia devoted to excellence in robotics research.

The fifth edition of Field and Service Robotics edited by Peter Corke and Salah Sukkarieh offers in its eleven-chapter volume a collection of a broad range of topics in advanced robotics. The contents of these contributions represent a cross-section of the current state of robotics research from one particular aspect: field and service applications, and how they reflect on the theoretical basis of subsequent developments. Pursuing technologies aimed at realizing robots operating in complex and dynamic environments is the big challenge running throughout this focused collection.

Rich by topics and authoritative contributors, FSR culminates with this unique reference on the current developments and new directions in field and service robotics. A fine addition to the series!

Naples, Italy
February 2006

Bruno Siciliano
STAR Editor

Preface

FSR2005

The goal of FSR is to report and encourage the development of field and service robotics. These are non-factory robots, typically mobile, that must operate in complex, and dynamic environments. Typical field robotic applications include mining, agriculture, building and construction, forestry, cargo handling and so on. Field robots may operate on the ground (of Earth or planets), under the ground, underwater, in the air or in space. Service robots are those that work closely with humans, importantly the elderly and sick, to help them with their lives.

The first meeting was held in Canberra, Australia, in 1997. Since then the meeting has been held every 2 years in the pattern Asia, America, Europe. This meeting marks the beginning of the second cycle.

The meeting is vibrant and in good health. This year we had 75 submissions from which we accepted 42 for oral and 11 for poster presentation. All papers were reviewed by three people drawn from the reviewer pool:

| | |
|--------------------|---------------------|
| Hajime Asama | John Bares |
| Raja Chatila | Howie Choset |
| Peter Corke | Matthew Dunbabin |
| Aarne Juhani Halme | John Hollerbach |
| Oussama Khatib | Jong-Hyuk Kim |
| Simon Lacroix | Christian Laugier |
| John Leonard | Eduardo Nebot |
| Cedric Pradalier | Erwin Alex Prassler |
| Jonathan Roberts | Miguel A. Salichs |
| Steve Scheding | Roland Siegwart |
| Sanjiv Singh | Salah Sukkarieh |
| Ashley Tews | Christel-loic Tisse |
| Charles E. Thorpe | Sebastian Thrun |
| Takashi Tsubouchi | Ben Upcroft |

| | |
|-----------------|-------------------|
| Kane Usher | David Wettergreen |
| Stefan Williams | Kazuya Yoshida |
| Alex Zelinsky | |

This meeting has been sponsored and organized jointly by the CSIRO ICT Centre and the Australian Centre for Field Robotics. The Australian Robotics and Automation Association (ARAA) has also sponsored the meeting and supported the registration process. Finally, and most importantly, we would like to thank the local arrangements people who have put together a great technical and social agenda: Matthew Dunbabin, Cedric Pradalier, Elise Koenderman, Jonathan Roberts, and Salah Sukkarieh.

Permanent Program Committee

Hajime Asama
John Bares
Raja Chatila
Howie Choset
Peter Corke
Aarne Juhani Halme
John Hollerbach
Oussama Khatib
Simon Lacroix
Christian Laugier
John Leonard
Erwin Alex Prassler
Jonathan Roberts
Miguel A. Salichs
Roland Siegwart
Sanjiv Singh
Charles E Thorpe
Sebastian Thrun
Takashi Tsubouchi
David Wettergreen
Kazuya Yoshida
Alex Zelinsky

Peter Corke (CSIRO)
Salah Sukkarieh (ACFR)
General Chairs

Table of Contents

Part 1 – Keynotes

| | |
|--|---|
| Container Port Automation | 3 |
| <i>Graeme Nelmes</i> | |
| The Berkeley Lower Extremity Exoskeleton | 9 |
| <i>H. Kazerooni</i> | |

Part 2 – Outdoor Vision

| | |
|--|----|
| Autonomous Helicopter Tracking and Localization Using a Self-Surveying Camera Array | 19 |
| <i>Masayoshi Matsuoka, Alan Chen, Surya P.N. Singh, Adam Coates, Andrew Y. Ng, Sebastian Thrun</i> | |
| Visual Motion Estimation for an Autonomous Underwater Reef Monitoring Robot | 31 |
| <i>Matthew Dunbabin, Kane Usher, Peter Corke</i> | |
| Road Obstacle Detection Using Robust Model Fitting | 43 |
| <i>Niloofer Gheissari, Nick Barnes</i> | |
| Real-Time Regular Polygonal Sign Detection | 55 |
| <i>Nick Barnes, Gareth Loy</i> | |
| Distinctness Analysis on Natural Landmark Descriptors | 67 |
| <i>Kai-Ming Kiang, Richard Willgoss, Alan Blair</i> | |
| Bimodal Active Stereo Vision | 79 |
| <i>Andrew Dankers, Nick Barnes, Alex Zelinsky</i> | |

Part 3 – Navigation

| | |
|---|-----|
| A System for Automatic Marking of Floors in Very Large Spaces | 93 |
| <i>Patric Jensfelt, Gunnar Gullstrand, Erik Förell</i> | |
| Development of an Angular Characterization System for Cooperative UAV/UGV Applications | 105 |
| <i>Paul Thompson, Salah Sukkarieh</i> | |
| Topological Global Localization for Subterranean Voids | 117 |
| <i>David Silver, Joseph Carsten, Scott Thayer</i> | |
| A Navigation System for Automated Loaders in Underground Mines .. | 129 |
| <i>Johan Larsson, Mathias Broxvall, Alessandro Saffiotti</i> | |

Part 4 – Mapping and Localization

Outdoor Simultaneous Localisation and Mapping Using RatSLAM 143
David Prasser, Michael Milford, Gordon Wyeth

Implementation Issues and Experimental Evaluation of D-SLAM 155
Zhan Wang, Shoudong Huang, Gaminu Dissanayake

Scan-SLAM: Combining EKF-SLAM and Scan Correlation 167
Juan Nieto, Tim Bailey, Eduardo Nebot

A Non-rigid Approach to Scan Alignment and Change Detection
 Using Range Sensor Data 179
Ralf Kaestner, Sebastian Thrun, Michael Montemerlo, Matt Whalley

An Efficient Extension of Elevation Maps for Outdoor Terrain Mapping 195
Patrick Pfaff, Wolfram Burgard

Online Reconstruction of Vehicles in a Car Park 207
Christopher Tay Meng Keat, Cédric Pradalier, Christian Laugier

Wavelet Occupancy Grids: A Method for Compact Map Building 219
Manuel Yguel, Olivier Aycard, Christian Laugier

Further Results with Localization and Mapping Using Range
 from Radio 231
Joseph Djugash, Sanjiv Singh, Peter Corke

Experiments with Robots and Sensor Networks for Mapping
 and Navigation 243
Keith Kotay, Ron Peterson, Daniela Rus

Part 5 – Planning

Applying a New Model for Machine Perception and Reasoning
 in Unstructured Environments 257
*Richard Grover, Steve Scheduling, Ross Hennessy, Suresh Kumar,
 Hugh Durrant-Whyte*

Constrained Motion Planning in Discrete State Spaces 269
Mihail Pivtoraiko, Alonzo Kelly

Vision-Based Grasping Points Determination by Multifingered Hands . . 281
Madjid Boudaba, Alicia Casals, Dirk Osswald, Heinz Woern

Embodied Social Interaction for Service Robots
 in Hallway Environments 293
Elena Pacchierotti, Henrik I. Christensen, Patric Jensfelt

| | |
|--|-----|
| Intentional Motion Online Learning and Prediction | 305 |
| <i>Dizan Vasquez, Thierry Fraichard, Olivier Aycard, Christian Laugier</i> | |

Part 6 – Design

| | |
|---|-----|
| Design and Locomotion of a Semi-passive Mobile Platform | 319 |
| <i>Amir Shapiro, Shraga Shoval</i> | |

| | |
|--|-----|
| Wheel Control Based on Body Configuration for Step-Climbing Vehicle | 331 |
| <i>Daisuke Chugo, Kuniaki Kawabata, Hayato Kaetsu, Hajime Asama, Taketoshi Mishima</i> | |

| | |
|---|-----|
| Ball-Shaped Robots: An Historical Overview and Recent Developments at TKK | 343 |
| <i>Jussi Suomela, Tomi Ylikorpi</i> | |

| | |
|--|-----|
| Development of a Water-Hydraulic Self-Propelled Robotic Drill for Underground Mining | 355 |
| <i>Michael N. Wendt, Garry A. Einicke</i> | |

| | |
|---|-----|
| A Wearable GUI for Field Robots | 367 |
| <i>Andreas Hedström, Henrik I. Christensen, Carl Lundberg</i> | |

| | |
|---|-----|
| Design and Implementation of Machine Control Systems with Modern Software Development Tools | 377 |
| <i>Matti Öhman, Arto Visala</i> | |

| | |
|---|-----|
| Long-Term Activities for Autonomous Mobile Robot – Autonomous Insertion of a Plug into Real Electric Outlet by a Mobile Manipulator . | 389 |
| <i>Tomofumi Yamada, Keiji Nagatani, Yutaka Tanaka</i> | |

Part 7 – Telepresence

| | |
|--|-----|
| Synthesized Scene Recollection for Robot Teleoperation | 403 |
| <i>Naoji Shiroma, Hirokazu Nagai, Maki Sugimoto, Masahiko Inami, Fumitoshi Matsuno</i> | |

| | |
|--|-----|
| Development of a Networked Robotic System for Disaster Mitigation – Test Bed Experiments for Remote Operation over Rough Terrain and High Resolution 3D Geometry Acquisition | 415 |
| <i>Kazuya Yoshida, Keiji Nagatani, Kiyoshi Kiyokawa, Yasushi Yagi, Tadashi Adachi, Hiroaki Saitoh, Hiroyuki Tanaka, Hiroyuki Ohno</i> | |

Part 8 – Aerial Robots

| | |
|---|-----|
| Towards Intelligent Miniature Flying Robots | 429 |
| <i>Samir Bouabdallah, Roland Siegwart</i> | |

Design of an Ultra-lightweight Autonomous Solar Airplane
for Continuous Flight 441
André Noth, Walter Engel, Roland Siegwart

Control and Guidance for a Tail-Sitter Unmanned Air Vehicle 453
R. Hugh Stone

The Development of a Real-Time Modular Architecture
for the Control of UAV Teams 465
*David T. Cole, Salah Sukkarieh, Ali Haydar Göktoğan, Hugh Stone,
Rhys Hardwick-Jones*

Part 9 – Off-Road

Trajectory Generation on Rough Terrain Considering Actuator
Dynamics 479
Thomas M. Howard, Alonzo Kelly

Results in Combined Route Traversal and Collision Avoidance 491
Stephan Roth, Bradley Hamner, Sanjiv Singh, Myung Hwangbo

Adaptation to Rough Terrain by Using COF Estimation
on a Quadruped Vehicle 505
Shogo Okamoto, Kaoru Konishi, Kenichi Tokuda, Satoshi Tadokoro

Multi-solution Problem for Track-Terrain Interaction Dynamics
and Lumped Soil Parameter Identification 517
S. Hutangkabodee, Y.H. Zweiri, L.D. Seneviratne, K. Althoefer

3D Position Tracking in Challenging Terrain 529
Pierre Lamon, Roland Siegwart

Efficient Braking Model for Off-Road Mobile Robots 541
Mihail Pivtoraiko, Alonzo Kelly, Peter Rander

Part 10 – Applications

Autonomous Excavation Using a Rope Shovel 555
Matthew Dunbabin, Peter Corke

Automated Inspection System for Large Underground Concrete
Pipes Under Operating Conditions 567
Norbert Elkmann, Bert Reimann, Erik Schulenburg, Heiko Althoff

An Autonomous Weeding Robot for Organic Farming 579
*Tijmen Bakker, Kees van Asselt, Jan Bontsema, Joachim Müller,
Gerrit van Straten*

V Shape Path Generation for Loading Operation by Wheel Loader 591
Shigeru Sarata, Yossewee Weeramhaeng, Akira Horiguchi,
Takashi Tsubouchi

Development of an Autonomous Forest Machine for Path Tracking 603
Thomas Hellström, Thomas Johansson, Ola Ringdahl

Author Index 615

Container Port Automation

Graeme Nelmes

Patrick Cooperation

1 Introduction

Patrick is a focused transport logistics company specialised in the loading and unloading of ships and the efficient land based collection, distribution and storage of cargo for import, export and within Australia.

Patrick began as a one-time small stevedore and now leader in port related transport and logistics. We've evolved into a major transport operator and our background in the deployment and adaptation of technology gives us the understanding of how to apply technology to the efficient movement of freight across the whole supply chain whether it is at the port, on rail, air, and road or at the warehouse.



2 Background

In 1995 Patrick began collaboration with the Australian Centre for Field Robotics (ACFR) at the University of Sydney with the objective of improving waterfront efficiency. An initial project was undertaken to improve quay-crane cycle times through the design of new reeving systems and application of advanced high-speed estimation and control techniques. This was followed in 1996 with a project to design an autonomous straddle carrier. Patrick and ACFR worked with straddle carrier manufacturers



Kalmar Industries to deliver a fleet of new “fly-by-wire” straddle carriers to Patrick terminals in Australia. In 1997, with the help of a START grant, the ACFR undertook a programme to automate one of these straddle carriers. This included the development of control algorithms, navigation and positioning methods, path planning and safety systems. The development was completed in 2000 with the demonstration of a fully autonomous straddle carrier in a restricted area at Port Botany. The system was handed over to a new company, Patrick Technology & Systems (PTS), whose objective was to take this demonstration system through to a production container terminal. The company engaged personnel from both Patrick and the ACFR involved in the development phase, together with new employees skilled in production engineering and terminal operations.

A container terminal at Fisherman’s Island in Brisbane was acquired and redesigned to be automated. Systems engineering, component and software production engineering, port planning and traffic management systems were designed.

Five new automated straddle carriers were produced and the system was put into initial operation in 2002. The automated container terminal was opened for public inspection and to trade media at the General Stevedoring Council Meeting held in Brisbane in April 2002. Since this time the system has been handling ships docking at the terminal and has also undergone further development in readiness for deployment at larger terminals.

3 Operating Principle

The vehicle control system allows for the planning and execution of free-ranging paths over the port area. The autonomous straddle carrier fleet is controlled by the Terminal Operating System which coordinates the motion of all vehicles on the terminal, and which schedules container movement to get best throughput from the terminal. Paths and container movement tasks are generated remotely by the Terminal Operating System. These tasks are interpreted into control actions by the central Traffic Manage-



ment System and then transferred to the straddle carrier's Vehicle Management System. The execution system then tracks execution of these actions relative to the planned path by comparison with navigation system output. A number of methods are employed for evaluating path and control execution integrity during motion.

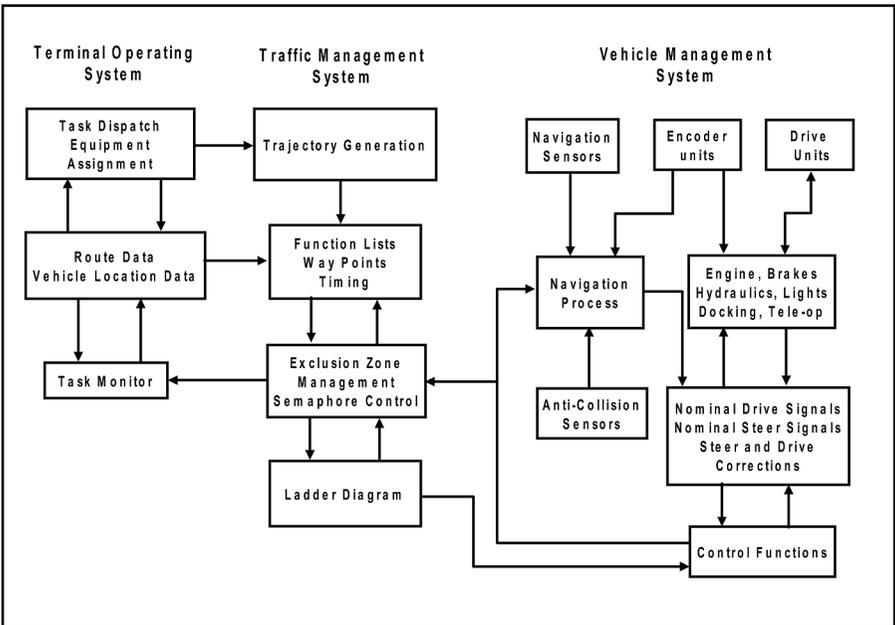
4 System Description

The autonomous straddle carrier system is based on a conventional manned straddle carrier. This is a vehicle capable of picking up, carrying and placing shipping containers, allowing movement of containers from land-side vehicles, to holding yards, to quay cranes and back.

The autonomous (unmanned) straddle carrier is fitted with automatic motion control systems that allow on-board computers to control engine speed, wheel speed, steering angle, and container hoisting.

The navigation system, which determines the position, attitude and velocity of the vehicle, employs a multi-loop multi-sensor position estimation algorithm designed both for high accuracy and also for very high reliability and integrity. The system employs a precision GPS/inertial navigation loop running in parallel with a millimetre-wave radar (MMWR)/encoder loop. Each loop outputs independent navigation information, which can be used to cross check for potential sensor faults and subsequently combined to provide high reliability, high accuracy position estimates.

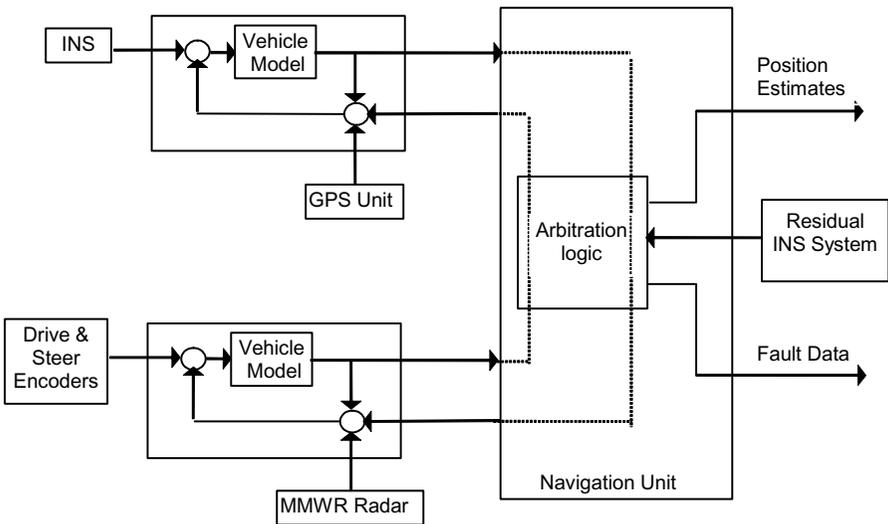
The vehicle control system allows for the planning and execution of free-ranging paths over the port area. Paths and container movement tasks are generated remotely by the port central control system. These tasks are interpreted into control actions by the straddle carrier. The vehicle pilot system also maintains a map of the terminal area to determine which areas



are passable and to implement traffic management systems. Speeds of up to 25Km/h are achieved by the control system.

An independent safety system monitors the integrity and health of various system functions and vehicle components. This is integrated with the existing vehicle PLC controller. A time-of-flight laser system is also used to detect potential collisions in the path of the vehicle. The autonomous straddle carrier fleet is controlled by the terminal management system which coordinates the motion of all vehicles on the terminal, and which schedules container movement to get best throughput from the terminal. The yard area of the container terminal is completely unmanned and secured with physical and electronic barriers to ensure a high degree of safety for the complete system.

The completed system can be operated by a single individual stationed at a control centre on site and the operation of the Fisherman's Island terminal is monitored remotely from PTS offices in Sydney (1000km away).



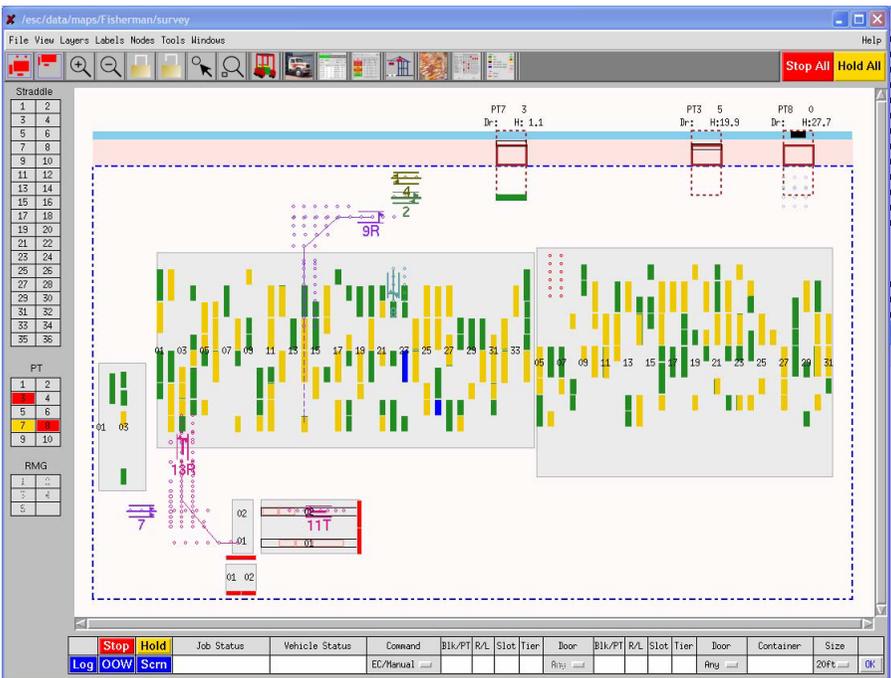
5 Results

The Fisherman's island terminal is currently being exercised with commercial container traffic. Ships docking at the terminal are unloaded by quay-crane and the automated straddle carrier system undertakes all internal container movements including loading on to road-side vehicles. Detailed issues such as dealing with non-standard containers, interfacing to cranes, land-side traffic and the rail network, maintenance of autonomous systems components, are all being dealt with by the team at PTS. The Patrick Fisherman's Island terminal is now one of the most technically

advanced container terminals in the world and demonstrates for the first time the possibility of operating fully automated water-front operations in small to medium sized terminals.

The transition from a university development project into a commercially operational system was achieved by PTS through innovation in systems engineering, design and operational methods. The complexity of the system, including use of advanced control algorithms, robotic methods in navigation and multi-vehicle cooperation, the use of radar and other sensor technologies, makes this project one of the most advanced commercial robotic systems ever deployed.

The autonomous straddle carrier system will revolutionise waterfront operations both in Australia and overseas. Operationally, the end users have been very receptive to the opportunities for improved efficiency and planning offered, and have been surprised and pleased by the huge improvements in maintenance and terminal management made possible.



The Berkeley Lower Extremity Exoskeleton

H. Kazerooni

University of California, Berkeley
Berkeley, CA 94720, USA
exo@me.berkeley.edu

Abstract. The first functional load-carrying and energetically autonomous exoskeleton was demonstrated at U.C. Berkeley, walking at the average speed of 1.3 m/s while carrying a 34 kg (75 lb) payload. Four fundamental technologies associated with the Berkeley Lower Extremity Exoskeleton (BLEEX) were tackled during the course of this project. These four core technologies include: the design of the exoskeleton architecture, control schemes, a body local area network (bLAN) to host the control algorithm and an on-board power unit to power the actuators, sensors and the computers. This article gives an overview of one of the control schemes. The analysis here is an extension of the classical definition of the sensitivity function of a system: the ability of a system to reject disturbances or the measure of system robustness. The control algorithm developed here increases the closed loop system sensitivity to its wearer's forces and torques without any measurement from the wearer (such as force, position, or electromyogram signal). The control method has little robustness to parameter variations and therefore requires a relatively good dynamic model of the system.

1 Definition

The primary objective of this project at U.C. Berkeley is to develop fundamental technologies associated with the design and control of energetically autonomous lower extremity exoskeletons that augment human strength and endurance during locomotion. The first fieldoperational lower extremity exoskeleton (commonly referred to as BLEEX) is comprised of two powered anthropomorphic legs, a power unit, and a backpack-like frame on which a variety of heavy loads can be mounted. This system provides its pilot (i.e. the wearer) the ability to carry significant loads on his/her back with minimal effort over any type of terrain. BLEEX allows the pilot to comfortably squat, bend, swing from side to side, twist, and walk on ascending and descending slopes, while also offering the ability to

step over and under obstructions while carrying equipment and supplies. Because the pilot can carry significant loads for extended periods of time without reducing for extended periods of time without reducing his/her agility, physical effectiveness increases significantly with the aid of this class of lower extremity exoskeletons. In order to address issues of field robustness and reliability, BLEEX is designed such that, in the case of power loss (e.g. from fuel exhaustion), the exoskeleton legs can be easily removed and the remainder of the device can be carried like a standard backpack.



Fig. 1. Berkeley Lower Extremity Exoskeleton (BLEEX) and pilot Ryan Steger. 1: Load occupies the upper portion of the backpack and around the Power Unit; 2: Rigid connection of the BLEEX spine to the pilot's vest; 3: Power unit and central computer occupies the lower portion of the backpack; 4: Semirigid vest connecting BLEEX to the pilot; 5: One of the hydraulic actuators; 6: Rigid connection of the BLEEX feet to the pilot's boots. More photographs can be found at <http://bleex.me.berkeley.edu>

BLEEX was first unveiled in 2004, at U.C. Berkeley's Human Engineering and Robotics Laboratory (Fig. 1). In this initial model, BLEEX offered a carrying capacity of 34 kg (75 lbs), with weight in excess of that allowance being supported by the pilot. BLEEX's unique design offers an ergonomic, highly maneuverable, mechanically robust, lightweight, and durable outfit to surpass typical human limitations. BLEEX has numerous potential appli-

cations; it can provide soldiers, disaster relief workers, wildfire fighters, and other emergency personnel the ability to carry heavy loads such as food, rescue equipment, first-aid supplies, communications gear, and weaponry, without the strain typically associated with demanding labor. Unlike unrealistic fantasy-type concepts fueled by movie-makers and science-fiction writers, the lower extremity exoskeleton conceived at Berkeley is a practical, intelligent, load-carrying robotic device. It is our vision that BLEEX will provide a versatile and realizable transport platform for mission-critical equipment.

The effectiveness of the lower extremity exoskeleton stems from the combined benefit of the human intellect provided by the pilot and the strength advantage offered by the exoskeleton; in other words, the human provides an intelligent control system for the exoskeleton while the exoskeleton actuators provide most of the strength necessary for walking. The control algorithm ensures that the exoskeleton moves in concert with the pilot with minimal interaction force between the two. The control scheme needs no direct measurements from the pilot or the human-machine interface (e.g. no force sensors or EMG); instead, the controller estimates, based on measurements from the exoskeleton only, how to move so that the pilot feels very little force. This control scheme, which has never before been applied to any robotic system, is an effective method of generating locomotion when the contact location between the pilot and the exoskeleton is unknown and unpredictable (i.e. the exoskeleton and the pilot are in contact in variety of places). This control method differs from compliance control methods employed for upper extremity exoskeletons, [9], [10], and [12], and haptic systems [11], and [14] because it requires no force sensor between the wearer and the exoskeleton.

The basic principle for the control of BLEEX rests on the notion that the exoskeleton needs to shadow the wearer's voluntary and involuntary movements quickly, and without delay. This requires a high level of sensitivity in response to all forces and torques on the exoskeleton, particularly the forces imposed by the pilot. Addressing this need involves a direct conflict with control science's goal of minimizing system sensitivity in the design of a closed loop feedback system. If fitted with a low sensitivity, the exoskeleton would not move in concert with its wearer. We realize, however, that maximizing system sensitivity to external forces and torques leads to a loss of robustness in the system.

Taking into account this new approach, our goal was to develop a control system for BLEEX with high sensitivity. We were faced with two realistic concerns; the first was that an exoskeleton with high sensitivity to external forces and torques would respond to other external forces not initiated by its pilot. For example, if someone pushed against an exoskeleton that had high sensitivity, the exoskeleton would move just like the way it would move in

response to the forces from its pilot. Although the fact that it does not stabilize its behavior on its own in response to other forces may sound like a serious problem, if it did (e.g. using a gyro), the pilot would receive motion from the exoskeleton unexpectedly and would have to struggle with it to avoid unwanted movement. The key to stabilizing the exoskeleton and preventing it from falling in response to external forces depends on the pilot's ability to move quickly (e.g. Stepp back or sideways) to create a stable situation for himself and the exoskeleton. For this, a very wide control bandwidth is needed so the exoskeleton can respond to both pilot's voluntary and involuntary movements (i.e. reflexes). The second concern is that systems with high sensitivity to external forces and torques are not robust to variations and therefore the precision of the system performance will be proportional to the precision of the exoskeleton dynamic model. Although this is a serious drawback, we have accepted it as unavoidable. Nevertheless, various experimental systems in our laboratory have proved the overall effectiveness of the control method in shadowing the pilot's movement.

2 Brief History

In the early 1960s, the Defense Department expressed interest in the development of a man-amplifier, a powered suit of armor which would augment soldiers lifting and carrying capabilities.

In 1962, the Air Force had the Cornell Aeronautical Laboratory study the feasibility of using a master-slave robotic system as a man-amplifier. In later work, Cornell determined that an exoskeleton, an external structure in the shape of the human body which has far fewer degrees of freedom than a human, could accomplish most desired tasks [19]. From 1960 to 1971, General Electric developed and tested a prototype man-amplifier, a master-slave system called the Hardiman ([2], [3], [4], [17], and [20]). The Hardiman was a set of overlapping exoskeletons worn by a human operator. The outer exoskeleton (the slave) followed the motions of the inner exoskeleton (the master), which followed the motions of the human operator. All these studies found that duplicating all human motions and using master-slave systems were not practical. Additionally, difficulties in human sensing and system complexity kept it from walking.

Several exoskeletons were developed at the University of Belgrade in the 60's and 70's to aid paraplegics [5] and [24]. Although these early devices were limited to predefined motions and had limited success, balancing algorithms developed for them are still used in many bipedal robots. The "RoboKnee" is a powered knee brace that functions in parallel to the wearer's knee and transfers load to the wearer's ankle (not to the ground) [22]. "HAL" is an

orthosis, connected to thighs and shanks, that moves a patient's legs as a function of the EMG signals measured from the wearer ([6] and [7]).

In our research work at Berkeley, we have separated the technology associated with human power augmentation into lower extremity exoskeletons and upper extremity exoskeletons. The reason for this was two-fold; firstly, we could envision a great many applications for either a stand-alone lower or upper extremity exoskeleton in the immediate future. Secondly, and more importantly for the division is that the exoskeletons are in their early stages, and further research still needs to be conducted to ensure that the upper extremity exoskeleton and lower extremity exoskeleton can function well independently before we can venture an attempt to integrate them. With this in mind, we proceeded with the designs of the lower and upper extremity exoskeleton separately, with little concern for the development of an integrated exoskeleton.

In the mid-1980s, we initiated several research projects on upper extremity exoskeleton systems, billed as "human extenders" [8], [9], and [10]. The main function of an upper extremity exoskeleton is human power augmentation for manipulation of heavy and bulky objects. These systems, which are also known as assist devices or human power extenders, can simulate forces on a worker's arms and torso. These forces differ from, and are usually much less than the forces needed to maneuver a load. When a worker uses an upper extremity exoskeleton to move a load, the device bears the bulk of the weight by itself, while transferring to the user as a natural feedback, a scaled-down value of the loads actual weight. For example, for every 40 pounds of weight from an object, a worker might support only 4 pounds while the device supports the remaining 36 pounds. In this fashion, the worker can still sense the load's weight and judge his/her movements accordingly, but the force he/she feels is much smaller than what he/she would feel without the device. In another example, suppose the worker uses the device to maneuver a large, rigid, and bulky object, such as an exhaust pipe. The device will convey the force to the worker as if it was a light, single-point mass. This limits the cross-coupled and centrifugal forces that increase the difficulty of maneuvering a rigid body and can sometimes produce injurious forces on the wrist. In a third example, suppose a worker uses the device to handle a powered torque wrench. The device will decrease and filter the forces transferred from the wrench to the worker's arm so the worker feels the low-frequency components of the wrenchs vibratory forces instead of the high- frequency components that produce fatigue.

The Berkeley Lower Extremity Exoskeleton (BLEEX) is not an orthosis or a brace; unlike the above systems it is designed to carry a heavy load by transferring the load weight to the ground (not to the wearer). BLEEX has four new features. First, a novel control architecture was developed that controls the exoskeleton through measurements of the exoskeleton itself [1].

This eliminated problematic human induced instability [14] due to sensing the human force. Second, a series of high specific power and specific energy power supplies were developed that were small enough to make BLEEX a true field-operational system [18], [23], [25]. Third, a body LAN (Local Area Network) with a special communication protocol and hardware were developed to simplify and reduce the cabling task of all the sensors and actuators needed for exoskeleton control [15] and [16]. Finally, a versatile architecture was chosen to decrease complexity and power consumption.

References

- [1] Chu, A., Kazerooni, H., Zoss, A., "On the Biomimetic Design of the Berkeley Lower Extremity Exoskeleton", IEEE Int. Conf. on Robotics and Automation, April 2005, Barcelona.
- [2] GE Co., "Hardiman I Arm Test", General Electric Report S-70-1019, Schenectady, NY, 69.
- [3] General Electric Co., "Hardiman I Prototype Project, Special Interim Study", General Electric Report S-68-1060, Schenectady, NY, 1968.
- [4] Groshaw, P. F., General Electric Co., "Hardiman I Arm Test, Hardiman I Prototype", General Electric Report S-70-1019, Schenectady, NY, 1969.
- [5] Hristic, D., Vukobratovic, M., "Development of Active Aids for Handicapped", Proc. III International Conference on bio-Medical Engineering, Sorrento, Italy, 1973.
- [6] Kawamoto, H., Kanbe, S., Sankai, Y., "Power Assist Method for HAL-3 Estimating Operator's Intention Based on Motion Information", in Proc. of 2003 IEEE Workshop on Robot and Human Interactive Communication, Millbrae, CA, 2003.
- [7] Kawamoto, H., Sankai, Y., "Power Assist System HAL-3 for gait Disorder Person", ICCHP, July 2002, Austria.
- [8] Kazerooni, H., "The Human Power Amplifier Technology at the University of California, Berkeley", Journal of Robotics and Autonomous Systems, Elsevier, Volume 19, 1996, pp. 179-187.
- [9] Kazerooni, H., "Human-Robot Interaction via the Transfer of Power and Information Signals, IEEE Transactions on Systems and Cybernetics, Vol. 20, No. 2, April 1990, pp. 450-463.
- [10] Kazerooni, H., Guo, J., "Human Extenders, ASME Journal of Dynamic Systems, Measurements, and Control, Vol. 115, No. 2(B), June 1993, pp 281-289.
- [11] Kazerooni, H., and Her, M. "The Dynamics and Control of a Haptic Interface Device, IEEE Transactions on Robotics and Automation, Vol. 10, No. 4, August 1994, pp 453-464.
- [12] Kazerooni, H., Mahoney, S., "Dynamics and Control of Robotic Systems Worn By Humans, ASME J. of Dynamic Systems, Measurements, and Control, Vol. 113, No. 3, , Sept. 1991.

- [13] Kazerooni, H., Racine, J.-L., Huang, L., Steger, R., "On the Control of the Berkeley Lower Extremity Exoskeleton (BLEEX)", IEEE Int. Conf. on Robotics and Automation, April 2005, Barcelona.
- [14] Kazerooni, H., Snyder, T., A Case Study on Dynamics of Haptic Devices: Human Induced Instability in Powered Hand Controllers, AIAA J. of Guidance, Control, and Dyn., Vol. 18, No. 1, 95.
- [15] Kim, S., Anwar, G., Kazerooni, H., "High-speed Communication Network for Controls with Application on the Exoskeleton", American Control Conference, Boston, June 2004.
- [16] Kim, S., Kazerooni, H., "High Speed Ring-based distributed Networked control system For Real-Time Multivariable Applications", ASME International Mechanical Engineering Congress, Anaheim, CA, November 2004.
- [17] Makinson, B. J., General Electric Co., "Research and Development Prototype for Machine Augmentation of Human Strength and Endurance, Hardiman I Project", General Electric Report S-71- 1056, Schenectady, NY, 1971.
- [18] McGee, T., Raade, J., and Kazerooni, H., "Monopropellant-Driven Free Piston Hydraulic Pump for Mobile Robotic Systems", Journal of Dynamic Systems, Measurement and Control, Vol. 126, March 2004, pp. 75-81.
- [19] Mizen, N. J., "Preliminary Design for the Shoulders and Arms of a Powered, Exoskeletal Structure", Cornell Aeronautical Lab. Report VO-1692-V4, 1965.
- [20] Mosher, R. S., "Force-Reflecting Electrohydraulic manipulator", Electro-Technology, Dec. 1960.
- [21] Neuhaus, P., Kazerooni, H., "Industrial-Strength Human-Assisted Walking Robots", IEEE Robotics and Automation Magazine, Vol. 8, No. 4., December 2001, pp. 18-25.
- [22] Pratt, J., Krupp, B., Morse, C., Collins, S., "The RoboKnee: An Exoskeleton for Enhancing Strength and Endurance During Walking", IEEE Conf. on Robotics and Aut., New Orleans, 2004.
- [23] Raade, J., Kazerooni, H., "Analysis and Design of a Novel Power Supply for Mobile Robots", IEEE International Conference on Robotics and Automation, New Orleans, LA, April 2004.
- [24] Vukobratovic, M., Ciric, V., Hristic, D., "Contribution to the Study of Active Exoskeletons", Proc. Of the 5th IFAC Congress, Paris, 1972.
- [25] Amundson, K., Raade, J., Harding, N., and Kazerooni, H. "Hybrid Hydraulic-Electric Power Unit for Field and Service Robots," Proc. of IEEE Intelligent Robots and Systems, Edmonton, Canada, 2005.

Autonomous Helicopter Tracking and Localization Using a Self-Surveying Camera Array

Masayoshi Matsuoka, Alan Chen, Surya P. N. Singh, Adam Coates, Andrew Y. Ng, and Sebastian Thrun

Stanford University; Stanford, CA 94305

{m.matsuoka, aychen, spns, acoates, ayn, thrun}@stanford.edu

Summary. This paper describes an algorithm that tracks and localizes a helicopter using a ground-based trinocular camera array. The three cameras are placed independently in an arbitrary arrangement that allows each camera to view the helicopter's flight volume. The helicopter then flies an unplanned path that allows the cameras to self-survey utilizing an algorithm based on structure from motion and bundle adjustment. This yields the camera's extrinsic parameters allowing for real-time positioning of the helicopter's position in a camera array based coordinate frame. In fielded experiments, there is less than a $2m$ RMS tracking error and the update rate of $20Hz$ is comparable to DGPS update rates. This system has successfully been integrated with an IMU to provide a positioning system for autonomous hovering.

Keywords: structure from motion, bundle adjustment, self-surveying cameras, camera tracking, camera localization

1 Introduction

Position estimation is of critical importance in autonomous robotics research as it is the principal measurement used in machine control and localizing collected data. [1] We utilize three ground based cameras to track and localize one of the Stanford autonomous helicopters (Fig. 1). This system replaces an onboard DGPS system, making the positioning system more robust during aggressive flight maneuvers. DGPS is unreliable because directional GPS antennas are prone to signal occlusions during rolls and omnidirectional antennas are susceptible to multipath during upright flight. Also, by moving the positioning equipment off the helicopter, the weight is reduced allowing the helicopter more power for maneuvering. The cameras are placed on the ground in unsurveyed positions that will allow them to see the helicopter at all times. Because the rotation and translation relationship between each camera is unknown, this extrinsic data will need to be extracted through self-surveying of

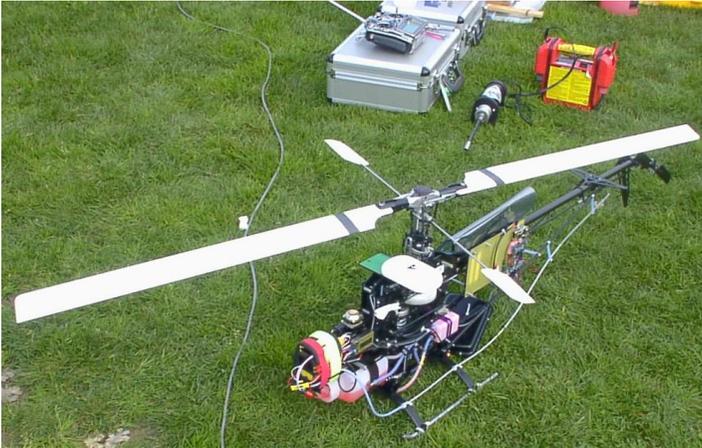


Fig. 1. One of Stanford's autonomous helicopters

the array. Once the extrinsic data has been determined, then the 3-D location of the aerial vehicle can be accurately and robustly tracked in a camera array based coordinate frame with standard least squares, LS, techniques.

The core problems in this project are the localization of the helicopter in each image frame and the self-surveying of the extrinsic parameters for the three cameras. Background differencing is used to locate the helicopter in each image. Essentially, by identifying the background through an average of previous scenes the moving helicopter can be identified as cluster of points in the foreground image. The center of this cluster identifies the approximate center of the helicopter.

Extrinsic information is usually obtained via calibration of the cameras in the scene utilizing a calibration object, such as a cube with a checkerboard pattern, or the cameras are fixed in locations and orientations with known extrinsic parameters. [2] This is not ideal in a field environment because the above methods would require a recalibration of the cameras with a large calibration aid every time a camera is jostled or would require a large structure that would fix the cameras in relation to each other while providing enough coverage to view the entire scene. Thus, the process of camera self-surveying is crucial to the tracking problem. Through this, the camera array will be able to estimate its geometry on the fly while deployed in the field without requiring modifications to the scene or the helicopter. Our approach uses multiple observations of the same scene motion to recover the extrinsic relationships between the cameras. In particular, this is done using a variant of the structure from motion (SFM) algorithm [3] and bundle adjustment [4].

Surveying and calibration will be used interchangeably throughout this paper. When we talk about calibration however, we are only referring to calibrating the extrinsic parameters of our camera array.

2 Background

There are several related localization approaches in the field. [5, 6, 7] Approaches like DGPS [5, 6] and radar provide high precision localization accuracy, but tend to be expensive, hard to relocate, prone to occlusion, or have to be deployed on the vehicle. Like the directional GPS antenna, an on-board camera system is also susceptible to occlusions when the helicopter rolls and pitches.[7] Inertial techniques provide high fidelity, but introduce significant drift error. Our system can be useful as a low-cost portable alternative to standard positioning systems without adding hardware to the helicopter.

The self-surveying ability of our system allows us to place the cameras anywhere on a field such that the cameras cover the operating space where the helicopter will fly and have the helicopter in focus. Self-calibration to acquire extrinsic parameters has been done by groups in the past. [8, 9] The main difference is that they move the stereo cameras in order to extract parameters while we will be moving a point in the image to extract the same type of information. For example, Knight and Reid use a stereo head that rotates around an axis to give calibration and head geometry. [10] Zhang shows that four points and several images from a stereo pair which has moved randomly, but is constant with respect to each other, can be used to compute the relative location and orientation of the cameras along with the 3-D structure of the points up to a scale factor. [11] Our self-surveying technique utilizes an algorithm developed by Poelman and Kanade. They use one camera tracking several feature points and take a stream of images while moving the camera. With this data, they can determine the motion of the camera and the coordinates of each of the feature points. [3]

3 Tracking Approach

A background differencing method is utilized to extract the location of the helicopter in images coordinates from the black and white pictures. First, the statistical model of the background is built by updating a running average of the image sequence over time, with I as a pixel intensity value:

$$I_j^{background}(u, v) = (1 - \alpha)I_{j-1}^{background}(u, v) + \alpha I_j^{current}(u, v) \quad (1)$$

where α regulates updating speed. [4] Next, the algorithm takes an image difference of the current image and the background image, and then thresholds out the image difference caused by noise:

$$I_j^{difference}(u, v) = \begin{cases} I = I_j^{current}(u, v) - I_j^{background}(u, v) & |I| \geq I^{threshold} \\ 0 & |I| < I^{threshold} \end{cases} \quad (2)$$

Finally, the estimate of a moving object in the image coordinate (u_j, v_j) is estimated by the population mean of the non-zero pixel distribution of the image difference:

$$u_j = \frac{1}{k} \sum_m \sum_n m I_j^{difference}(m, n) \quad (3)$$

$$v_j = \frac{1}{k} \sum_m \sum_n n I_j^{difference}(m, n) \quad (4)$$

Here, the search window (m, n) is a square mask, containing k pixels, centered at the helicopter location in the previous time step. This eliminates unrealistic abrupt jumps in the helicopter location estimate caused by noise and other moving objects elsewhere in the image.

This simple windowed background differencing method works when the helicopter is the principal actively moving object in the search window. Although slow-moving disturbances like clouds in the sky can be distinguished from the helicopter by tuning α and the threshold to appropriate values, this algorithm may be confused when other fast moving objects are in its windowed view, such as swaying trees or airplanes in the background.

As suggested in related literature, the tracking performance can be greatly improved by taking the probabilities of the predicted target dynamics into consideration, for instance, using Kalman filtering [2], the condensation algorithm [12], or multiple hypothesis tracking [13]. In this research, the Kalman filter approach is implemented to improve robustness in maintaining a lock on the helicopter in this specific helicopter tracking environment. However, in the experimental setup used in section 5.3 (in which the helicopter flies above the treeline in each of the camera views), the algorithm does well even without a Kalman filter.

4 Self-Calibration Algorithm

4.1 Structure from Motion

To calibrate the extrinsic parameters of the system, a structure from motion technique based on the algorithm defined by Poelman and Kanade in 1997 will be used for an initial estimate. [3] As opposed to taking a single camera and taking a stream of images of an object as we move the camera, we will use static cameras and take a stream of images as we move the object in the scene. This will provide the data necessary to utilize the algorithm described below.

The equation below shows the standard camera conversion equations:

$$p_j = R_i(P_j + t_i) \quad (5)$$

$$R_i = \begin{pmatrix} i_x \\ j_x \\ k_x \end{pmatrix}, t_i = \begin{pmatrix} t_{ix} \\ t_{iy} \\ t_{iz} \end{pmatrix}, p_j = \begin{pmatrix} p_{jx} \\ p_{jy} \\ p_{jz} \end{pmatrix}, P_j = \begin{pmatrix} P_{jx} \\ P_{jy} \\ P_{jz} \end{pmatrix} \quad (6)$$

- M : number of cameras (3)
- N : length of flight
- i : camera (1,2,...,M)
- j : sampling epoch (1,2,...,N)
- t_i : the location of the camera i in the world frame
- P_j : the helicopter trajectory in the world frame
- p_{ij} : the helicopter trajectory in camera i frame
- R_i : rotation matrix for camera i
- u_{ij}, v_{ij} : pixel values of the helicopter at epoch j in camera i

To convert from 3-D camera frame coordinates to a 2-D image frame coordinate system, a scaled orthographic projection, also know as “weak perspective,” will be used. This projection technique, shown in the equation below, approximates perspective projections when the object in the image is near the image center and does not vary a large amount in the axis perpendicular to the camera’s image plane. The equations below assume unit focal length and that the world’s origin is now fixed at the center of mass of the objects in view.

$$x_i = \frac{t_i \cdot i_i}{z_i}, \quad y_i = \frac{t_i \cdot j_i}{z_i}, \quad z_i = t_i \cdot k_i \quad (7)$$

$$u_{ij} = \frac{p_{jx}}{z_i} = m_i \cdot P_j + x_i \quad (8)$$

$$v_{ij} = \frac{p_{jy}}{z_i} = n_i \cdot P_j + y_i \quad (9)$$

$$m_i = \frac{i_i}{z_i}, \quad n_i = \frac{j_i}{z_i} \quad (10)$$

$$W = R^*P + t^* \quad (11)$$

$$W = \begin{pmatrix} u_{11} & \dots & u_{1N} \\ v_{11} & \dots & v_{1N} \\ \vdots & & \vdots \\ u_{M1} & \dots & u_{MN} \\ v_{M1} & \dots & v_{MN} \end{pmatrix}, \quad R^* = \begin{pmatrix} m_1 \\ n_1 \\ \vdots \\ m_M \\ n_M \end{pmatrix}, \quad t^* = \begin{pmatrix} x_1 & \dots & x_1 \\ y_1 & \dots & y_1 \\ \vdots & & \vdots \\ x_M & \dots & x_M \\ y_M & \dots & y_M \end{pmatrix} \quad (12)$$

Using the helicopter’s trajectory in each of the cameras, (u_{ij}, v_{ij}) , we can solve for the measurement matrix W^* . Taking the singular value decomposition of W^* and ignoring any right or left singular eigenvectors that correspond with the 4th or higher singular values (that appear due to noise) results with:

$$x_i = \frac{1}{N} \sum_{j=1}^N u_{ij}, \quad y_i = \frac{1}{N} \sum_{j=1}^N v_{ij} \quad (13)$$

$$W^* = W - t^* = R^*P \approx U_{2M \times 3} \Sigma_{3 \times 3} V_{3 \times N}^T = \tilde{R} \tilde{P} \quad (14)$$

$$\tilde{R} = U, \quad \tilde{P} = \Sigma V^T \quad (15)$$

\tilde{R} and \tilde{P} represent the affine camera positions and the affine structure of the points in the scene respectively which can then be transferred back to Euclidian space with a matrix Q . To determine Q we will use the $2M + 1$ linear constraints defined below. The last constraint will avoid the trivial solution satisfied by everything being zero.

$$W^* = \tilde{R}QQ^{-1}\tilde{P} \quad (16)$$

$$|m_i|^2 = |n_i|^2 = \frac{1}{z_i^2} \Rightarrow |m_i| - |n_i| = 0 \quad (17)$$

$$m_i \cdot n_i = 0 \quad (18)$$

$$|m_1| = 1 \quad (19)$$

With these constraints and the Jacobi Transformation of Q the affine system can then be converted back into Euclidian space. If the resulting Q is not positive definite, then distortions, possibly due to noise, perspective effects, insufficient rotation in the system, or a planar flight path, has overcome the third singular value of W . [3]

We multiply all the rotation matrices and the newly found matrix of points by R_1^{-1} to convert everything into a coordinate frame based on the camera 1 image frame.

After this process, the only remaining extrinsic parameters still unknown is t_i . To find t_i , LS can be used by expanding the equation below to encompass all the points in each camera.

$$\begin{pmatrix} u_{ij} \\ v_{ij} \\ z_i \end{pmatrix} - \begin{pmatrix} i_i \cdot p_j \\ j_i \cdot p_j \\ 0 \end{pmatrix} = R_i t_i \quad (20)$$

The minimum number of points required to self-survey with structure from motion is defined by

$$2MN > 8M + 3N - 12 \quad (21)$$

Given that three cameras will be used, a minimum of four points will be necessary to self-survey. Because our cameras are static, we can fly the helicopter to four different locations and record images at each location. This will provide the minimum points necessary to self-survey. [14]

4.2 Camera Frame to World Frame

The resulting extrinsic parameters of the camera array are unscaled and given in the camera 1 image frame. To extract the unknown scale factor inherent to these type of vision problems, the distance L between camera 1 and camera 2 is measured. The ratio of that distance to the unscaled distance between camera 1 and camera 2 is defined as the scale factor.

To rotate the extrinsic parameters from the camera 1 image frame to a world frame, a rotation matrix is created based on the following assumptions (see also Fig. 2(b)):

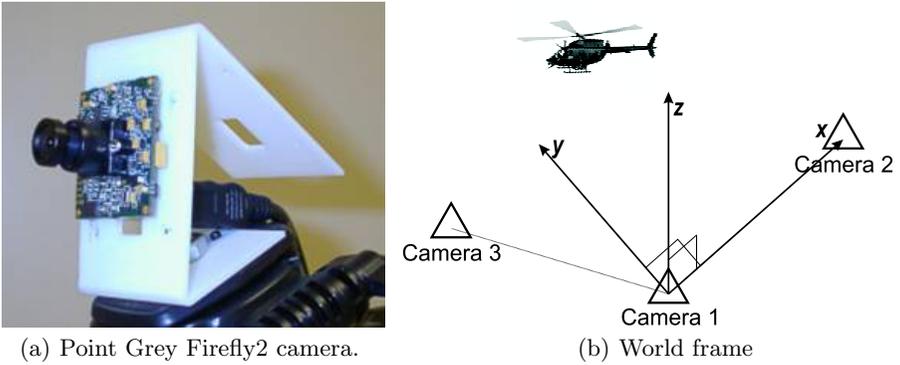


Fig. 2. Camera setup

1. Camera 1 is at the origin of the world frame.
2. The vector from camera 1 to camera 2 is the x axis.
3. All the cameras are in the $x - y$ plane.
4. The y axis is defined as towards the helicopter, but orthogonal to the x axis and in the $x - y$ plane.
5. The z axis is then defined by the right hand rule (approximately straight up)

This results in:

$$t_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, t_2 = \begin{bmatrix} L \\ 0 \\ 0 \end{bmatrix}, t_3 = \begin{bmatrix} x_3 \\ y_3 \\ 0 \end{bmatrix} \quad (22)$$

4.3 Bundle Adjustment

Given the SFM solution as initial estimate, the calibration parameters can be refined further by solving nonlinear perspective equations directly via iterative LS, bundle adjustment. [4] The bundle adjustment technique optimizes the calibration parameters, exploring the best array geometry that matches to the set of visual tracking measurements collected during a calibration flight.

The calibration parameters estimated by the LS batch process include the camera locations in the world, the camera orientations, and the helicopter trajectory. Specifically, the following extrinsic parameters are the unknowns to be estimated: t_i , P_j , and the Euler angles associated with R_i , (α_i , β_i , and γ_i).

The set of normalized 2-D tracking points, (u_{ij}, v_{ij}) , in the image coordinates is the sole measurement used in this calibration process (except for the measurement L). The following perspective geometry equations relate all

the unknown parameters to the 2-D tracking points via a nonlinear perspective model. [2] (23) is different than (8) and (9) because here we are using a perspective model for the cameras.

$$u_{ij} = \frac{P_{jx}}{P_{jz}}, \quad v_{ij} = \frac{P_{jy}}{P_{jz}} \quad (23)$$

$$R_i = \begin{bmatrix} \cos \gamma_i & \sin \gamma_i & 0 \\ -\sin \gamma_i & \cos \gamma_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta_i & 0 & -\sin \beta_i \\ 0 & 1 & 0 \\ \sin \beta_i & 0 & \cos \beta_i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_i & \sin \alpha_i \\ 0 & -\sin \alpha_i & \cos \alpha_i \end{bmatrix} \quad (24)$$

The bundle adjustment method linearizes the perspective equations (5), (23), and (24) into a Jacobian form, and then batch-estimates the unknown calibration parameters via the iterative LS by taking the pseudo-inverse of the Jacobian matrix J of the linearized measurement equations (25):

$$\delta \begin{bmatrix} u'_{ij} \\ v'_{ij} \end{bmatrix} = J \delta \begin{bmatrix} t_i \\ \alpha_i \\ \beta_i \\ \gamma_i \\ P_j \end{bmatrix} \Rightarrow \delta \begin{bmatrix} t_i \\ \alpha_i \\ \beta_i \\ \gamma_i \\ P_j \end{bmatrix} = (J^T J)^{-1} J^T \delta \begin{bmatrix} u'_{ij} \\ v'_{ij} \end{bmatrix} \quad (25)$$

For all the unknowns to be observable, the Jacobian matrix J must be well-conditioned. Capturing a certain geometry change by tracking the helicopter simultaneously at the three cameras yields enough observability for the LS estimate. Also, to ensure proper convergence in the nonlinear LS iteration, bundle adjustment is seeded with multiple sets of initial estimates centered around the SFM solution to avoid converging to a local minimum.

5 Field Demonstration

5.1 Experimental Setup

The current prototype system consists of a helicopter platform and a ground-based camera array, Fig. 3. The camera array includes three compact digital cameras (Point Grey Firefly2 cameras, Fig. 2(a), using a Firewire interface) all connected to a single PC. An image from each camera is captured, nearly simultaneously, at a resolution of 640×480 in an 8-bit grayscale format at a rate of $20Hz$.

5.2 Tracking

The tracking algorithm based on the background differencing method was implemented in the field on each camera to track a common helicopter. Fig.



Fig. 3. Experimental setup

4(a) shows an image from one of the cameras during the test. The black box is the tracking marker centered at the estimated helicopter location and the thin white larger box is the search window of the background differencing method.

This particular flight test was conducted in an open field on Stanford’s campus next to a road where moving cars and walking people constantly came in and out of the scene. While the windowed background differencing-only method frequently failed to track a low flying helicopter in such a busy environment, the Kalman filter was able to maintain the lock on the helicopter during the flight.

Fig. 4(b) shows the resulting helicopter trajectory in the image coordinate for camera 1. The solid lines show the helicopter trajectory tracked by the Kalman filter. The dashed lines show the helicopter trajectory manually post-traced in the logged images as true reference. Although the Kalman filter was able to keep tracking the helicopter, the tracking markers were sometimes lagging in tracking the helicopter when the helicopter accelerated faster than the pre-defined dynamic model in the Kalman filter equations; we believe that fine tuning the process noise covariance will further improve performance. The mean errors between the Kalman filter and the true references were roughly 7.5 pixels, as shown in Table 1(a).

Table 1. Error tables

(a) Tracking errors in pixels

| | mean(pixel) | std(pixel) |
|----------|-------------|------------|
| Camera 1 | 6.9 | 5.0 |
| Camera 2 | 7.2 | 5.2 |
| Camera 3 | 8.3 | 6.2 |

(b) Localization error

| | x(m) | y(m) | z(m) |
|------|-------|------|------|
| mean | -0.17 | 1.39 | 0.27 |
| std | 1.07 | 0.99 | 0.52 |

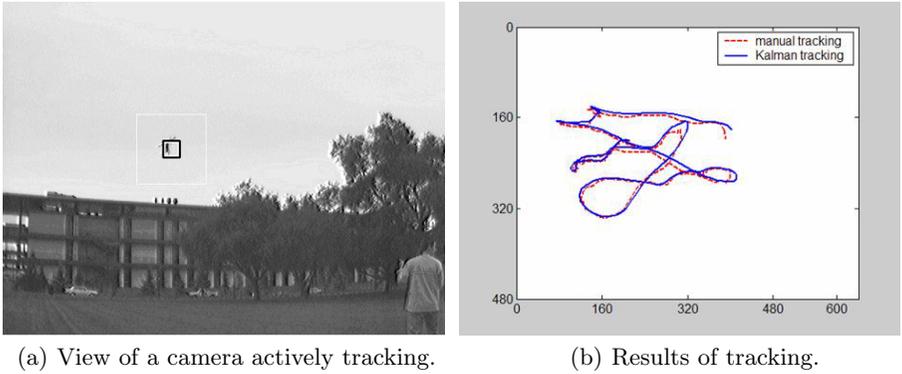


Fig. 4. Tracking the helicopter in a busy scene.

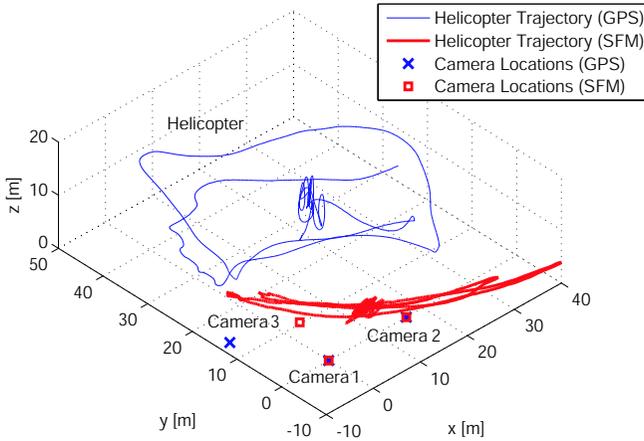
5.3 Localization

To check the validity of the localization algorithm the results from the calibration algorithm are compared with DGPS data, Fig. 5(a) and Fig. 5(b). Fig. 5(a) shows the results from SFM which is used to feed bundle adjustment. As the plot shows, care needs to be taken in picking points to initialize SFM because of the near-perspective assumption. Fig. 5(b) shows that the result from bundle adjustment follows DGPS pretty well. There are some small offsets that are probably due to the assumptions made in section 4.2. There is also some small variations in the trajectory reported by the vision system which likely result from small errors in the helicopter tracking system. Overall, the vision results match fairly well with the DGPS data. The errors are reported in Table 1(b).

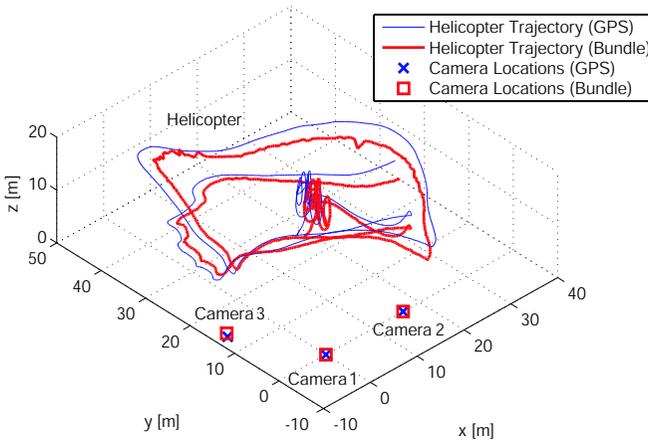
6 Conclusions

The self-surveying and tracking camera array presented in this paper produces an effective localization system that extends ideas from SFM and bundle adjustment. This is then combined with stereo tracking methods to generate a least squares measurement of the helicopter’s location.

The results presented document the tracking performance of this method in a field environment for a series of cameras whose extrinsic parameters are not known *a priori*. The calibration performance was tested against DGPS and was found to have less than a 2m RMS tracking error. The 20Hz update rate of the system is comparable to DGPS, and we obtain a tracking latency of less than 100ms. This makes it feasible to use this system as part of an autonomous flight controller.



(a) Structure from Motion.



(b) Bundle adjustment.

Fig. 5. 3-D plots of DGPS vs. ...

Because of the near-perspective assumption, it is better to run SFM on fewer points where the helicopter is near the center of the image as opposed to a large set of data where the helicopter's route spans the entire image. To make this procedure more robust, a paraperspective SFM [3] or a perspective SFM [13] can be used to initialize bundle adjustment.

Recent autonomous hover flights have demonstrated the capability of this system for real-time fielded operations. [15] Future work will test its use for acrobatic flights, find ways to maximize the flight volume, and make the system more robust to dropouts where the helicopter leaves one camera's field of view.

Acknowledgment

We would like to thank Ben Tse, our helicopter pilot, for his help and advice during our numerous experiments. We would also like thank Gary Bradski for helpful conversations about this work.

References

1. P. Maybeck, Stochastic Models, Estimation, and Control, Volume 1. Academic Press, Inc, 1979.
2. E. Trucco and A. Verri, Introductory Techniques for 3-D Computer Vision. Prentice Hall, 1998.
3. C. Poelman and T. Kanade, “A paraperspective factorization method shape and motion recovery,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 3, Mar. 1997.
4. D. A. Forsyth and J. Ponce, Computer Vision: A Modern Approach. Prentice Hall, 2003.
5. M. Whalley, M. Freed, M. Takahashi, D. Christian, A. Patterson-Hine, G. Schulein, and H. R., “The NASA / Army Autonomous Rotorcraft Project,” in Proceedings of the American Helicopter Society 59th Annual Forum, Phoenix, Arizona, 2003.
6. S. Saripalli, J. Montgomery, and G. Sukhatme, “Visually-guided landing of an autonomous aerial vehicle,” IEEE Transactions on Robotics and Automation, 2002.
7. J. M. Roberts, P. I. Corke, and G. Buskey, “Low-cost flight control system for a small autonomous helicopter,” in IEEE International Conference on Robotics and Automation, 2003.
8. P. Liang, P. Chang, and S. Hackwood, “Adaptive self-calibration of vision-based robot systems,” IEEE Transactions on Systems, Man and Cybernetics, vol. 19, no. 4, pp. 811–824, July 1989.
9. G. Mayer, H. Utz, and G. Kraetzschmar, “Towards autonomous vision self-calibration for soccer robots,” Proc. of the Intelligent Robot and Systems (IROS) Conference, vol. 1, pp. 214–219, 2002.
10. J. Knight and I. Reid, “Self-calibration of a stereo rig in a planar scene by data combination,” In Proc. of the Internatoinal Conference on Pattern Recognition, pp. 1411–1414, Sept. 2000.
11. Z. Zhang, “Motion and structure of four points from one motion of a stereo rig with unknown extrinsic parameters,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 12, Dec. 1995.
12. S. Blackman, “Multiple hypothesis tracking for multiple target tracking,” IEEE Aerospace and Electronic Systems Magazine, vol. 19, no. 1, Jan. 2004.
13. M. Han and T. Kanade, “Perspective factorization methods for euclidean reconstruction,” Carnegie Mellon, Tech. Rep. CMU-RI-TR-99-22, Aug. 1999.
14. S. Thrun, G. Bradski, and D. Russakoff, “Struction from motion,” Feb. 2004, lecture Notes from CS223b.
15. A. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, “Inverted autonomous helicopter flight via reinforcement learning,” International Symposium on Experimental Robotics, 2004.

Visual Motion Estimation for an Autonomous Underwater Reef Monitoring Robot

Matthew Dunbabin, Kane Usher, and Peter Corke

CSIRO ICT Centre, PO Box 883 Kenmore QLD 4069, Australia

Summary. Performing reliable localisation and navigation within highly unstructured underwater coral reef environments is a difficult task at the best of times. Typical research and commercial underwater vehicles use expensive acoustic positioning and sonar systems which require significant external infrastructure to operate effectively. This paper is focused on the development of a robust vision-based motion estimation technique using low-cost sensors for performing real-time autonomous and untethered environmental monitoring tasks in the Great Barrier Reef without the use of acoustic positioning. The technique is experimentally shown to provide accurate odometry and terrain profile information suitable for input into the vehicle controller to perform a range of environmental monitoring tasks.

1 Introduction

In light of recent advances in computing and energy storage hardware, Autonomous Underwater Vehicles (AUVs) are emerging as the next viable alternative to human divers for remote monitoring and survey tasks. There are a number of remotely operated (ROV) and AUVs performing various monitoring tasks around the world [17]. These vehicles are typically large and expensive, require considerable external infrastructure for accurate positioning, and need more than one person to operate a single vehicle. These vehicles also generally avoid the highly unstructured reef environments such as Australia's Great Barrier Reef, with limited research performed on shallow water applications and reef traversing. Where surveying at greater depths is required, ROV's have been used for video transects and biomass identification, however, these vehicles still require the human operator in the loop.

Knowing the position and distance a AUV has moved is critical to ensure that correct and repeatable measurements are being taken for reef surveying applications. It is important to have accurate odometry to ensure survey transect paths are correctly followed. A number of techniques are used to estimate vehicle motion. Acoustic sensors such as Doppler velocity logs are a common means of obtaining accurate motion information. The use of vision

for motion estimation is becoming a popular technique for underwater use allowing navigation, station keeping, and the provision of manipulator feedback information [16, 12, 15]. The accuracy of underwater vision is dependent on visibility and lighting, as well as optical distortion resulting from varying refractive indices, requiring either corrective lenses or careful calibration[4].

Visual information is often fused with various acoustic sensors to achieve increased sensor resolution and accuracy for underwater navigation [10]. Although this fusion can result in very accurate motion estimation compared to vision only, it is typically performed off-line and in deeper water applications.

A number of authors have investigated different techniques for odometry estimation using vision as the primary sensor. Amidi [2] provides a detailed investigation into feature tracking for visual odometry for an autonomous helicopter. Another technique to determine camera motion is structure-from-motion (SFM) with a comparison of a number of SFM techniques in terms of accuracy and computational efficiency given by Adams[1]. Corke [7] presents experimental results for odometry estimation of a planetary rover using omnidirectional vision and compares robust optic flow and SFM methods with very encouraging results.

This research is focused on autonomously performing surveying tasks based around the Great Barrier Reef using low-cost AUV's and vision as the primary sensor for motion estimation. The use of vision in this environment is considered a powerful technique due to the feature rich terrain. However, at the same time it can cause problems for traditional processing techniques with highly unstructured terrain, soft swaying corals, moving biomass and lighting ripple due to surface waves.

The focus of this paper is on the development of a robust real-time vision-based motion estimation technique for a field deployed AUV which uses intelligently fused low-cost sensors and hardware, and without the use of acoustic positioning or artificial lighting.

2 Vision System

2.1 Vehicle

The vehicle developed and used in this research was custom designed to autonomously perform the environmental monitoring tasks required by the reef monitoring organisations [14]. To achieve these tasks, the vehicle must navigate over highly unstructured surfaces at fixed altitudes (300-500mm above the sea floor) and at depths in excess of 100m, in cross currents of 2 knots and know its position during linear transects to within 5% of total distance travelled. It was also considered essential that the vehicle be untethered to reduce risk of entanglement, the need for support vessels and reducing drag imposed on the vehicle operating in strong currents.

Fig. 1 shows the hybrid vehicle design named “Starbug” developed as part of this research. The vehicle can operate remotely or fully autonomously. Details of the vehicle performance and system integration are given in [9].

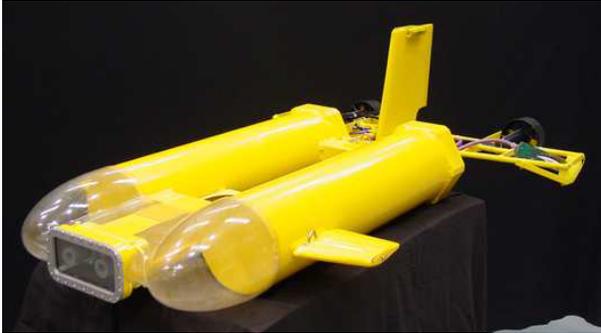


Fig. 1. The “Starbug” Autonomous Underwater Vehicle.

2.2 Sensors

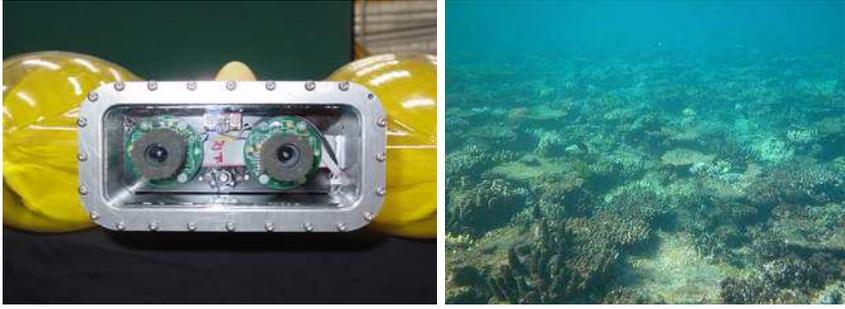
The sensor platform developed for the Starbug AUV and used in this research has been based on past experience with the CSIRO autonomous airborne system [6] and enhanced to allow a low-cost navigation suite for the task of long-term autonomous reef monitoring [8]. The primary sensing component of the AUV is the stereo camera system. The AUV has two stereo heads with one looking downward to estimate altitude above the sea-floor and odometry, and the other looking forward for obstacle avoidance (not used in this study). The cameras used are a colour CMOS sensor from Omnivision with 12mm diameter screw fit lenses which have a nominal focal length of 6mm.

Each stereo pair has the cameras set with a baseline of 70mm which allows an effective distance resolution in the range 0.2 to 1.7m. The cameras look through 6mm thick flat glass. The two cameras are tightly synchronized and line multiplexed into PAL format composite video signal. Fig. 2 shows the stereo camera head used in the AUV and an representative image of the typical terrain and visibility that system operates.

In addition to the vision sensors, the vehicle has a magnetic compass, custom built IMU (see [8] for details), pressure sensor (2.5mm resolution), a PC/104 800MHz Crusoe computer stack running the Linux OS, and a GPS which is used when surfaced.

3 Optimised Vision-Based Motion Estimation

Due to the unique characteristics of the reef environment such as highly unstructured and feature rich terrain, relatively shallow waters and sufficient



(a) Stereo camera pair

(b) Typical reef terrain

Fig. 2. Forward looking stereo camera system and representative reef environment.

natural lighting, vision is considered a viable alternative to typical expensive acoustic positioning and sonar sensors for navigation.

The system uses reasonable quality CMOS cameras with low-quality miniature glass lenses. Therefore, it is important to have an accurate model of the cameras intrinsic parameters as well as good knowledge of the camera pair extrinsic parameters. Refraction due to the air-water-glass interface also requires consideration as discussed in [8]. In this investigation the cameras are calibrated using standard automatic calibration techniques (see e.g. Bouguet[3]) to combine the effects of radial lens distortion and refraction.

In addition to assuming an appropriately calibrated stereo camera pair, it is also assumed that the AUV is initialised at a known start position and heading angle. The complete procedure for this odometry technique is outlined in Algorithm 1.

The key components of this technique are image processing which we have termed three-way feature matching (steps 1-7) which utilises common well behaved procedures, and motion estimation (steps 8-10) which is the primary contribution of this paper. These components are discussed in the following sections.

3.1 Three-Way Feature Matching

Feature extraction

In this investigation, the Harris feature detector [5] has been implemented due to its speed and satisfactory results. Roberts[13] compared the temporal stability for outdoor applications and found the Harris operator to be superior to other feature extraction methods. Only features that are matched both in stereo (spatially) for height reconstruction, and temporally for motion reconstruction are considered for odometry estimation. Typically, this means that

Algorithm 1 Visual motion estimation procedure.

1. Collect a stereo image.
 2. Find all features in the entire image.
 3. Take the 100 most dominant features as template (typically this number is more like 10-50 features).
 4. Match corners between stereo images by calculating the normalized cross-correlation (*ZNCC*).
 5. Store stereo matched features.
 6. Using stereo matched features at current time step, match these with stereo matched features from images taken at previous time step using *ZNCC*.
 7. Reconstruct those points which have been both spatially and temporally matched into 3D.
 8. Using the dual search optimisation technique outlined in Algorithm 2, determine the camera transformation that best describes motion from the previous to the current image.
 9. Using measured world heading, roll and pitch angles, transform the differential camera motion to a differential world motion.
 10. Integrate differential world motion to determine a world camera displacement.
 11. Go to step 1 and repeat.
-

between ten and fifty strong features are tracked at each sample time and during ocean trials with poor water clarity this was observed to be less than ten.

We are currently working on an improved robustness to feature extraction that consists of a combination of this higher frame rate extraction method with a slower loop running a more computationally expensive KLT (or similar) type tracker to track features over a longer time period. This will help to alleviate long term drift in integrating differential motion.

Stereo matching

Stereo matching is used in this investigation to estimate vehicle altitude, provide scaling for temporal feature motion and to generate coarse terrain profiles.

For stereo matching, the correspondences between features in the left and right images are found. The similarity between the regions surrounding each corner is computed (left to right) using the normalised cross correlation similarity measure (*ZNCC*).

To reduce computation, epipolar constraints are used to prune the search space and only the strongest corners are evaluated. Once a set of matches is found, the results are then refined with sub-pixel interpolation. Additionally, rather than correcting the entire image for lens distortion and refraction effects, the correction is applied only to the coordinate values of the tracked features, hence saving considerable computation.

Optic flow (motion matching)

The tracking of features temporally between image frames is similar to the spatial stereo matching as discussed above. Given the full set of corners extracted during stereo matching, similar techniques are used to find the corresponding corners from the previous image. Differential image motion (du, dv) is then calculated in both the u and v directions on a per feature basis.

To maintain suitable processing speeds, motion matching is currently constrained by search space pruning, whereby feature matching is performed within a disc of specified radius. The reduction of this search space size can potentially be achieved with a motion prediction model to estimate where the features lie in the search space.

In this motion estimation technique, temporal feature tracking currently only has a one frame memory. This reduces problems due to significant appearance change over time. However, as stated earlier, longer term tracking will improve integration drift problems.

3D feature reconstruction

Using the stereo matched corners, standard stereo reconstruction methods are then used to estimate a feature’s three-dimensional position. In our previous vision-based motion estimation involving aerial vehicles [6], the stereo data was processed to find a consistent plane. The underlying assumption for stereo and motion estimation was the existence of a flat ground plane. In this current application, it cannot be assumed that the ground is flat. Hence, vehicle height estimation must be performed on a per feature basis.

The primary purpose of 3D feature reconstruction in this investigation is for scaling feature disparity to enable visual odometry.

3.2 Motion Estimation

The first step in the visual motion estimation process is to find a set of points (features) which give a three-way match, that is, those points which have both a stereo match in the current frame and a corresponding matching corner from the previous frame as discussed in Section 3.1. Given this correspondence, the problem is formulated as one of optimization to find at time k a vehicle rotation and translation vector (\mathbf{x}_k) which best explains the observed visual motion and stereo reconstruction as shown in Fig. 3.

Fig. 3 shows the vehicle looking at a ground plane (not necessarily planar) at times $k - 1$ and k with the features as seen in the respective image planes shown for comparison. The basis behind this motion estimation is to optimise the differential rotation and translation pose vector ($d\mathbf{x}_{est}$) such that when used to transform the features from the current image plane to the previous image plane, minimises the median squared error between the predicted image displacement (du', dv') (as shown in the “reconstructed image plane”) and the

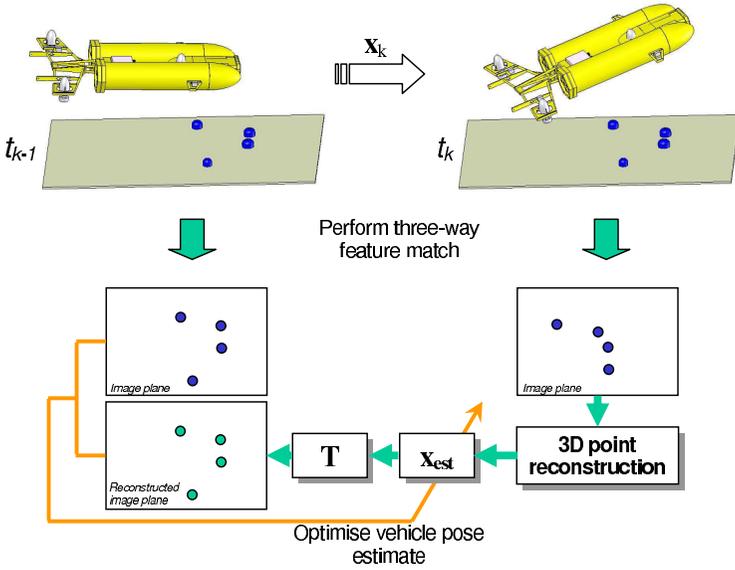


Fig. 3. Motion transformation from previous to current image plane.

actual image displacement (du, dv) provided from optic flow for each three-way matched feature.

During the pose vector optimisation, the Nelder-Mead simplex method[11] is employed to update the pose vector estimate. This nonlinear optimisation routine was chosen in this analysis due to its solution performance and the fact that it does not require the derivatives of the minimised function to be predetermined. The lack of gradient information allows this technique to be ‘model free’.

The pose vector optimisation consists of a two stage process at each time step to best estimate vehicle motion. Since the differential rotations (roll, pitch, yaw) are known from IMU measurements, the first optimisation routine is restricted to only update the translation components of the differential pose vector with the differential rotations held constant at their measured values. This is aimed at keeping the solution away from local minima. As there may be errors in the IMU measurements, a second search is conducted using the results from the first optimisation to seed the translation component of the pose estimate, with the entire pose vector now updated during the optimisation. This technique was found to provide more accurate results than a single search step as it helps in avoiding spurious local minima. Algorithm 2 describes the pose optimisation function used in this analysis for the first stage of the motion estimation. Note that in the second optimisation stage, the procedure is identical to Algorithm 2, however, $d\theta$, $d\alpha$ and $d\psi$ are also updated in Step 3 of the optimisation.

Algorithm 2 Pose optimisation function.

1. Seed search using the previous time step's differential pose estimate such that

$$d\mathbf{x} = [dx \ dy \ dz \ d\theta \ d\alpha \ d\psi]$$

where dx , dy and dz are the differential pose translations between the two time frames with respect to the current camera frame, and $d\theta$, $d\alpha$ and $d\psi$ are the differential roll, pitch and yaw angles respectively obtained from the IMU.

2. Enter optimisation loop.
3. Estimate the transformation vector from the previous to the current camera frame.

$$\mathbf{T} = \mathbf{R}_x(d\theta) \mathbf{R}_y(d\alpha) \mathbf{R}_z(d\psi) [dx \ dy \ dz]^T$$

4. For $i = 1..$ number of three-way matched features, repeat steps 5 to 9.
5. Displace the observed 3D reconstructed feature coordinates (x_i, y_i, z_i) from current frame to estimate where it was in the previous frame $(x_{e_i}, y_{e_i}, z_{e_i})$.

$$[x_{e_i} \ y_{e_i} \ z_{e_i}]^T = \mathbf{T} [x_i \ y_i \ z_i]^T$$

6. Project the current 3D feature points to the image plane to give (u_{o_i}, v_{o_i}) .
7. Project the displaced feature (step 5) to the image plane to give (u_{d_i}, v_{d_i}) .
8. Estimate the observed feature displacement on the image plane.

$$[du'_i \ dv'_i]^T = [u_{o_i} \ v_{o_i}]^T - [u_{d_i} \ v_{d_i}]^T$$

9. Compute the squared error between the estimated and actual feature displacement (du, dv) observed from optic flow.

$$e_i = (du_i - du'_i)^2 + (dv_i - dv'_i)^2$$

10. Using the median square error value (e_m) from all three-way matched features, update $d\mathbf{x}$ using the Nelder-Mead simplex method.
 11. If e_m is less than a preset threshold, end, else go to step 3 and repeat using the updated $d\mathbf{x}$.
-

The resulting optimised differential pose estimate at time k (\mathbf{x}_k) which is with respect to the camera coordinate system attached to the AUV can then be transformed to a consistent coordinate system using the roll, pitch and yaw data from the IMU. In this investigation, a homogeneous transformation (\mathbf{T}_H) of the camera motion is performed to determine the differential change in the world coordinate frame.

The differential motion vectors are then integrated over time to obtain the overall vehicle motion position vector at time t_f such that

$$\mathbf{x}_{t_f} = \sum_{k=0}^{t_f} \mathbf{T}_{H_k} d\mathbf{x}_k \quad (1)$$

It was observed that during ocean trials, varying lighting and structure could degrade the motion estimation performance due to insufficient three-way matched features being extracted. Therefore, a simple constant velocity vehicle model and motion limit filters (based on measured vehicle performance limitations) were added to improve motion estimation and discard obviously erroneous differential optimisation solutions. A more detailed hydrodynamic model is currently being evaluated to further improve predicted vehicle motion and aid in pruning the search space and optimisation seeding.

4 Experimental Results

The performance of the visual motion estimation technique described in Section 3 was evaluated in a test tank constructed at CSIRO’s QCAT site and during ocean trials. The test tank has a working section of 7.90 x 5.10m with a depth of 1.10m. The floor is lined with a sand coloured matting with pebbles, rocks of varying sizes and large submerged 3D objects to provide a texture and terrain surface for the vision system. Fig. 4 shows the AUV in the test tank and the ocean test site off Peel Island in Brisbane’s Moreton Bay.



(a) CSIRO QCAT test tank

(b) Ocean test site

Fig. 4. AUV during visual motion estimation experiments.

In the test tank the vehicle’s vision-based odometry system was ground truthed using two vertical rods attached to the AUV which protruded from the water’s surface. A SICK laser range scanner (PLS) was then used to track these points with respect to a fixed coordinate frame. By tracking these two points, both position and vehicle heading angle can be resolved. Fig. 5 shows

the results of the vehicle's estimated position using only vision-based motion estimation fused with inertial information during a short survey transect in the test tank. The ground truth obtained by the laser tracking system is shown for comparison.

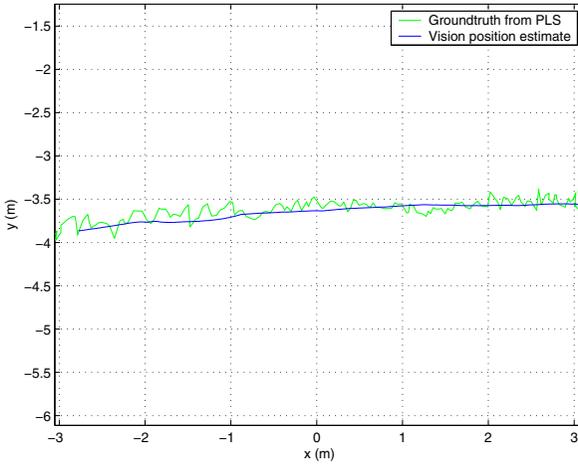


Fig. 5. Position estimation using only vision and inertial information in short survey transect. Also shown is a ground truth obtained from the laser system.

As seen in Fig. 5, the motion estimation compares very well with the ground truth estimation with a maximum error of approximately 2% at the end of the transect. Although, this performance is encouraging, work is being conducted to improve the position estimation over greater transect distances.

The ground truth system is not considered perfect (as seen by the noisy position trace in Fig. 5) due to resolution of the laser scanner and the size of the rods attached to the vehicle causing slight geometric errors. However, the system provides a stable position estimate over time for evaluation purposes.

A preliminary evaluation of the system was conducted during ocean tests over a hard coral and rock reef in Moreton Bay. The vehicle was set off to perform an autonomous untethered transect using the proposed visual odometry technique. The vehicle was surfaced at the start and end of the transect to obtain a GPS fix and provide a ground truth for the vehicle. Fig. 6 shows the results of a 53m transect as measured by the GPS.

In Fig. 6, the circles represent the GPS fix locations, and the line shows the vehicles estimated position during the transect. The results show that the vehicles position was estimated to within 4m of the actual end GPS given location or to within 8% of the total distance travelled. Given the poor water clarity and high wave action experienced during the experiment, the results are extremely encouraging.

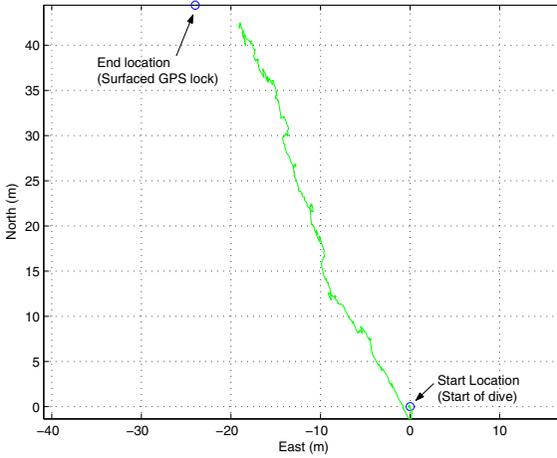


Fig. 6. Position estimation results for ocean transect.

5 Conclusion

This paper presents a new technique to estimate the egomotion and provide feedback for the real-time control of an autonomous underwater vehicle using only vision fused with low-resolution inertial information. A 3D motion estimation function was developed with the vehicle pose vector optimised using the nonlinear Nelder-Mead simplex method to minimise the median squared error between the predicted to observed camera motion between consecutive image frames. Experimental results show that the system performs well in representative tests with position estimation accuracy during simple survey transects of approximately 2% and in open ocean tests to 8%. The technique currently runs at better than 4Hz sample rate on the vehicle's onboard 800MHz Crusoe processor without code optimisation. Research is currently being undertaken to improve algorithm performance and processing speed.

Other areas of active research focus include improving system robustness against issues such as heading inaccuracies, lighting (wave “flicker”) and terrain structure variations including surface texture composition such as sea-grass, hard and soft corals to allow reliable in-field deployment.

Acknowledgment

The authors would like to thank the rest of the CSIRO robotics team: Graeme Winstanley, Jonathan Roberts, Les Overs, Stephen Brosnan, Elliot Duff, Pavan Sikka, and John Whitham.

References

1. H. Adams, S. Singh, and D. Strelow. An empirical comparison of methods for image-based motion estimation. In *Proceedings of the 2002 IEEE/IRJ International Conference on Intelligent Robots and Systems*, October 2002.
2. O. Amidi. *An Autonomous Vision-Guided Helicopter*. PhD thesis, Dept of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, 1996.
3. J.Y. Bouguet. MATLAB camera calibration toolbox. In *TR*, 2000.
4. M. Bryant, D. Wettergreen, S. Abdallah, and A. Zelinsky. Robust camera calibration for an autonomous underwater vehicle. In *Proceedings of the 2000 Australian Conference of Robotics and Automation*, August 2000.
5. C. Charnley, G. Harris, M. Pike, E. Sparks, and M. Stephens. The droid 3d vision system - algorithms for geometric integration. Technical Report Tech. Rep. 72/88/N488U, Plessey Research Roke Manor, December 1988.
6. P. Corke. An inertial and visual sensing system for a small autonomous helicopter. *Journal of Robotic Systems*, 21(2):43–51, February 2004.
7. P.I. Corke, D. Strelow, and S. Singh. Omnidirectional visual odometry for a planetary rover. In *Proceedings of IROS 2004*, pages 4007–4012, 2004.
8. M. Dunbabin, P. Corke, and G. Buskey. Low-cost vision-based AUV guidance system for reef navigation. In *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, pages 7–12, April 2004.
9. M. Dunbabin, J. Roberts, Usher K., G. Winstanley, and P. Corke. A hybrid AUV design for shallow water reef navigation. In *Proceedings of the 2005 IEEE International Conference on Robotics & Automation*, April 2005.
10. R. Eustice, O. Pizarro, and H. Singh. Visually augmented navigation in an unstructured environment using a delay state history. In *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, pages 25–32, April 2004.
11. J. Lagarias, R. Reeds, and M. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.
12. P. Rives and J-J. Borrelly. Visual servoing techniques applied to an underwater vehicle. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pages 1851–1856, April 1997.
13. J. M. Roberts. *Attentive visual tracking and trajectory estimation for dynamic scene segmentation*. PhD thesis, University of Southampton, UK, 1994.
14. English S., C. Wilkinson, and V. Baker, editors. *Survey manual for tropical marine resources*. Australian Institute of Marine Science, Townsville, Australia, 1994.
15. J. Santos-Victor and G. Sandini. Visual behaviors for docking. *Computer Vision and Image Understanding*, 67(3):223–238, September 1997.
16. S. van der Zwaan, A. Bernardino, and J. Santos-Victor. Visual station keeping for floating robots in unstructured environments. *Robotics and Autonomous Systems*, 39:145–155, 2002.
17. L. Whitcomb, D. Yoerger, H. Singh, and J. Howland. Advances in underwater robot vehicles for deep ocean exploration: Navigation, control and survey operations. In *Proceedings of Ninth International Symposium of Robotics Research (ISRR'99)*, pages 346–353, October 9-12 1999.

Road Obstacle Detection Using Robust Model Fitting

Niloofer Gheissari¹ and Nick Barnes^{1,2}

¹ Autonomous Systems and Sensing Technologies, National ICT Australia
Locked bag 8001, Canberra, ACT 2601, AUSTRALIA

`ngheissari@groupwise.swin.edu.au`

² Department of Information Engineering, The Australian National University
`Nick.Barnes@nicta.com.au`

Summary. Awareness of pedestrians, other vehicles, and other road obstacles is key to driving safety, and so their detection is a critical need in driver assistance research. We propose using a model-based approach which can either directly segment the disparity to detect obstacles or remove the road regions from an already segmented disparity map. We developed two methods for segmentation: first, by directly segmenting obstacles from the disparity map; and, second by using morphological operations followed by a robust model fitting algorithm to reject road segments after the segmentation process. To test the success of our methods, we have tested and compared them with an available method in the literature.

1 Introduction

Road accidents have been considered as the third largest killer after heart disease and depression. Annually about one million people are killed and a further 20 million are injured or disabled. Road accidents not only cause fatality and disability, but also they cause stress, anxiety and financial side effects on people's daily life. In the computer vision and robotics communities, there have been various efforts to develop systems which assist the driver to avoid pedestrians, cars and road obstacles. However, road structure, lighting, weather conditions, and interaction between different obstacles may significantly affect the performance of these systems. Hence, providing a system that is reliable in a variety of conditions is necessary.

According to Bertozzi *et al.*, [4] the use of visible vision and image processing methods for obstacle detection in intelligent vehicles can be classified as motion based [11], stereo based [12], shaped based [3] and texture based [5] methods. For more details on the available literature, readers are referred to [10]. Among these different approaches, stereo-based vision have been reported as the most promising approach to obstacle detection [7]. The recent

works in stereo-based obstacle detection for intelligent vehicles include the Inverse Perspective Method (IPM) [2] and the u- and v-disparity map [9].

IPM relies on the fact that if every pixel in the image is mapped to the ground plane, then in the projected images obstacles located on the ground plane are distorted. This distortion generates a fringe in the image resulting from subtracting the left and right projected images and helps us to locate an obstacle in the image. This method requires the camera parameters and the base line to be known as a *a priori*. In fact, IPM is very sensitive to camera calibration accuracy. Furthermore, the existence of shadows, reflections or markings on the road may reduce the performance of this method. The other recent method in obstacle detection for intelligent vehicles is based on generating u- and v-disparity maps [9], which are histograms of the disparity map in the vertical and horizontal directions. An obstacle is represented by a vertical line in v-disparity while by a horizontal line in u-disparity. The ground plane can be detected as a line with a slope. Hence, techniques such as Hough Transform can be applied to detect obstacles. Obstacle detection using u- and v- disparity maps appear to outperform IPM [8], however they have other shortcomings. For example, the u- and v-disparity maps are usually noisy and unreliable. In addition, accumulating in the horizontal and vertical direction of the disparity map causes objects behind each other (or next to each other) be incorrectly merged. The other disadvantage of this method is that small objects or objects which are located in a far distance from camera tend to be undetected. This may occur due to line segments in these regions that are either too short to detect, or too long and so easily merged with other lines in the v- or u-disparity map.

To overcome the above problems, this paper presents two new obstacle detection algorithms for application in intelligent vehicles. Both algorithms segment the disparity map. The first algorithm is based on the fact that the obstacles are located approximately parallel to the image plane, and directly segments them using a robust model fitting method applied to the quantised disparity space. The second algorithm incorporates some simple morphological operations and then a robust model fitting approach to separate the road regions from the image. As this robust fitting method is only applied to a part of image, the computation time is low. Another advantage of our model based approach is that we do not require calibration information, which is in sharp contrast with methods such as IPM.

Note that for finding pedestrians and cars in a road scene, typically stereo data is used as a first stage, then fused with other data for classification. This paper addresses the first stage only, and is highly suitable for incorporation with other data at a later stage, or direct fusion with other cues.

2 Algorithm 1: Robust Model Fitting

This algorithm relies on the idea that a constant model can describe the disparity map associated with every obstacle approximately parallel to the image plane. This is a true assumption where objects:

1. have no significant rotation angle;
2. have rotation but are not too close to the camera; or,
3. have rotation but have no significant width.

Later we will show that, by assuming overlapping regions in our algorithm, we may allow small rotations about the vertical or horizontal axis. In the algorithm, we first apply a contrast filtering method to the image and remove areas of low contrast from the disparity map. It allows us to remove regions whose disparity map, due to the lack of texture, is unreliable. This contrast filtering method is described in Section 4. We then quantise the disparity space by dividing it to a number of overlapping bins of length g pixels. Each bin has $g/2$ pixels overlap with the next bin. This overlap can help to prevent regions being split across two successive bins. In our experiments we set $g=8$ pixels. This quantisation approach has some advantages; first we apply our robust fitting method to each bin separately and hence we avoid expensive approaches such as random sampling. Second, we take the quantisation noise of pixel-based disparity into account. Finally, it allows an obstacle to rotate slightly around the vertical axis or have a somewhat non-planar profile (such as a pedestrian). After disparity quantisation, we fit the constant model to the whole bin. We compute the constant parameter and the residuals. If the noise is Gaussian, the squared residuals will be subject to a χ square distribution with $n-1$ degrees of freedom and thus the scale of noise will be $\delta = \frac{\sum r_i^2}{n-1}$: where r_i is the residual of the i^{th} point. We compute the scale of noise and select the points whose corresponding residual is less than the scale of noise multiple by the significance level T (which can be looked up from Gaussian distribution table). These points are inliers to the constant model and thus do not belong to the road. Now we have a preliminary knowledge about the inliers/outliers. In the next stage we iteratively fit the model to the inliers, recompute the constant parameter with more confidence and compute the final scale of noise only using the inliers. We used 3 iterations in our experiments.

We now have a final estimation of the model parameter. However iteration has shrunken the inlier space. To create larger regions and simultaneously maintain our degree of confidence, we fit the final estimated model to the bin (including inliers and outliers) and reject outliers using the final scale of noise.

This above task gives us different sets of inliers of different depths that create a segmentation map. However, this does not guarantee the locality of each segment. To enforce the locality constraint we compute the regional maximum of the segmentation map, assuming that we are only interested in areas which are closer to us than the surrounding background. Finally a 4-

connected labelling operation provides us with the final segmentation map. As a post processing stage we may apply a dilation operation to fill the holes.

Figure 1-3 show the contrast filtering result mapped on the disparity map, the result of the 4-connected labelling operation on the segmented image (and the dilation) and the final result for frame 243. As can be seen from this figure, the missed white colour car (at the right side of the image) does not have enough reliable disparity data and thus is not detected as a separate region.

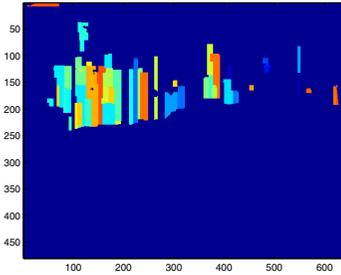


Fig. 1. The contrast filtering result mapped on the disparity map.

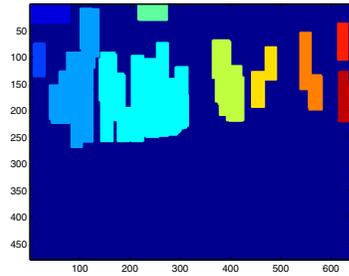


Fig. 2. The 4-connected labelling operation result

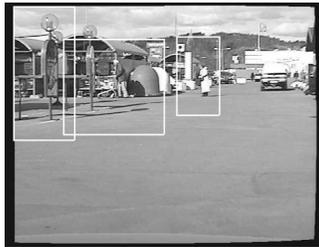


Fig. 3. Final results

3 Algorithm 2: Basic Morphological Operations

The second segmentation algorithm presented here is a simple set of morphological operations, followed by a road separation method. We first compute the edges of the disparity map. Again, we apply our contrast filtering method to the intensity image, and from the edge map we remove areas which have low contrast. We apply a dilation operation to thicken the edges. Then we fill the holes and small areas. We apply an erosion operation to create more distinct areas. To remove isolated small areas we use a closing operation next to an opening operation. Finally as a post processing step we dilate the resulting

region using a structural element of size 70×10 . This step can fill small holes inside a region and join closely located regions. This algorithm relies on the removal of road areas. An algorithm for this is explained below.

3.1 Road Separation

Assume that we are given the disparity map and an initial segmentation in the form of a set of overlapping rectangular regions. The camera parameters and the base line is assumed to be unknown, which is an advantage of our method over the existing methods. We aim at rejecting those regions which belong to the road. We assume the road plane to be piecewise linear. It can be easily proved that the disparity of pixels located on the road can be modelled by the following equation [6]: $d = \frac{B}{H} f_x (\frac{y}{f_y} \cos \alpha + \sin \alpha)$ where y is the image coordinate in the horizontal direction, H is the distance of camera from the road, B is the base line and α is the tilt angle of camera with respect to the road. The parameters f_x and f_y are the scaled camera focal length. Thus for simplicity we can write $d = ay+b$: where a and b are some unknown constant parameters.

That means we describe the road with a set of linear models, i.e., modelling the road as piecewise linear (any road that is not smooth and piecewise linear certainly is an obstacle). We fit the linear model to every segment in the image. We compute the parameters a , b and the residuals. If the noise is Gaussian, the squared residuals will be subject to a χ square distribution with $n-2$ degrees of freedom and thus the scale of noise will be $\delta = \frac{\sum r_i^2}{n-2}$: here r_i is the residual of the i^{th} point. We compute the scale of noise and select the points whose corresponding residual is less than the scale of noise as inliers. Since these points are inliers to the assumed road model, they are not part of an obstacle. Then we select the regions whose number of inliers is more than a threshold. This threshold represents the maximum number of road pixels which can be located in a region and that region be still regarded as an obstacle region. Again we apply the previously discussed robust model fitting approach to the inliers to estimate the final scale of noise and model parameters. We create a new set of inliers/outliers. We reject a region as a road region only if its sum of squared residuals is greater than the scale of noise. Once we make our final decision, we can compute the final road parameters if we require. We also can compute a reliability measure for each region based on its scale of noise and its number of outliers to the road model (obstacle pixels).

4 Contrast Filtering

If an area does not have sufficient texture, then the disparity map will be unreliable. To avoid such areas we have applied a contrast filtering method which includes two median kernels of size 5×5 and 10×10 . The sizes of these

pixels were chosen heuristically so that we ignore areas (smaller than 10×10) in which the contrast is constant. We convolved our intensity image with both median kernels. This results in two images I_1 and I_2 , in each of which, every pixel is the average of the surrounding pixels (with respect to the kernel size). We compute the absolute difference between I_1 and I_2 and construct matrix M , so that $M=I_1-I_2$. We reject every pixel i where $M_i < FTH$. The threshold FTH is set to be 2 in all experiments.

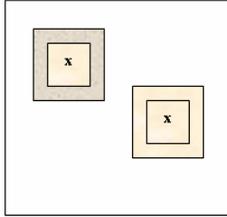


Fig. 4. If the contrast varies significantly between two embedded regions, then the filter results in a high value (e.g., the left embedded squares), while, for regions with no contrast the filter results in a low value (e.g., the right embedded squares).

5 Experimental Results

Within the ANU/NICTA intelligent vehicles project, we have a continuing project to develop pedestrian and obstacle detection, to run on our road vehicle, see Figure 5. For the purposes of this paper, we have applied our algorithm to a noisy image sequence containing pedestrians, cars and buildings. Here, we have provided a few samples of our results. We also have included the results of applying the stereo-vision algorithm reported in [8] on the same



Fig. 5. Inside the ANU/NICTA intelligent vehicle. Cameras to monitor the road scene and find obstacles appear in place of the rear-vision mirror.

image sequence. This algorithm, which uses the u - and v -disparity map, has been shown to be successful in comparison with other existing methods [8].

The example frames shown here were chosen to illustrate different aspects (strength and weakness) of both algorithms. We also compared the three methods quantitatively in figure 18. The computation time for both proposed methods is about one second per frame in a non-optimized Matlab implementation on a standard PC. We expect it to be better than frame rate in a C optimized implementation, and so comfortably real-time.

As the following results indicate all the algorithms may miss a number of regions. However, it has been observed (from figure 18) that the model based algorithm misses fewer regions and performs better. However, a drawback of this algorithm is that if the disparity map is noisy, and some obstacles may be rejected as outliers (in the robust fitting stage). This can be solved by assuming a larger significance level T . However, it may cause under-segmentation. In future work we plan to devise an adaptive approach to compensate for a poor and noisy disparity map.

The morphological algorithm is only applicable where the disparity map is sparse, otherwise for a dense disparity map, we will have a considerable under segmentation. In this case, using the model based algorithm is suggested.

Figures 6-8 show that the model based algorithm has detected all the obstacles correctly (in frame 8), while the morphological based algorithm has under-segmented the data, and the u - and v -disparity based algorithm only detected one obstacle.

As can be seen from figure 9-11, the model fitting based algorithm has detected all obstacles, but failed to segment a pedestrian from the white car (in frame 12). The morphological based algorithm has again missed the white car. In contrast, the u - and v - disparity based algorithm has only succeeded in detecting one of the pedestrians.

Figure 12-14, show that all of the different algorithm have successfully ignored the rubbish and the manhole on the road. The model fitting based algorithm has detected all obstacles except for the pedestrian close to the camera. The morphological based algorithm has again missed the small white car while it has successfully detected the pedestrians. In contrast, the u - and v - disparity based algorithm has only succeeded in detecting the pedestrian near to the camera.

The last example is frame 410 of the sequence. Figure 15-17 show that while the model based algorithm tends to generate a large number of different regions, the morphological operations based algorithm tends to detect more major (larger and closer) obstacles. The pedestrian has a considerable rotation angle and so the model based algorithm split the pedestrian across two regions. This can be easily solved by a post-processing stage. Both the u - and v -disparity and the model based algorithms miss the car at the right side of image. However, small obstacles at further distances, which are ignored by the u - and v - disparity based algorithm, are detected by the model based one. Furthermore, although the u - and v - disparity based algorithm generates more

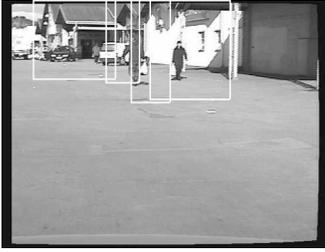


Fig. 6. Results of applying model based algorithm on frame 8.

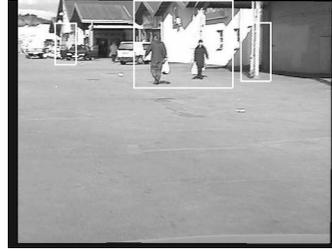


Fig. 7. Results of applying morphological based algorithm on frame 8.

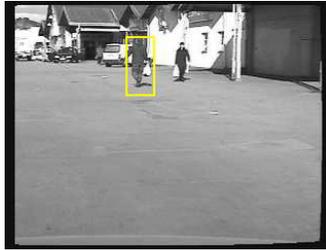


Fig. 8. Results of applying u- and v-disparity based algorithm in [8] on frame 8.

precise boundaries for the pedestrian, it generates a noisy segmentation. This may happen in all algorithms and is mainly due to noise in disparity. This is best dealt with using other cues.

5.1 Comparison Results

In figure 18 we show the results of applying the two algorithms to 50 successive images of a road image sequence. These 50 frames were chosen because all of them have four major obstacles, a reasonably high number in real applications. The ground truth results and also the results of applying the u- and v-disparity based algorithm [8] have been shown in different colors. Ground truth was labelled manually by choosing the most significant obstacles. Figure 18 clearly show that both proposed algorithms outperform the u- and v-disparity based algorithm. More importantly the model based method for obstacle detection has been more successful than the other two approaches. The complete sequence is available at:

<http://users.rsise.anu.edu.au/~nmb/fsr/gheissaribarnesfsr.html>.

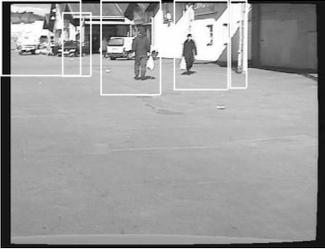


Fig. 9. Results of applying model based algorithm on frame 12.



Fig. 10. Results of applying morphological based algorithm on frame 12.

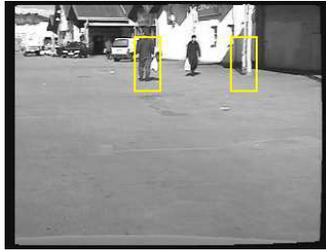


Fig. 11. Results of applying u- and v-disparity based algorithm frame 12.

Note that with careful tuning the u- and v-disparity based algorithm may generate better results. Despite its poor performance, this method has the advantage of generating a more precise bounding box.

6 Future Work

Although the results of our algorithms show better performance than using u- and v-disparity map, still there is a large space for improvement. The robust model based approach can be improved by using a smarter quantization method than the current one. The partitioning method of Bab-Hadiashar and Suter [1] can also be used to improve the results. In addition, we may use a model selection criterion to decide if a region is located on the road or on an obstacle approximately parallel to the camera image plane.

To reduce the false negatives, we will fuse the disparity segmentation result with other cues such as shape, texture or motion. We also will include a tracking method to track obstacles over the image sequence such as Kalman filtering or particle filtering.



Fig. 12. Results of applying model based algorithm on frame 170.



Fig. 13. Results of applying morphological based algorithm on frame 170.

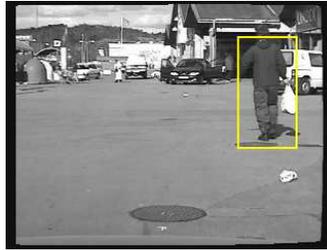


Fig. 14. Results of applying u- and v-disparity based algorithm frame 170.

Finally, as this algorithm only aims at detecting the obstacle region with no classification, later we will include a classification approach to decide whether an obstacle is a car, a pedestrian etc.

7 Conclusion

The main contribution of this paper is proposing a model based approach for obstacle detection in driving assistant applications. The first algorithm relies on the fact that a constant model can describe the disparity map associated with every obstacle approximately parallel to the camera image plane. We quantize the disparity space, use a robust model fitting method to estimate the constant model parameter, and compute the scale of noise that is used to partition the data. In the second algorithm that only can be applied to sparse disparity maps, we use some basic morphological operations to segment the data. Our main contribution here is not the segmentation algorithm itself but it is the way we separate the road data from our image. This road separation (or detection) method again is based on a model-based approach.

Both algorithms have been extensively tested and compared and have shown to be more successful than the u- and v- disparity segmentation al-

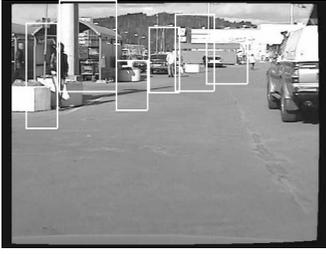


Fig. 15. Results of applying model based algorithm on frame 410



Fig. 16. Results of applying morphological based algorithm on frame 410

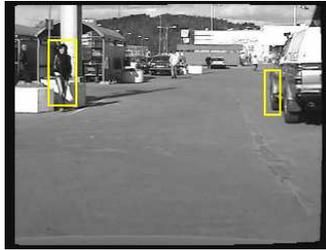


Fig. 17. Results of applying u- and v-disparity based algorithm frame 410

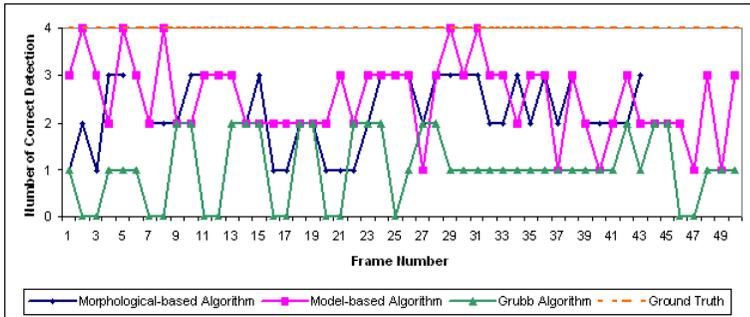


Fig. 18. Comparison of our two algorithms with the u- and v- disparity based algorithm.

algorithm. The model based algorithm misses fewer regions than the other algorithms. This indicates that the model based approach is effective for obstacle detection, and is worthy of further study to improve its performance further.

Acknowledgment

National ICT Australia is funded by the Australian Department of Communications, Information Technology and the Arts and the Australian Research Council through *Backing Australia's ability* and the ICT Centre of Excellence Program.

References

1. Bab-Hadiashar A., Suter D., Robust Segmentation of Visual Data Using Ranked Unbiased Scale Estimator, International Journal of Information, Education and Research in Robotics and Artificial Intelligence, ROBOTICA, volume 17, 649-660,1999.
2. Bertozzi, M., Broggi A., Fascioli A., Stereo Inverse Perspective Mapping: Theory and Applications, Image and Vision Computing Journal, 16(8), pp. 585-590, 1998.
3. Bertozzi, M., Broggi A., Fascioli A., and Sechi, M., Shape-based Pedestrian Detection, Proceedings of IEEE Intelligent Vehicles Symposium, pp. 215-220, Oct. 2000.
4. Bertozzi, M., Broggi, A., Grisleri, P., Graf, T., and Meinecke, M., Pedestrian Detection in Infrared Images, Proceedings of IEEE Intelligent Vehicles Symposium, pp. 662-667, June 2003.
5. Curio, C., Edelbrunner J., Kalinke T., Tzomakas, C., and Seelen W. von, Walking Pedestrian Recognition, IEEE Transactions on Intelligent Transportation Systems, vol. 1, pp. 155-163, Sep, 2000.
6. Franke, U. and Kutzbach, Fast Stereo based Object Detection for Stop and Go Traffic, Intelligent Vehicles Symposium, pp. 339-344 , 1996.
7. Gavrila, D. M., Giebel, J., and Munder, S., Vision-Based Pedestrian Detection: The PROTECTOR System, pp. 13-18, 2004.
8. Grubb Grant, Alexander Zelinsky, Lars Nilsson, Magnus Rible, 3D Vision Sensing for Improved Pedestrian Safety, Intelligent Vehicles Symposium(2004), pp. 19- 24, Parma Italy, June 2004.
9. Labayrade, R., Aubert, D., and Tarel, J.-P., Real Time Obstacle Detection in Stereovision on Non Flat Road Geometry Through "V-disparity" Representation, pp. 646-651, June 2002.
10. Sun, Z., Bebis, G., and Miller, R., On-Road Vehicle Detection Using Optical Sensors: A Review, Proceedings of IEEE Intelligent Transportation Systems Conference , Washington, D.C. USA, pp. 585-590, 2002.
11. Viola, P., Jones, M., and Snow, D., Detecting Pedestrians Using Patterns of Motion and Appearance, Proceedings of the International Conference on Computer Vision (ICCV), pp. 734-741, Oct. 2003.
12. Zhao, L. and Thorpe Charles E., Stereo- and Neural Network- Based Pedestrian Detection IEEE Transactions on Intelligent Transportation Systemes, vol. 1, pp. 148-154, Sep, 2000.

Real-Time Regular Polygonal Sign Detection

Nick Barnes¹ and Gareth Loy²

¹ National ICT Australia, Locked Bag 8001, Canberra, ACT 2601,
Department Of Information Engineering, The Australian National University
Nick.Barnes@nicta.com.au

² Computer Vision and Active Perception Laboratory, Royal Institute of
Technology (KTH), Stockholm, Sweden
gareth@nada.kth.se

Summary. In this paper, we present a new adaptation of the regular polygon detection algorithm for real-time road sign detection for autonomous vehicles. The method is robust to partial occlusion and fading, and insensitive to lighting conditions. We experimentally demonstrate its application to the detection of various signs, particularly evaluating it on a sequence of roundabout signs taken from the ANU/NICTA vehicle. The algorithm runs faster than 20 frames per second on a standard PC, detecting signs of the size that appears in road scenes, as observed from a camera mounted on the rear-vision mirror. The algorithm uses the symmetric nature of regular polygonal shapes, we also use the constrained appearance of such shapes in the road scene to the car in order to facilitate their fast, robust detection.

1 Introduction

Improving safety is a key goal of road vehicle technology. Driver support systems aim to improve safety by helping drivers react to changing road conditions. Although full automation of road vehicles may be achievable in the future, our research focusses on systems that can assist drivers immediately. Rather than replacing the driver, we aim to keep the driver in the loop, while supporting them in controlling the car.

Road signs present information to alert drivers to changes in driving conditions. Critical information signs, such as speed, give-way, roundabout, and stop give information that a driver must react to, as opposed to informational and directional signs. These signs appear clearly in the road scene, and are well distinguished. However, drivers may sometimes miss such signs due to distractions or a lack of concentration. This makes detecting critical information signs and making the driver aware of any they may have missed a key target for improving driver safety. The lack of driver awareness of a sign may be detected through a lack of response.

We have previously demonstrated the application of the radial symmetry operator [1] to detecting speed signs, demonstrating real-time performance [2].

Here detection took advantage of the circle that must appear on Australian speed signs. The radial symmetry algorithm is a shape detector, based on image gradient, and so is robust to varying lighting conditions and occlusions, or the incomplete appearance of edges due to sign degradation under weather conditions. This detection algorithm was coupled with recognition to allow a full system that reported the current speed limit to the driver in real-time.

The regular polygon detector is a more general algorithm than radial symmetry that is able to detect all regular polygons including triangular, square, and octagonal shapes [3]. General regular polygons may be detected in an image up to a similarity transform. All the properties of the radial symmetry algorithm are maintained in terms of robustness to illumination changes, occlusion, and poor edges. In this paper, we adapt this algorithm specifically to real-time road sign detection. By only examining gradients that could be part of the road sign being sought, we are both able to simplify the detection process without significantly impacting robustness. This allows the detection to run at less than 50 ms per image on a standard PC installed in the vehicle. We demonstrate this performance on a real road sequence, showing robust and effective detection.

This class of regular polygon detectors has complexity $\mathcal{O}(Nkl)$, where l is the maximum width of the segments of the shape, k is the number of scales that are being considered, and N is the number of image pixels. Note that for small shapes k and l are small numbers. The generalised Hough transform [4] is a general algorithm that can perform the same function, however, even modern hardware-based implementations take multiple seconds to recognise a single shape [5]. This improved computational performance is the benefit of specialised algorithms for detection of known features.

Other work in perceptual grouping [6] takes a similar approach in terms of finding local support for shapes. However, this work does not use pixel-based gradient information, and works at the level of edge segments for gradient. A complexity of k^2 was reported where k is the number of edge segments, however for a cluttered image, if the segment size is one pixel, k may be of order of the size of the image.

The regular polygon shape detector class of algorithms is specialised for real-time performance by exploiting the nature of regular polygon-like shapes that have a defined centre point, using it as the voting centre. This makes the algorithm robust to pixels that are missed due to occlusion, poor gradient direction estimates or broken edges, as the vote at the centre point will still be high. The regular polygon detectors are parametric in the formulation, and can be easily and efficiently applied in situations where constraints are available from the embodiment of the vision system within, such as the appearance of a road sign to a car. In this paper, we focus specifically on the aspect of adapting the detector for road sign detection.

This detector improves on previous sign detection results by facilitating fast, robust detection that can reduce the region of interest for sign recognition to only a few pixels, returning the position, centroid, scale and shape

of candidate signs. Thus, few pixels require examination for recognition, and the sign size and shape parameters are known. This allows computationally expensive recognition as processing is well-targetted, so comparatively little computation is needed to assess a candidate. The enhancements detailed in this paper reduce the number of false positive sign detections by ignoring all gradient edges that are not at an angle that could be part of a valid road sign.

2 Background

Road sign recognition research has been around since the mid 1980's. A direct approach is to apply normalised cross correlation to the raw traffic scene image. This brute force method is computationally prohibitive, but can be eased somewhat by approaches such as simulated annealing [7], or applying templates to an edge image of the road scene [8]. However, these methods are still computationally intensive and so unsuitable for a real-time system.

Many approaches have introduced separate stages for sign detection and classification of sign type (e.g., [9, 10, 11]). We argue that this is an effective means of managing computation for even a small number of sign types if the detection algorithm has low computational cost, facilitating real-time operation. In this manner, computationally intensive classification is only performed on a small fraction of the input stream.

Colour segmentation is the most common method for detection. Typically, this is based on the assumption that the wavelength that arrives at the camera from a traffic sign is invariant to the intensity of incident light. This assumption usually manifests in the statement that HSV (or HSI) space is invariant to lighting conditions [11]. A great deal of the research in this area exploits a detection stage based on this assumption (e.g., [11, 12, 9, 13, 14, 15, 16]), either finding the signs, or eliminating much of the image from further processing. However, the camera image is *not* invariant to changes of incident light chromaticity. Such variance occurs under different conditions, such as direct sunlight, heavy cloud, smog, and under headlights at night. Further, the colour changes as signs fade over time.

Another approach to detection is *a priori* assumptions about image formation. At its simplest, one can assume that the road is approximately straight, so much of the image can be ignored. Combining such assumptions with colour segmentation, Hsu and Huang [16] look for signs in only a restricted part of the image. However, such assumptions can break down on curved roads, or if the road is not planar. A more sophisticated approach is to use some form of detection to facilitate scene understanding, and thus eliminate a large region of the image. For example, Piccoli *et. al* [12] suggest large uniform regions of the image correspond to the road and sky, and thus signs are only likely to appear in the region alongside the road and below the sky. However, this is inadequate in more difficult road scenes, for example a scene with trees overhead casting shadows across the road. They also suggest ignoring one side of

the image as signs will only come up on one side, however, this is not always the case, for example see Figure 3, where the sign is in the centre of the road.

In this paper we present an adaptation of the regular polygon detection algorithm to real-time detection of road signs. The detector uses gradient elements to detect signs based on shapes, detecting triangle, square, diamond, octagonal, and circular signs. The adaptation of the regular polygon detector that we present can detect all signs in an image in 50 ms at all the sizes that are practical for our driver assistance system vehicle platform.

3 The Structure of the Road Environment

There is much possible variation in sign appearance. Throughout the day, and at night time, lighting conditions can vary enormously. A sign may be well lit by direct sunlight, or headlights, or on the other hand it may be completely in shadow on a bright day. Further, heavy rain may blur the image of the sign. Ideally, signs have clear colour contrast, but over time they can become quite faded, but are still quite clear to drivers. Although signs appear by the road edge, this may be far from the car on a multi-lane highway — to the left or right, or very close on a single lane exit ramp — and whilst signs are generally a standard distance above the ground, they can also appear overhead or on temporary roadworks signs at ground level. With this nature, it is not simple to restrict the possible viewing positions of a sign within the image. By modelling the road [17], it may be possible to dictate parts of the image where a sign cannot appear, but road modelling has its own computational expense, and, as discussed previously, colour-based methods are not robust. However, the roadway is well structured. Critical signs, including give way, stop and roundabout signs have a highly regulated appearance. Unless the sign has been tampered with, signs will appear upright and approximately orthogonally to the road direction. Finally, signs are always placed to be easily visible, so the driver can easily see them without having to look away from the road. In this paper we assume that the signs are correctly oriented, however, with minor accidents, signs can occasionally appear tilted.

The regular polygon algorithm searches for near regular polygonal features. As a legal critical information signs must always appear orthonormal to the direction of the road, provided the system camera points in the direction of vehicle motion, the surface of signs will be approximately parallel to the image plane and so the polygonal shape of these signs will appear undistorted. On a rapidly curving road it may be that the sign only appears parallel to the image plane briefly, but this will be when the vehicle is close to the sign, so it will appear large in the image. If we are processing images at 20 Hz, and we are able to recognise a sign reliably from only a small number of frames then generally we are safe to assume that the sign will be approximately parallel to the image plane for many images.

4 Adapting Regular Polygon Detection to Real-Time Sign Detection

Detecting regular polygons has been shown to be a powerful means of locating triangular, square and octagonal road signs in images [3]. The method used was an extension of the radial symmetry transform [1], and utilised the intrinsic symmetry of equiangular regular polygons to detect these shapes independently of orientation. However, in the case of road sign detection the orientation of the target is known *a priori*, and thus orientation independence is not required. This section demonstrates how the introduction of an orientation constraint can be applied to the algorithm to: reduce the complexity; and increase the speed to real-time levels.

For the sake of completeness this section summarises the base part of the algorithm described in [3]. This sets the basis for the adaptation to road scenes and the real-time implementation described in the remainder of the section.

4.1 Review of Regular Polygon Detection

The existing algorithm detects the centroids of n -sided regular polygons in greyscale images as follows. Firstly the image gradient vector field is computed, all elements with magnitudes under a predefined threshold are set to zero, and the resulting vector field is denoted \mathbf{g} . For each radius under consideration, a vote image O_r is computed of the same size as the input image, and initialised to zero. Each non-zero element of \mathbf{g} is considered in scan-line order, and the shape centroid locations voted for and against by this gradient element are computed. For a single gradient element $\mathbf{g}(\mathbf{p})$ these are given:

$$L_+ = \{L(\mathbf{p}, m) | m \in [-w, w]\}, \quad (1)$$

$$L_- = \{L(\mathbf{p}, m) | m \in [-2w, -w] \cup (w, 2w]\}, \quad (2)$$

respectively. Here $L(\mathbf{p}, m)$ describes a line of pixels in front of and behind the gradient element a distance r away, as shown in Figure 1, and is given by:

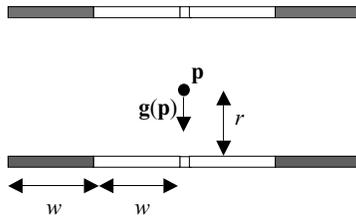


Fig. 1. Lines of pixels voted for by gradient element $\mathbf{g}(\mathbf{p})$, light lines indicate positive votes, dark lines negative votes.

$$L(\mathbf{p}, m) = \mathbf{p} + \text{round}(m\bar{\mathbf{g}}(\mathbf{p}) \pm r\mathbf{g}(\mathbf{p})), \quad (3)$$

where $\bar{\mathbf{g}}(\mathbf{p})$ is a unit vector perpendicular to $\mathbf{g}(\mathbf{p})$, m is an integer and

$$w = r \tan \frac{\pi}{n}. \quad (4)$$

Figure 2 shows the different voting lines for the same gradient element when searching for different shapes, the white lines show positive votes L_+ and the dark lines are the negative votes L_- . The negative votes are included to attenuate the response of straight lines too long to belong to the target shape. As each significant (non-zero) element of \mathbf{g} is considered the total votes for each pixel are accumulated in the vote image \mathbf{O}_r .

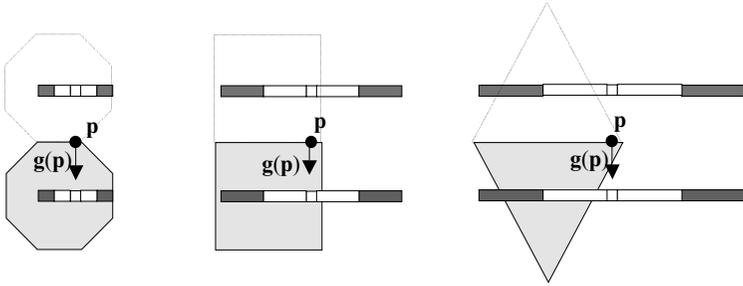


Fig. 2. Voting lines associated with a gradient element for different shapes.

In [3] a second vote image is constructed called the n -angle image. This votes for angles to ensure that the shapes detected have the required number, and does this in such a way as to facilitate the detection of these shapes at any orientation simultaneously. The n -angle image is a key component and accounts for a substantial portion of the computational time required. However, it is not required if the orientation of the target is known *a priori*.

4.2 Detecting Regular Polygons of Known Orientation

If the orientation of the targeted shape is known, the search can be targeted much more specifically, and a number of speedups and simplifications can be applied to both reduce the complexity and increase the performance of the method originally proposed in [3].

The orientation together with the number of sides of a regular polygon provides a constraint on the angles of gradient elements that are likely to occur at the edges of the shape. Specifically, for an n -sided figure with orientation θ , the gradient orientations likely to occur around this shape are given by



Fig. 3. Images from the start (52), middle (75), and end (99) of the sequence showing the roundabout sign used in this paper.

$$\angle \mathbf{g}(\mathbf{p}) = \left(\theta + \frac{2k\pi}{n} \pm \epsilon \right) \bmod \pi, \text{ for } k \text{ an integer,} \quad (5)$$

where ϵ is the angular tolerance on detected shape orientation with respect to that expected. This allows for some variation in the angle of real signs, and for some error in gradient values. Note that ϵ was actually a few degrees. Note that this method is equivalent to the previous algorithm, with *a priori* information replacing the second vote image to apply joint orientation selectivity. It will also remove any triangular candidates that are not oriented as signs, and so remove this noise from the image.

Only gradient elements satisfying the orientation constraint in Equation 5 need be considered when computing O_r . This both reduces the computation required to compute O_r and implicitly performs the same task that the n -angle image did in the unconstrained case, namely attenuating responses for votes whose angular spacing is not consistent with that of the target shape.

4.3 Embodied Vision Issues

In practice, we process a continuous image stream, not single images. Between two images at 20 fps, only slight motion will occur. Accidental features are common, where two image features align coincidentally, such as alignment of multiple edges to form a regular polygon. We may partly deal with this issue by requiring that a detected regular polygon is present for t images, does not move far between frames, and that its radius does not decrease in size over time, and only increases by an amount appropriate for the vehicle speed. Further, as there will be artefacts at multiple radii for a single sign, we apply non-maximal suppression.

Further, the size that the sign will appear in the camera image mounted in the car is constrained by the closest distance that it will be from the car, and the camera focal length. There is also no point in detecting signs that are smaller than the minimum size required to classify the sign.



Fig. 4. Inside the ANU/NICTA intelligent vehicle. Cameras to monitor the road scene appear in place of the rear-vision mirror.

5 Real-Time In-Vehicle Implementation

We integrated regular polygonal sign detection into the ANU/NICTA Intelligent vehicle, see Figure 4. The algorithm was implemented in C++. The maximum radii of the largest circle that could be inscribed on the triangle was 12 pixels, and four radii was sufficient for the visible range of give way signs as they appeared from the vehicle. For a 320 x 240 image, the detection of four radii algorithm was able to run at 20 fps, running in less than 50 ms per frame, on a 3Mhz dual processor PC. This is a vast improvement over the approximately 1 fps performance of the original algorithm [3]. In previous work we found that classification using normalised cross correlation was able to run in less than 1ms per candidate, and so there is sufficient time left for each frame to classify several candidates. As discussed in [18], the threshold of gradients was set to only take the top 20% of the gradients in the image, improving the signal to noise ratio and enhancing computational speed.

6 Results

Previously, we ran the detector on a series of signs taken from the internet with the general algorithm. Images from this set were rerun with the new adaptation of the algorithm to demonstrate that it has not lost any generality for road scenes. The images and their regular polygon transform for the correct radius are shown in Figure 5. The results are quite comparable with the full algorithm, and are less noisy due to the exclusion of gradients that are irrelevant to the constrained orientation case of road signs.

We took a real image sequence from the cameras mounted in the vehicle. The sequence consists of 100 images. The sign is large enough to be classified starting from frame 52, until it falls outside the image in frame 99. Over this period the vehicle was travelling at approximately 60 kmph. This sequence was run through the complete system without any specific adjustments to the sequence itself. The first, last, and mid images in the sequence are shown

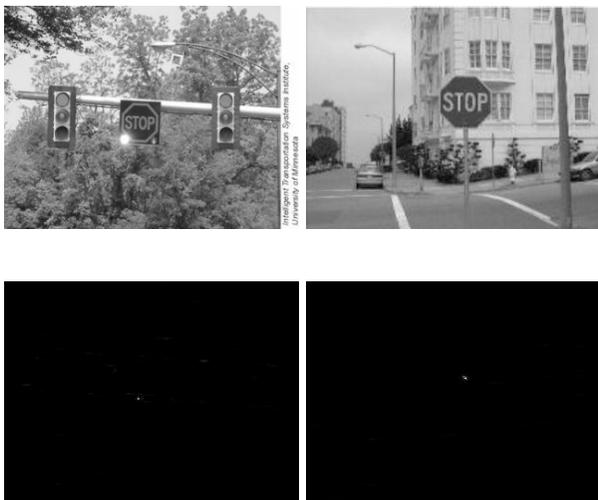


Fig. 5. Images of two stop signs from the internet, and corresponding regular polygon transform image using the algorithm presented for radius 20, and 28 respectively. Although not completely clear to eye in this reproduction, in both cases the sign is detected as the highest peak in the image.

in Figure 3. Figure 6 shows the vote images that were produced for the mid image in the sequence. At the position of the sign, the increasing radius can be seen as the lines produced by voting move towards, and then past the centre of the sign, the peak is across (b) and (c). For this image, the best radius was at (232,52) from top left. For this sequence, we took the best candidate in each frame for each radius, and checked if it was in the best ten candidates at each radius for the next t images. For the sequence, the frame by frame results for different t are shown in Figure 7. Figure 8 shows a summary of the number of times the sign was correctly detected during the sequence, and the number of false positives, given different values for t . Regardless of the detection rates over the sequence, it can be clearly seen that the sign is easily detected many times over the sequence. Also, with this small number of false positives, the recognition part of the system will be required to investigate a maximum of four image locations in any frame. Figure 9 shows the false positive rates for the images where the sign was too small to be detected. We expect that this can be improved by computationally simple post-processing of detected candidates. In initial trials with normalised cross correlation-based recognition, the sign was recognised 17 times in the sequence. Although this shows the sign being recognised many times while it is visible, the literature would suggest better recognition rates can be achieved with tuning.

In previous work round signs [2], we found that false positive repeat signs were rare for circular features. As can be seen from Figure 8 this was less the

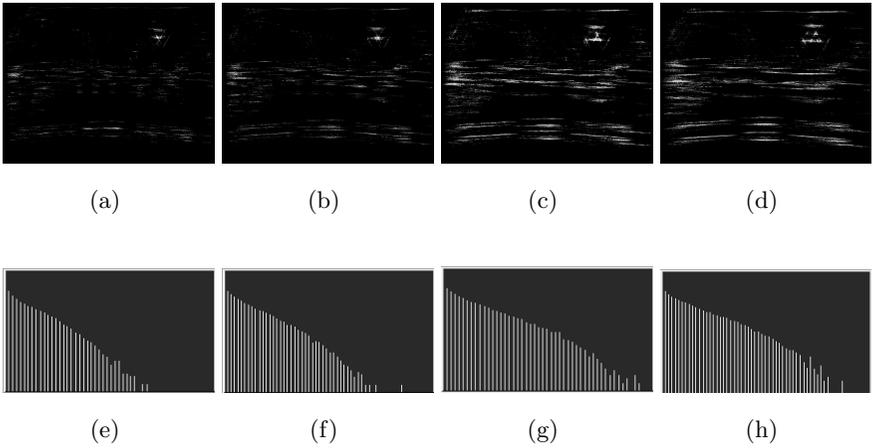


Fig. 6. The regular polygon detection image for image 76 shown above, for radii of (a) 6, (b) 8, (c) 10, and (d) 12. The images underneath show vote histograms. (b) and (f) show the correct radius with a single clear peak of votes at the sign, with few other candidates with a close number of votes.

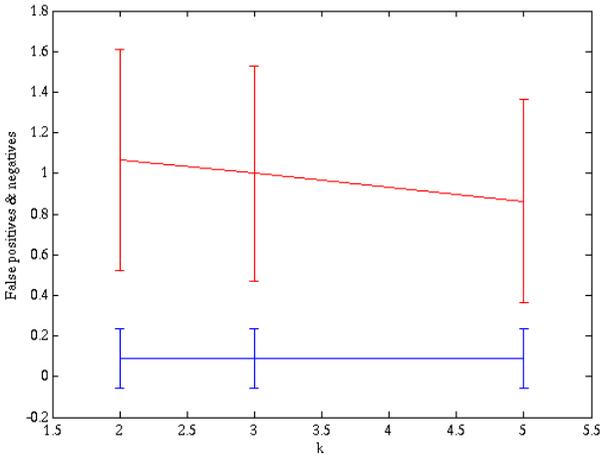


Fig. 7. Average and half standard deviation of false positives (top) and false negatives (bottom), given number of frames t that the sign is required to be present. Note that the error bars are set to one half of the standard deviation for clarity.

case for triangular signs. This is to be expected, as it is not uncommon for a multiple edges to align in an urban road-scene. However, as the number of expected edges increases towards the limit of a circle they will become less common as this requires more coincidentally placed edges.

| numframes | times correct | false positives |
|-----------|---------------|-----------------|
| 2 | 44 | 47 |
| 3 | 44 | 44 |
| 4 | 44 | 38 |
| 5 | 44 | 37 |

Fig. 8. For the number of frames the sign is required to be present, the number of correct classification and the number of false positives across the whole 48 frames where the sign was detectable.

| t number of frames | average false positives |
|----------------------|-------------------------|
| 2 | 3.01 |
| 3 | 2.80 |
| 4 | 2.46 |
| 5 | 2.26 |

Fig. 9. Number of false positives per frame versus t for the 52 frames where the sign was not large enough to be detected.

7 Conclusion

We have adapted the regular polygon algorithm for road sign detection. By constraining the gradient image voting, we were able to simplify the algorithm in terms of computation, and process images in less than 50 ms. The detection results over sample images show no significant loss of detection signal with respect to the original regular polygon detection algorithm. New image sequence results demonstrate stable performance at real time, and reliable detection, with sufficiently few false positives to support real-time classification.

Acknowledgements

National ICT Australia is funded by the Australian Department of Communications, Information Technology and the Arts and the Australian Research Council through *Backing Australia's ability* and the ICT Centre of Excellence Program. The support of the STINT foundation through the KTH-ANU grant IG2001-2011-03 is gratefully acknowledged.

References

1. G. Loy and A. Zelinsky, "Fast radial symmetry for detecting points of interest," *IEEE Trans Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 959–973, Aug. 2003.
2. N. Barnes and A. Zelinsky, "Real-time radial symmetry for speed sign detection," in *Proc IEEE Intelligent Vehicles Symposium*, Parma, Italy, 2004.

3. G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system," in *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems.IROS2004*, 2004, in press.
4. D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.
5. R. Strzodka, I. Ihrke, and M. Magnor, "A graphics hardware implementation of the generalized hough transform for fast object recognition, scale and 3d pose detection," in *Proc 12th International Conference on Image Analysis and Processing*, Mantova, Italy, 2003, pp. 188–193.
6. G. Guy and G. Medioni, "Inferring global perceptual contours from local features," *International Journal of Computer Vision*, vol. 20, no. 1-2, pp. 113–33, Oct. 1996.
7. M. Betke and N. Makris, "Fast object recognition in noisy images using simulated annealing," A.I. Lab, M.I.T., Cambridge, Mass, USA, Tech. Rep. AIM-1510, 1994. [Online]. Available: citeseer.nj.nec.com/betke94fast.html
8. D. M. Gavrila, "A road sign recognition system based on dynamic visual model," in *Proc 14th Int. Conf. on Pattern Recognition*, vol. 1, Aug 1998, pp. 16–20.
9. P. Paclik, J. Novovicova, P. Somol, and P. Pudil, "Road sign classification using the laplace kernel classifier," *Pattern Recognition Letters*, vol. 21, pp. 1165–1173, 2000.
10. J. Miura, T. Kanda, and Y. Shirai, "An active vision system for real-time traffic sign recognition," in *Proc 2000 IEEE Int Vehicles Symposium*, Oct 2002, pp. 52–57.
11. L. Prieze, J. Klieber, R. Lakmann, V. Rehrmann, and R. Schian, "New results on traffic sign recognition," in *Proc. Intelligent Vehicles Symposium*. Paris: IEEE Press, Aug. 1994, pp. 249–254. [Online]. Available: citeseer.nj.nec.com/prieze94new.html
12. G. Piccioli, E. D. Micheli, P. Parodi, and M. Campani, "Robust method for road sign detection and recognition," *Image and Vision Computing*, vol. 14, no. 3, pp. 209–223, 1996.
13. B. Johansson, "Road sign recognition from a moving vehicle," Master's thesis, Centre for Image Analysis, Swedish University of Agricultural Sciences, 2002.
14. C. Y. Fang, C. S. Fuh, S. W. Chen, and P. S. Yen, "A road sign recognition system based on dynamic visual model," in *Proc IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, 2003, pp. 750–755.
15. D. G. Shaposhnikov, L. N. Podladchikova, A. V. Golovan, and N. A. Shevtsova, "A road sign recognition system based on dynamic visual model," in *Proc 15th Int Conf on Vision Interface*, Calgary, Canada, 2002.
16. S.-H. Hsu and C.-L. Huang, "Road sign detection and recognition using matching pursuit method," *Image and Vision Computing*, vol. 19, pp. 119–129, 2001.
17. R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation," in *Proc IEEE Int Vehicles Symposium*, June 2002.
18. N. Barnes, "Improved signal to noise ratio and computational speed for gradient-based detection algorithms," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA '96)*, Barcelona, Spain, April 2005.

Distinctness Analysis on Natural Landmark Descriptors

Kai-Ming Kiang¹, Richard Willgoss², and Alan Blair³

¹ School of Mechanical and Manufacturing Engineering, University of New South Wales, NSW, 2052, Sydney, Australia kai-ming.kiang@student.unsw.edu.au

² School of Mechanical and Manufacturing Engineering, University of New South Wales, NSW, 2052, Sydney, Australia r.willgoss@unsw.edu.au

³ School of Computer Science and Engineering, University of New South Wales, NSW, 2052, Sydney, Australia blair@cse.unsw.edu.au

Autonomous navigation using natural landmarks in an unexplored environment is a very difficult problem to handle. While there are many techniques capable of matching pre-defined objects correctly, few of them can be used for real-time navigation in an unexplored environment. One important unsolved problem is to efficiently select a minimum set of usable landmarks for localisation purposes. This paper presents a method which minimises the number of landmarks selected based on texture descriptors. This enables localisation based on only a few distinctive landmarks rather than handling hundreds of irrelevant landmarks per image. The distinctness of a landmark is calculated based on the mean and covariance matrix of the feature descriptors of landmarks from an entire history of images. The matrices are calculated in a training process and updated during real-time navigation.

1 Introduction

Autonomous navigation in an unexplored environment is more challenging than in a controlled environment. In particular, underwater environments are mostly unexplored and do not have GPS access. Therefore navigation is generally based on methods such as Simultaneous Localization and Mapping (SLAM) [1] [2]. Most existing SLAM algorithms rely on artificial landmarks which do not exist in an unexplored environment. Recently, methods have been developed for extracting natural landmarks with representations that are invariant to scaling, distortion and perspective. Most of these methods select landmarks based on local properties of points, such as extracting the extrema [3] [4] or corner features [5]. The surrounding properties of these points are then analysed and converted to a vector of feature descriptors. These methods can efficiently select invariant natural landmarks from each image, so that

the same landmarks can be picked up under different geometric or lighting conditions from different images.

However, such methods tend to generate hundreds of landmarks per image. For real-time SLAM applications, it is computationally infeasible to compare landmarks from the current image against a database of all landmarks previously seen. SLAM does not only require a method for selecting natural landmarks that are invariant, but also requires selection of a small enough set of distinctive landmarks for computing the similarity between landmarks. A method for selecting distinctive landmarks that is both economical and efficacious is described in this paper.

2 Background

Scale Invariant Feature Transformation (SIFT) [4] is a method which has received much attention recently for its robustness in representing landmarks. It analyses the local gradients of the extrema extracted using Difference of Gaussian (DOG) filters [6]. Its descriptors are claimed to be invariant under changes in scale, rotation, shift and illumination conditions.

We have previously devised another method of representing landmarks based on DOG and frequency distribution analysis [3] that could potentially provide more robust descriptors than those based on gradient properties because the frequency properties used are usually less sensitive to noise.

Corner-based approaches, such as the Harris Matrix [7], claim to be also invariant under affine transformation [5]. Other methods including phase congruency [8], wide baseline stereo matching [9], intensity transformation [10] and steerable filters [11], are also designed to provide invariant descriptors. Some comparison of these techniques has been reported [12].

All of the methods mentioned above can be described in two main steps. Firstly, select interest points based on local properties. Secondly, analyse and represent the local properties of interest points by descriptors. The reason that methods mentioned so far tend to generate hundreds of landmarks per image is because the selection process occurs prior to descriptor transformation and is therefore based only on raw image data. The main motivation of distinctness analysis presented in this paper is to have a further selection process based on the landmarks represented by descriptors i.e. a post-descriptor selection process.

3 Distinctivness Analysis

The question arises as to how a few relevant landmarks out of a potentially large set should be remembered. In Figure 1, it would be best to remember the center object because it is the most distinctive among the set. If one remembers any of the other objects, which are similar to each other, it will be

hard to distinguish between them later on. The algorithm should maximize the probability of recognizing and localizing correctly, based on the features of just a few chosen landmarks.

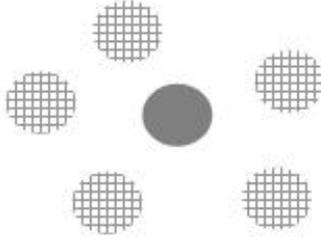


Fig. 1. Simple diagram of a distinctive object among other objects

3.1 Mathematical Distinctness

Mathematically speaking, if each landmark is represented by descriptors using a method noted in Section 2, each landmark becomes a feature vector of descriptors in parametric space. Distinctness can be judged from analyzing these vectors. The general philosophy of distinctness selection is to preserve a set of parameters that appear less frequently whilst deleting those that appear more frequently. If we consider all the feature vectors of landmarks as containing random variables, the probability of appearance for each of them can then be calculated by assuming a Gaussian distribution of the vectors using the formula [13]:

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^m \det(\mathbf{C})}} \times \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^t \mathbf{C}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\} \quad (1)$$

where:

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j \quad (2)$$

and

$$\mathbf{C} = \frac{1}{n-1} \sum_{j=1}^n (\mathbf{x}_j - \boldsymbol{\mu}) \cdot (\mathbf{x}_j - \boldsymbol{\mu})^t \quad (3)$$

where n = the number of landmarks.

A distinctness selection can then be made on the basis that the lower the probability, the more distinctive a landmark is judged to be.

3.2 Global Distinctness

Since distinctness selection is a process of minimizing the number of landmarks, the selection criteria must select landmarks consistently from a variety of images. The distinctness of a landmark must have a global meaning for it to be called distinctive i.e. any possible matches should pick out landmarks that are distinctive across images rather than within a particular image.

Referring to Equation 1, the distinctness of a landmark is calculated based on a mean vector $\boldsymbol{\mu}$ and a covariance matrix \mathbf{C} . To obtain these two matrices, the sample feature vectors must be selected over a wide range of images of the environment. However, remembering all the sampled feature vectors from each image can accumulate to a huge database. This is avoided because the mean and covariance are updated on every image without the need for recalculation later on.

Let us denote the mean and covariance for the global distinctness by $\boldsymbol{\mu}_t$ and \mathbf{C}_t respectively and those for the current image by $\boldsymbol{\mu}_c$ and \mathbf{C}_c . Then $\boldsymbol{\mu}_c$ and \mathbf{C}_c can be calculated from Equations 2 and 3; assuming $\boldsymbol{\mu}_t$ and \mathbf{C}_t have been initialised, they can be updated using the formula:

$$\boldsymbol{\mu}_t = \lambda \boldsymbol{\mu}_{t-1} + (1 - \lambda) \boldsymbol{\mu}_c \quad (4)$$

where λ is the innovation factor, which determines how much the system relies on history versus new data.

\mathbf{C}_t is calculated on the following formula:

$$C_{t(x,y)} = E(XY)_t - \mu_{t(x)}\mu_{t(y)} \quad (5)$$

where $E(XY)$ is the expectation value of the product of two dimensions X and Y, which can be calculated from:

$$E(XY)_t = \lambda E(XY)_{t-1} - (1 - \lambda)E(XY)_c \quad (6)$$

$E(XY)_{t-1}$ and $E(XY)_c$ can be obtained by rearranging Equation 5 using $E(XY)$ as the subject with appropriate $\boldsymbol{\mu}$ and \mathbf{C} matrices.

$\boldsymbol{\mu}_t$ and \mathbf{C}_t can be updated iteratively using $\boldsymbol{\mu}_c$ and \mathbf{C}_c . To initialize $\boldsymbol{\mu}_t$ and \mathbf{C}_t , they are assigned equal to $\boldsymbol{\mu}_c$ and \mathbf{C}_c for the first input image. $\boldsymbol{\mu}_t$ and \mathbf{C}_t require the system to run over a series of images in order to have confidence in global distinctness. A practical solution is to take a safe walk in the environment of interest e.g. move forward a few steps then move backward a few steps, before using the data for exploration into an unexplored environment.

3.3 Probability of Similarity

Once distinctive landmarks have been extracted, they are compared to form a judgment on how likely any two of them correspond to the same landmark.

This involves calculating the probability of similarity between two selected landmarks from different images.

Each landmark is extracted and converted into a feature descriptor i.e. a p -dimensional vector, which is subject to sources of randomness. Firstly there is random noise from the sensors. Secondly, the descriptor expression is itself a simplified representation of the landmark. Lastly the two images being compared could be viewing the landmark from a different perspective, which causes geometric distortion. Therefore, each landmark can be considered as a single sample of the observing object.

In making inferences from two landmarks in two different images, it is in principle a standard significance test. However, comparison is only made between two single samples. For this reason, the ANOVA test (The Analysis of Variance) cannot be used because the sample size required should be large.

For multidimensional vector comparison, the χ_v^2 (Chi-Squared) distribution test is appropriate. Chi-Squared distribution is a combined distribution of all dimensions which are assumed to be normally distributed. It includes an additional variable v describing the degrees of freedom. Details can be found in [13].

In multidimensional space, the χ_v^2 variable is defined by:

$$\chi_v^2 = N(\bar{\mathbf{x}} - \bar{\mathbf{y}})^t \Sigma^{-1} (\bar{\mathbf{x}} - \bar{\mathbf{y}}) \tag{7}$$

where:

$\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are the mean of the measurements of X and Y respectively;

Σ is the covariance matrix of noise;

N is a function related to the sample size of the two measurements.

Since our sample size is one, then $N = 1$, $\bar{\mathbf{x}} = \mathbf{x}$ and $\bar{\mathbf{y}} = \mathbf{y}$. Equation 7 simplifies to:

$$\chi_v^2 = (\mathbf{x} - \mathbf{y})^t \Sigma^{-1} (\mathbf{x} - \mathbf{y}) \tag{8}$$

If the noise of each dimension is independent of the other, the inverse covariance is a diagonal matrix and hence can be further simplified to:

$$\chi_v^2 = \sum_{i=1}^p \frac{(x_i - y_i)^2}{\sigma_i^2} \tag{9}$$

where p is the number of dimensions of \mathbf{x} .

Since \mathbf{x} contains p independent dimensions, then the degree of freedom v is p not $(p - 1)$ as usually defined for the categorical statistic. Also $\sigma_i = \sqrt{2}\sigma$, where σ is the standard deviation for a single random variable on each dimension.

With χ_v^2 and v obtained, the probability of similarity is defined to be equal to the integrated probability at the χ_v^2 value obtained. The integrated probability of Chi-Square distribution can be found in statistical tables.

4 Experimental

In this section, experiments were conducted on a series of sub-sea images (courtesy of ACFR, University of Sydney, Australia). The configuration was set such that the camera was always looking downwards on the sea floor. This configuration minimised the geometrical distortion caused by different viewpoints.

4.1 Initial Test of the Algorithm

For this experiment, the algorithm was written in Matlab V6.5 running on a PC with a P4 2.4GHz processor and 512Mb of RAM.

To demonstrate how the distinctness analysis algorithm worked, a typical analysis is now explained in detail. In the following example, we have trained the distinctness parameters μ_t and C_t over 100 images from the series. The texture analysis described in [3] generated invariant landmarks on two particular images shown in Figure 2 which consist of partially overlapping regions.

The distinctness analysis described in Section 3 was then applied to further select a smaller set of landmarks which were considered to be distinctive as shown in Figure 3. The innovation factor λ was chosen to be 0.9 weighting the past significantly more than the present. The threshold for distinctness in Equation 1 was chosen to be 0.2, a value that kept the number of landmarks chosen to relatively few. In Figure 4, the two highest matches of landmarks that scored higher than a threshold probability of 0.8 are shown with linked lines.

The first selection of landmarks based on DOG techniques generated many landmarks scattered all over the two images. More landmarks could usually mean more confidence for matching. However, the computational time for making comparison would also increase. In addition, since non-distinctive objects were not excluded, many of the matches could possibly have been generated by similar objects located at different places.

Figure 3 shows a selection of landmarks that the algorithm chose to be globally distinctive. The number of landmarks was significantly reduced when retaining useful matches between the two images. Since these landmarks should not appear frequently in the environment, the possibility that similar objects appear in different locations is minimised.

The run-time of this algorithm depended on the complexity of the images. On average, the time required to generate landmarks with descriptors took ~ 6 seconds per image while the selection process of distinctive landmarks required only ~ 0.05 second per image. Thus the extra time required to select distinctive landmarks was comparatively small. The time required to calculate the probability between any two landmarks was ~ 0.001 second. On average, the sub images could generate 150 landmarks. Therefore there were 150×149 potential comparisons required to calculate between two images. The maximum time required would be $\sim 0.001 \times 150 \times 150 = 22.5$ seconds. But after

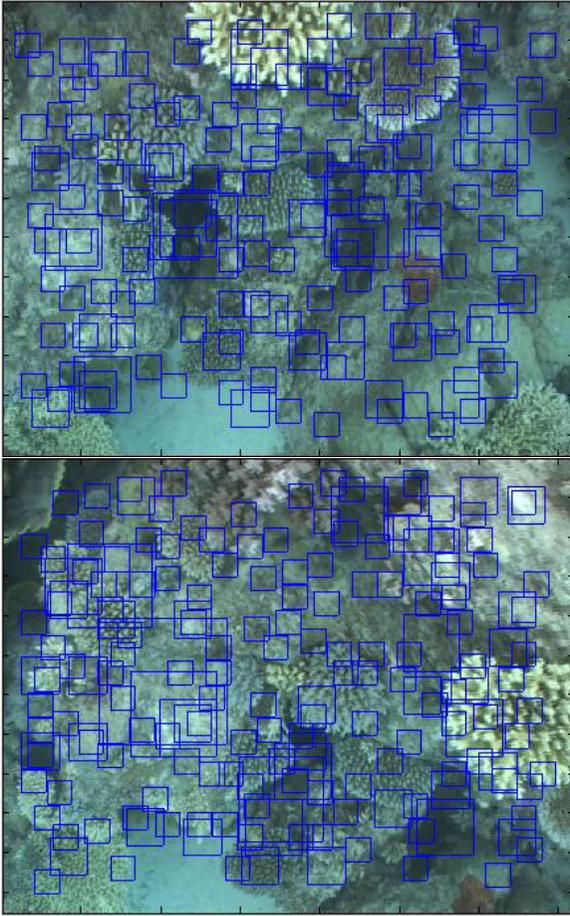


Fig. 2. Two particular images from the Sub sea images. The different sizes of boxes are landmarks generated using texture analysis described in [3].

applying the distinctness selection process, the number of landmarks reduced to ~ 10 per image. The time required to make comparison thus reduced to ~ 0.1 second. The algorithm is currently being re-implemented in C which should improve its speed significantly.

4.2 Global Distinctness Test

The performance of the algorithm was then tested with different images across the environment. The test should reveal whether the algorithm could select objects that are truly distinctive from a human's perspective. The task is in some ways subjective. A group of images are displayed in Figure 5 together

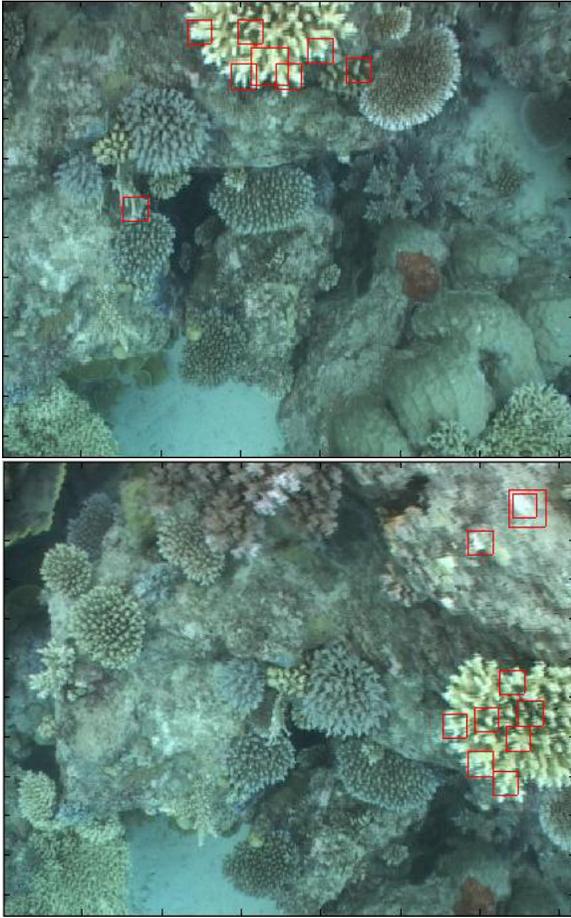


Fig. 3. On the same two images of Figure 2. After applying the Distinctness selection process described in Section 3, the number of landmarks is reduced.

with the landmarks selected by the algorithm. The reader can judge the performance of the algorithm by noting what has been picked out.

As can be seen, the distinctive landmarks are usually the complicated textural corals which tend to be sparsely distributed.

It can be seen that in some of these images, there is a single distinctive object, in which case, the algorithm has concentrated the landmarks in that region. However, in images that contain no obvious distinctive objects, the algorithm has chosen fewer distinctive landmarks scattered over the whole image.

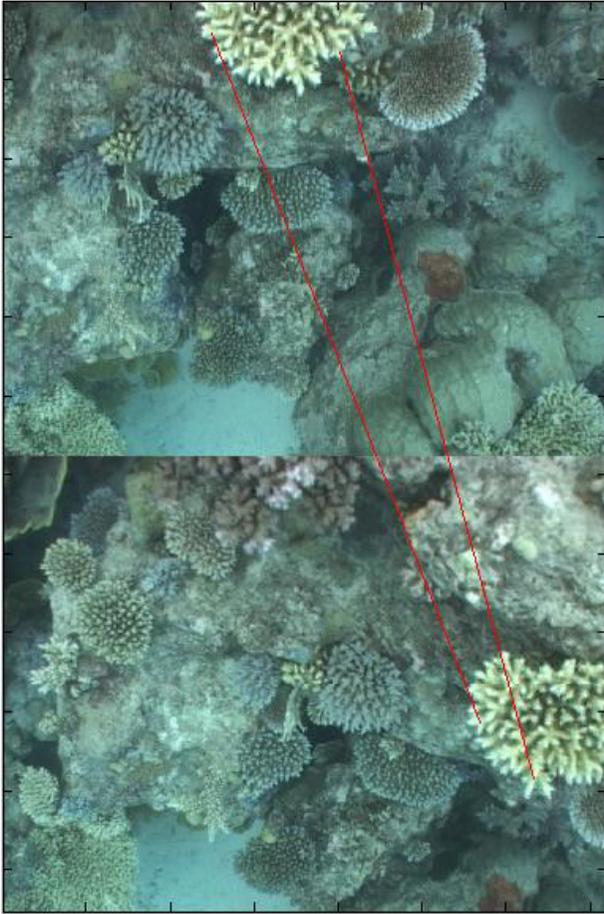


Fig. 4. After comparing each distinctive landmarks, two highest matches that contains probability of over 0.8 are joined by lines for illustration.

4.3 Stability Test

A final test was conducted to check on the stability of chosen landmarks. By stability, we mean that the same landmark should be picked out invariant to any changes in shift, rotation, scale and illumination. A selection of image pairs was made such that these pairs contained relatively large changes in the previously mentioned conditions and contained overlapping regions. After the algorithm was applied to each image to pick out distinctive landmarks, an inspection was made within the overlapping region to count the number of distinctive landmarks that appeared within a few pixels in corresponding locations of the two images. By comparing this number with the number of landmarks that did not correspond in both of the images, a measure of stability was obtained. For example in Figure 3, there were four distinctive

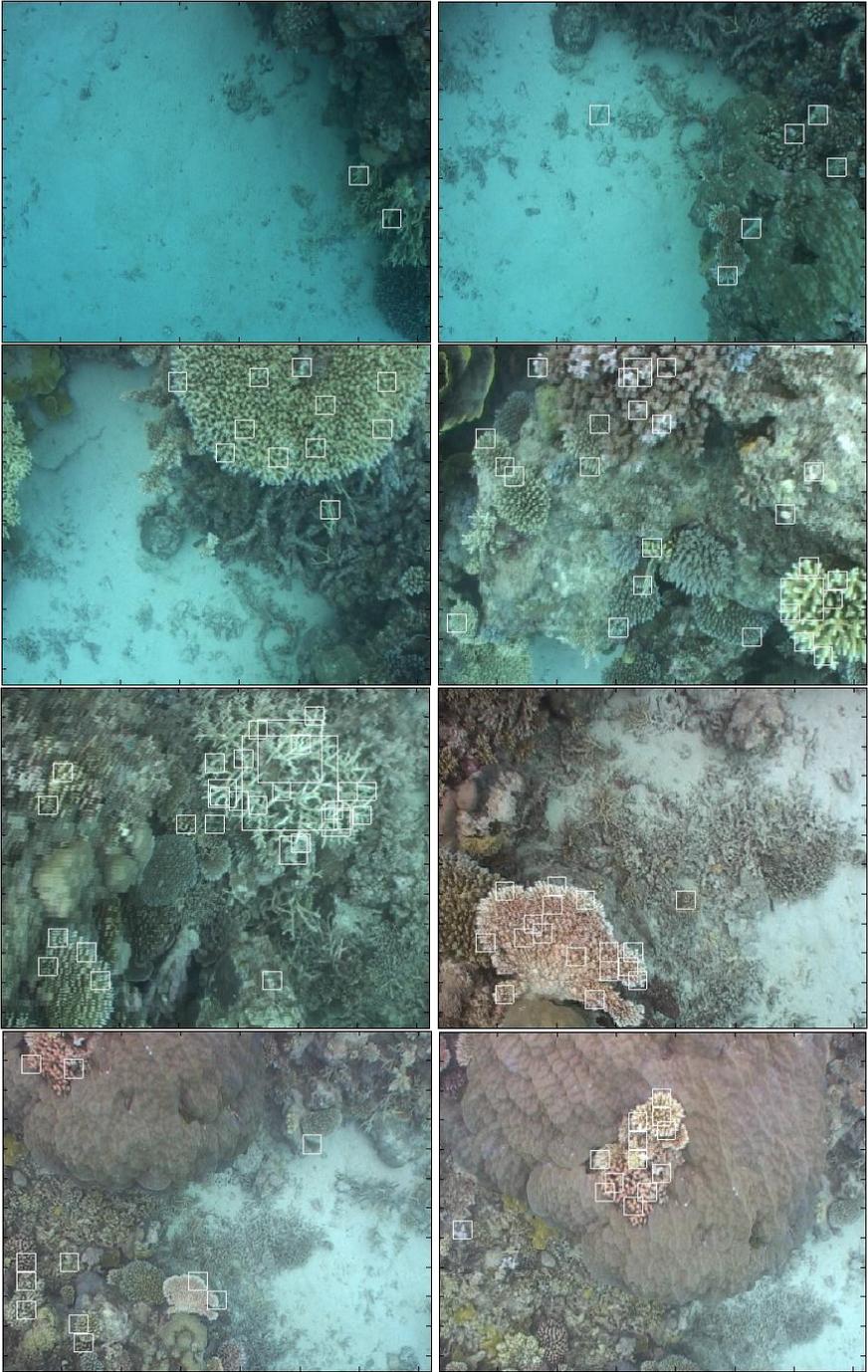


Fig. 5. Sample images from sub-sea series (courtesy of ACFR, University of Sydney, Australia)

landmarks appearing in corresponding locations of both images. On the other hand, there were three which do not correspond in both images.

In Figure 6, 20 pairs of images have been analysed in the way indicated above. On average, 47% of the landmarks selected as distinctive in one image appeared correspondingly in both images. This was deemed a relatively high hit rate for tracking good distinctive landmarks through image sequences and shows promise for enabling map building in a SLAM context.

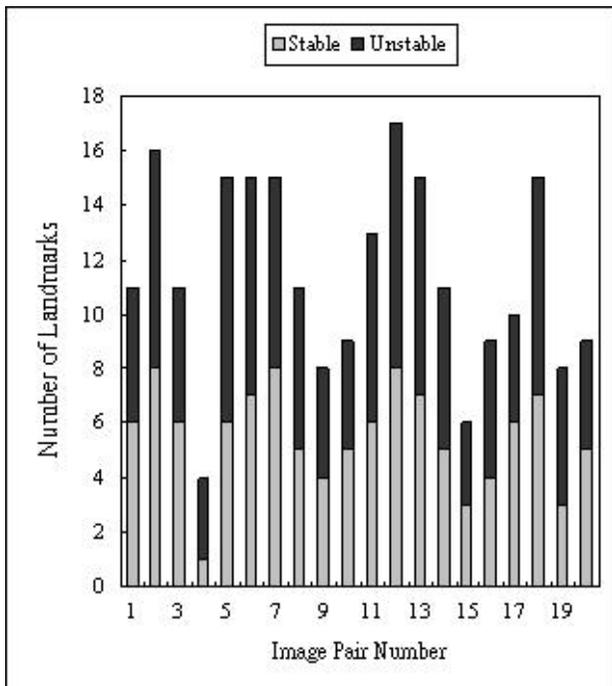


Fig. 6. An analysis of finding stable landmarks over 20 pairs of images.

5 Conclusion and Future Work

The work reported here has shown that it is possible to differentiate image data in such a way that distinctive features can be defined which can be tracked on images as the features progress through a sequence of images in an unexplored environment.

The paper presented an extended algorithm for selecting distinctive landmarks among numerous candidates, that could also be adapted and combined with existing invariant landmark generation techniques such as SIFT or Texture Analysis. In our experiments, the algorithm is demonstrated to discrimi-

nate a small enough set of landmarks that would be useful in techniques such as SLAM.

We are currently working to incorporate this landmark selection algorithm with inertia sensor information to form a functioning SLAM system and deploy it in a submersible vehicle.

Acknowledgment

This work is financially supported by the Australian Cooperative Research Centre for Intelligent Manufacturing Systems & Technologies (CRC IMST) and by the Australian Research Council Centre of Excellence for Autonomous Systems (ARC CAS).

References

1. Csorba M (1997) Simultaneously Localisation and Mapping. PhD thesis of Robotics Research Group, Department of Engineering Science, University of Oxford
2. Williams S B (2001) Efficient Solutions to Autonomous Mapping and Navigation Problems. PhD thesis of ACFR, Department of Mechanical and Mechatronic Engineering, the University of Sydney
3. Kiang K, Willgoss R A, Blair A (2004) "Distinctive Feature Analysis of Natural Landmarks as a Front end for SLAM applications", 2nd International Conference on Autonomous Robots and Agents, New Zealand, 206–211
4. Lowe D G (2004) "Distinctive image features from scale-invariant keypoint", International Journal of Computer Vision, 60, 2:91–110
5. Mikolajczyk K and Schmid C (2002) "An affine invariant interest point detector", 8th European Conference on Computer Vision Czech, 128–142
6. Lindeberg T (1994) "Scale-Space Theory: A Basic Tool for Analysing Structures at Different Scales", J. of Applied Statistics, 21, 2:224–270
7. Harris C, Stephen M (1988) "A combined Corner and edge detector", 4th Alvey Vision Conference Manchester, 147–151
8. Carneiro G, Jepson A D (2002) "Phase-based local features", 7th European Conference on Computer Vision Copenhagen, 1:282–296
9. Tuytelaars T, Van G L (2000) "Wide baseline stereo matching based on local, affinely invariant regions", 11th British Machine Vision Conference, 412–425
10. Schmid C, Mohr R (1997) "Local grayvalue invariants for image retrieval", Pattern Analysis and Machine Intelligence, 19, 5:530–534
11. Freeman W, Adelson E (1991) "The design and use of steerable filters", Pattern Analysis and Machine Intelligence, 13, 9:891–906
12. Mikolajczyk K, Schmid C (2003) "Local grayvalue invariants for image retrieval", Pattern Analysis and Machine Intelligence, 19, 5:530–534
13. Manly B (2005) Multivariate Statistical Methods A primer 3rd edition, Chapman & Hall/CRC

Bimodal Active Stereo Vision

Andrew Dankers^{1,2}, Nick Barnes^{1,2}, and Alex Zelinsky³

¹ National ICT Australia⁴, Locked Bag 8001, Canberra ACT Australia 2601

² Australian National University, Acton ACT Australia 2601

{andrew.dankers,nick.barnes}@nicta.com.au

³ CSIRO ICT Centre, Canberra ACT Australia 0200

alex.zelinsky@csiro.au

Summary. We present a biologically inspired active vision system that incorporates two modes of perception. A peripheral mode provides a broad and coarse perception of where mass is in the scene in the vicinity of the current fixation point, and how that mass is moving. It involves fusion of actively acquired depth data into a 3D occupancy grid. A foveal mode then ensures coordinated stereo fixation upon mass/objects in the scene, and enables extraction of the mass/object using a maximum a-posterior probability zero disparity filter. Foveal processing is limited to the vicinity of the camera optical centres. Results for each mode and both modes operating in parallel are presented. The regime operates at approximately 15Hz on a 3GHz single processor PC.

Keywords: Active Stereo Vision Road-scene Fovea Periphery

1 Introduction

The National ICT Australia (NICTA) Autonomous Systems and Sensing Technologies (ASSeT) *Smart Car* project focusses on *Driver Assistance Systems* for increased road safety. One aspect of the project involves monitoring the driver and road scene to ensure a correlation between where the driver is looking, and events occurring in the road scene [11]. The detection of objects in the road scene such as signs [19] and pedestrians [14], and the location of the road itself [2], form part of the set of observable events that the system aims to ensure the driver is aware of, or warn the driver about in the case that they have not noticeably observed such events. In this paper, we concentrate on the use of active computer vision as a scene sensing input to the driver assistance architecture. Scene awareness is useful for tracking objects, classifying them, determining their absolute position or fitting models to them.

⁴ National ICT Australia is funded by the Australian Department of Communications, Information Technology and the Arts and the Australian Research Council through *Backing Australia's ability* and the ICT Centre of Excellence Program.

1.1 Research Platform

The *Smart Car* (Fig. 1, left), a 1999 Toyota Landcruiser, is equipped with the appropriate sensors, actuators and processing hardware to provide an environment in which desired driver assistance competencies can be developed [9]. Positioned centrally inside the front windscreen is an active stereo vision mechanism. CeDAR, the Cable-Drive Active-Vision Robot [22], incorporates a common tilt axis and two pan axes each exhibiting a range of motion of 90° . Angles of all three axes are monitored by encoders that give an effective angular resolution of 0.01° . An additional CeDAR unit (Fig. 1, right) identical to the unit in the Smart Car is used for initial visual experiments. Although it is stationary and cannot replicate road conditions, it is convenient for algorithm development such as that presented in this paper.



Fig. 1. Research platform. Left: *Smart Car*, and *CeDAR* mounted behind the windscreen (centre). Right: CeDAR, laboratory apparatus.

2 Active Vision for Scene Awareness

A vision system able to adjust its visual parameters to aid task-oriented behaviour – an approach labeled *active* [1] or *animate* [4] vision – can be advantageous for scene analysis in realistic environments [3]. Foveal systems must be able to align their foveas with the region of interest in the scene. Varying the camera pair geometry means foveal attention can be maintained upon a subject. It also increases the volume of the scene that may be depth-mapped. Disparity map construction using a small disparity search range that is scanned over the scene by varying the camera geometry is less computationally expensive than a large static disparity search. A configuration where fixed cameras use pixel shifting of the entire images to simulate horopter re-configuration is more processor intensive than sending commands to a motion axis. Such *virtual* shifting also reduces the useful width of the image by the number of pixels of shift.

3 Bimodal Active Vision

We propose a biologically inspired vision system that incorporates two modes of perception. A peripheral mode first provides a broad and coarse perception

of where mass is in the scene in the vicinity of the current fixation point (regardless of where that may be) and how that mass is moving. The images are processed in their entirety. It does not, however, incorporate the notion of coordinated gaze fixation or object segmentation. Once the peripheral mode has provided a rough perception of where mass is in the scene, the foveal mode allows coordinated stereo fixation upon mass/objects in the scene, and enables extraction of the object or region of mass upon which fixation occurs. We limit foveal processing resources to the region of the images immediately surrounding the image centres.

The human vision system provides the motivation for bimodal perception. Humans find it difficult to fixate on *unoccupied space*. Empty space contains little information; we are more concerned with interactions with objects or mass. Additionally, the human visual system exhibits its highest resolution around the fixation point, over a region of approximately the size of a fist at arms length. The periphery, despite being less resolute, is very sensitive to salient scene features such as colourful or moving objects [21]. For resolute processing, humans centre objects detected in the periphery within the fovea.

3.1 Peripheral Perception

We first provide an overview of the process required to rectify epipolar geometry for active stereo image pairs. Rectified pairs are then used to construct depth maps which are incorporated into an occupancy grid representation of the scene. We also describe how the flow of mass in the occupancy grid is estimated. These techniques provide a coarse 3D perception of mass in the scene.

Active Rectification and Depth Mapping

In [7] we described a rectification method used to actively enforce *parallel epipolar geometry* [15] using camera geometric relations. Though the geometric relations can be determined by visual techniques (see [20]), we use a fixed baseline and encoders to measure camera rotations. We have shown the effectiveness of the rectification process by using it to create globally epipolar rectified mosaics of the scene as the cameras were moved (Fig. 2). The mosaic process allows the use of any static stereo algorithms on an active platform by imposing a globally static image frame and parallel epipolar geometry. Here, we use the process for active depth-mapping. Depth maps are constructed using a processor economical *sum of absolute differences* (SAD) technique with *difference of Gaussians* (DOG) pre-processing⁴ to reduce the effect of intensity variations [5].

⁴ DOG is an approximation to the *Laplacian of Gaussian*.



Fig. 2. Online output of the active rectification process: mosaic of rectified frames from right CeDAR camera.

A Space Variant Occupancy Grid Representation of the Scene

Occupancy grids can be used to accumulate diffuse evidence about the occupancy of a grid of small volumes of space from individual sensor readings and thereby develop increasingly confident maps [10]. Occupancy grids permit Bayesian integration of sensor data. Each pixel in a disparity map is a single measurement for which a sensor model is used to fuse data into the 3D occupancy grid. The occupancy grid is constructed such that the size of a cell at any depth corresponds to a constant amount of pixels of disparity at that depth. It is also constructed such that rays emanating from the origin pass through each layer of the occupancy grid in the depth direction at the same coordinates [7]. Fig. 3 (left) shows an example snapshot of occupancy grid construction.

As described in [8], the velocities of occupied cells in the 3D grid are calculated using an approach similar to that of [16]. This approach estimates 2D optical flow in each image and depth flow from consecutive depth maps. The mosaics remove the effect of camera rotations so that SAD based flow estimation techniques can be used to determine the vertical and lateral components of scene flow (Fig. 3, centre). We are able to assign sub-cell sized motions to the occupied cells in the occupancy grid. The occupancy grid approach was used to coarsely track the location and velocity of the ground plane and objects in the scene [8] (Fig. 3, right) at approximately $20Hz$.

3.2 Foveal Perception

We begin by assuming short baseline stereo fixation upon an object in the scene. We can ensure fixation on an object by placing it at the vergence point using saliency based attention mechanisms⁵. We want to find the boundaries of the object so we can segment it from its background, regardless of the type

⁵ Gaze arbitration combines 2D visual saliency operations with the occupancy grid perception. However, visual saliency and gaze arbitration are not within the scope of this paper.

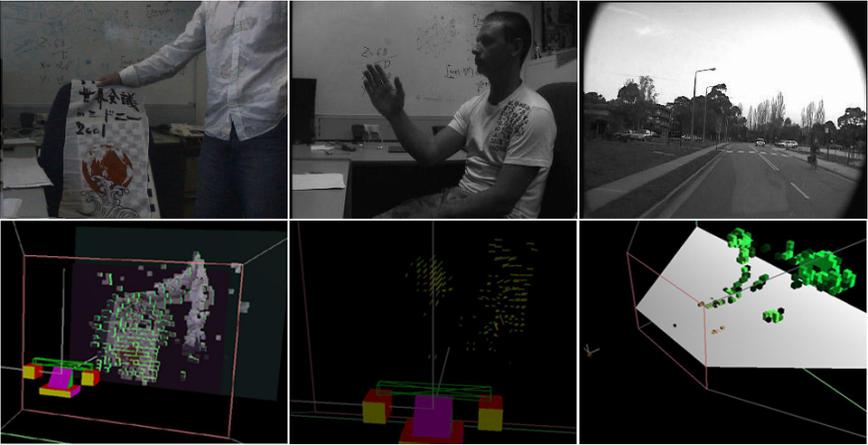


Fig. 3. Peripheral perception. Left: left camera image (top) and occupancy grid representation of mass in the scene with surface rendering (bottom). Centre: left camera image (top) and 3D mass flow vectors (bottom). Right: left camera image of road scene (top) and occupancy grid representation showing ground plane extraction (bottom).

of object or background configuration. Analogous to human vision, we define the fovea as approximately the size of a fist held a distance of 60cm from the camera. For our cameras, this corresponds to a region of about 60×60 pixels.

For humans, the boundaries of an object upon which we have fixated emerge effortlessly because the object is centred and appears identical in our left and right eyes, whereas the rest of the scene usually does not. For synthetic vision, the approach is the same. The object upon which fixation has occurred will appear with identical pixel coordinates in the left and right images, that is, it will be at *zero disparity*. For a pair of cameras with suitably similar intrinsic parameters, this condition does not require epipolar or barrel distortion rectification of the images. Camera calibration, intrinsic or extrinsic, is not required.

ZDF Formulation

A *zero disparity filter* (ZDF) is formulated to identify objects that map to image frame pixels at the same coordinates in the left and right fovea. Fig. 5 shows example ZDF output. Simply comparing the intensities of pixels in the left and right images at the same coordinates is not adequate due to inconsistencies in (for example) saturation, contrast and intensity gains between the two cameras, as well as focus differences and noise. A human can easily distinguish the boundaries of the object upon which fixation has occurred even if one eye looks through a tinted lens. Accordingly, the regime should be robust enough to cope with these types of inconsistencies. One approach is to



Fig. 4. NCC of 3x3 pixel regions at same coordinates in left and right images. Correlation results with higher values shown more white.

correlate a small template in one image with pixels in the same template in the other image. Fig. 4 shows the output of this approach. Bland areas in the images have been suppressed (set to 0.5) using DOG pre-processing. This is because untextured regions will always return a high NCC response whether they are at zero disparity or not. The output is sparse and noisy. The palm is positioned at zero disparity but is not categorised as such. To improve results, image context needs to be taken into account. For this reason, we adopt a Markov Random Field [13] (MRF) approach. The MRF formulation defines that the value of a random variable at the set of sites (pixel locations) P depends on the random variable configuration field f (labels at all sites) only through its neighbours $N \in P$. For a ZDF, the set of possible labels at any pixel in the configuration field is binary, that is, sites can take either the label *zero disparity* ($f(P) = l_z$) or *non-zero disparity* ($f(P) = l_{nz}$). For an observation O (in this case an image pair), Bayes law states that the a-posterior probability $P(f | O)$ of field configuration f is proportional to the product of the likelihood $P(O | f)$ of that field configuration given the observation and the prior probability $P(f)$ of realisation of that configuration:

$$P(f | O) \propto P(O | f) \cdot P(f). \quad (1)$$

The problem is thus posed as a MAP optimisation where we want to find the configuration field $f(l_z, l_{nz})$ that maximises the a-posterior probability $P(f | O)$. In the following two sections, we construct the terms in Eq. 1.

Prior $P(f)$

The prior encodes the properties of the MAP configuration we seek. It is intuitive that the borders of zero disparity regions co-incide with edges in the image. From the approach of [6], we use the Hammersly-Clifford theorem, a key result of MRF theory, to represent this property:

$$P(f) \propto e^{-\sum_C V_C(f)}. \quad (2)$$

Clique potential V_C describes the prior probability of a particular realisation of the elements of the clique C . For our neighbourhood system, MRF theory defines cliques as pairs of horizontally or vertically adjacent pixels. Eq. 2 reduces to:

$$P(f) \propto e^{-\sum_p \sum_{q \in N_p} V_{p,q}(f_p, f_q)}. \quad (3)$$

In accordance with [6], we assign clique potentials using the *Generalised Potts Model* where clique potentials resemble a well with depth u :

$$V_{p,q}(f_p, f_q) = u_{p,q} \cdot (1 - \delta(f_p - f_q)), \quad (4)$$

where δ is the unit impulse function. Clique potentials are isotropic ($V_{p,q} = V_{q,p}$), so $P(f)$ reduces to:

$$P(f) \propto e^{-\sum_{\{p,q\} \in \varepsilon_N} \begin{cases} 2u & \forall f_p \neq f_q, \\ 0 & \text{otherwise.} \end{cases}} \quad (5)$$

V_C can be interpreted as a cost of discontinuity between neighbouring pixels p, q . In practice, we assign the clique potentials according to how continuous the image is over the clique using the Gaussian function:

$$V_c = \frac{e^{-(\Delta I_C)^2}}{2\sigma^2}, \quad (6)$$

where ΔI_C is the change in intensity across the clique, and σ is selected such that 3σ approximates the minimum intensity variation that is considered smooth.

Note that at this stage we have looked at one image independently of the other. Stereo properties have not been considered in constructing the prior term.

Likelihood $P(O | f)$

This term describes how likely an observation O matches a hypothesized configuration f and involves incorporating stereo information for assessing how well the observed images fit the configuration field. It can be equivalently represented as:

$$P(O | f) = P(I_A | f, I_B), \quad (7)$$

where I_A is the primary image and I_B the secondary (chosen arbitrarily) and f is the hypothesized configuration field. In terms of image sites P (pixels), Eq. 7 becomes:

$$P(O | f) \propto \prod_P g(i_A, i_B, l_P), \quad (8)$$

where $g()$ is some symmetric function [6] that describes how well label l_P fits the image evidence $i_A \in I_A$ and $i_B \in I_B$ corresponding to site P (it could, for instance, be a Gaussian function of the difference in observed left and right image intensities at P ; we evaluate this instance – Eq. 11 – and propose alternatives later).

Energy minimisation

We have assembled the terms in Eq. 1 necessary to define the MAP optimisation problem:

$$P(f | O) \propto e^{-\sum_p \sum_{q \in N_p} V_{p,q}(f_p, f_q)} \cdot \prod_P g(i_A, i_B, l_P). \quad (9)$$

Maximising $P(f | O)$ is equivalent to minimising the energy function:

$$E = \sum_p \sum_{q \in N_p} V_{p,q}(f_p, f_q) - \sum_P \ln(g(i_A, i_B, l_P)). \quad (10)$$

Optimisation

A variety of methods can be used to optimise the above energy function including, amongst others, *simulated annealing* and *graph cuts*. For active vision, high-speed performance is a priority. At present, a graph cut technique is the preferred optimisation technique, and is validated for this class of optimisation as per [18]. We adopt the method used in [17] for MAP stereo disparity optimisation (we omit their use of α -*expansion* as we consider a purely binary field). In this formulation, the problem is that of finding the *minimum cut* on a *weighted graph*:

A weighted graph G comprising of vertices V and edges E is constructed with two distinct terminals l_{zd}, l_{nzd} (the source and sink). A cut $C = V^s, V^t$ is defined as a partition of the vertices into two sets $s \in V^s$ and $t \in V^t$. Edges t, s are added such that the cost of any cut is equal to the energy of the corresponding configuration. The cost of a cut $|C|$ equals the sum of the weights of the edges between a vertex in V^s and a vertex in V^t .

The goal is to find the cut with the smallest cost, or equivalently, compute the *maximum flow* between terminals according to the Ford Fulkerson algorithm [12]. The minimum cut yields the configuration that minimises the energy function. Details of the method can be found in [17]. It has been shown to perform (as worst) in low order polynomial time, but in practice performs in near linear time for graphs with many short paths between the source and sink, such as this [18].

Robustness

We now look at the situations where the ZDF performs poorly, and provide methods to combat these weaknesses. Fig. 5a shows ZDF output for typical input images where the likelihood term has been defined using intensity comparison. Output was obtained at approximately 25Hz for the 60x60 pixel fovea on a standard 3GHz single processor PC. For this case, $g()$ in Eq. 8 has been defined as:

$$g(i_A, i_B, f) = \begin{cases} \frac{e^{-(\Delta I_C)^2}}{2\sigma^2} & \forall f = l_z \\ 1 - \frac{e^{-(\Delta I_C)^2}}{2\sigma^2} & \forall f = l_{nz} \end{cases} \quad (11)$$

The variation in intensity at corresponding pixel locations in the left and

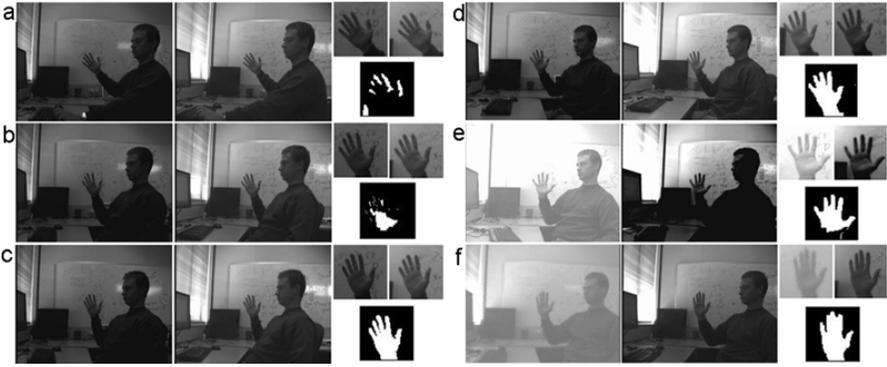


Fig. 5. Foveal perception. The left and right images and their respective foveas are shown with ZDF output (bottom right) for each case *a-f*. Result *a* involves intensity comparison, *b* involves NCC, and *c* DOG NCC for typical image pairs. Result *d-f* show NDT output for typical images *d*, and extreme conditions *e, f*.

right images is significant enough that the ZDF has not labeled all pixels on the hand as being at zero disparity. To combat such variations, NCC is instead used (Fig. 5b). Whilst the ZDF output improved slightly, processing time per frame was significantly increased ($\sim 12Hz$). As well as being slow, this approach requires much parameter tuning. Bland regions return a high correlation whether they are at zero disparity or not, and so the correlations that return the highest results cannot be trusted. A threshold must be chosen above which correlations are disregarded, which also has the consequence of disregarding the most meaningful correlations. Additionally, a histogram of correlation output results is not symmetric (Fig. 7, left). There is difficulty in converting such output to a probability distribution about a 0.5 mean, or converting it to an energy function penalty.

To combat the thresholding problem with the NCC approach, the images can be pre-processed with a DOG kernel. The output using this technique (Fig. 5c) is good, but is much slower than all previous methods ($\sim 8Hz$) and requires yet more tuning at the DOG stage. It is still susceptible to the problem of non-symmetric output.

We prefer a comparator whose output histogram resembles a symmetric distribution, so that these problems could be alleviated. For this reason we chose a simple *neighbourhood descriptor transform* (NDT) that preserves the relative intensity relations between neighbouring pixels, but is unaffected by brightness or contrast variations between image pairs.

In this approach, we assign a boolean descriptor string to each site and then compare the descriptors. The descriptor is assembled by comparing pixel intensity relations in the 3×3 neighbourhood around each site (Fig. 6). In its simplest form, for example, we first compare the central pixel at a site in the primary image to one of its four-connected neighbours, assigning a '1' to the

descriptor string if the pixel intensity at the centre is greater than that of its northern neighbour and a '0' otherwise. This is done for its southern, eastern and western neighbours also. This is repeated at the same pixel site in the secondary image. The order of construction of all descriptors is necessarily the same. A more complicated descriptor would be constructed using more than merely four relations⁶. Comparison of the descriptors for a particular site is trivial, the result being equal to the sum of entries in the primary image site descriptor that match the descriptor entries at the same positions in the string for the secondary image site descriptor, divided by the length of the descriptor string.

Fig. 7 shows histograms of the output of individual neighborhood comparisons using the NCC DOG approach (left) and NDT approach (right) over a series of sequential image pairs. The histogram of NDT results is a symmetric distribution about a mean of 0.5, and hence is easily converted to a penalty for the energy function.

Fig. 5d shows NDT output for typical images. Assignment and comparison of descriptors is faster than NCC DOG, ($\sim 25Hz$) yet requires no parameter tuning. In Fig. 5e, the left camera gain was maximised, and the right camera contrast was maximised. In Fig. 5f, the left camera was defocussed and saturated. The output remained good under these artificial extremes.

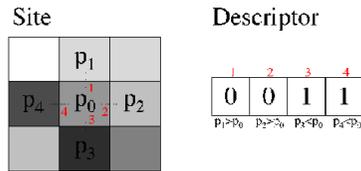


Fig. 6. NDT descriptor construction, four comparisons.

3.3 Bimodal Results

Fig. 8 shows a snapshot of output of the foveated and peripheral perception modes operating in parallel. The coarse peripheral perception detects mass near the (arbitrary) point of gaze fixation. Then the foveal response ensures gaze fixation occurs on an object or mass by zeroing disparity on peripherally detected mass closest to the gaze fixation point. By adjusting the camera geometry, the system is able to keep the object at zero disparity and centred within the foveas. Bimodal perception operates at approximately $15Hz$ without optimisation (threading and MMX/SSE improvements are expected).

⁶ Experiment has shown that a four neighbour comparator compares favorably (in terms of trade-offs between performance and processing time) to larger descriptors.

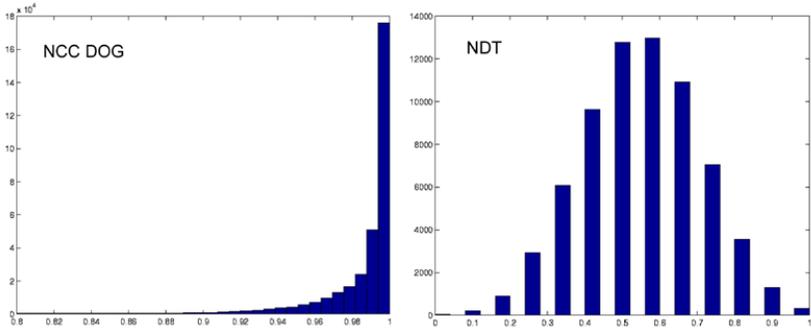


Fig. 7. Histograms of individual NCC DOG (left) and NDT (right) neighborhood comparisons for a series of observations.

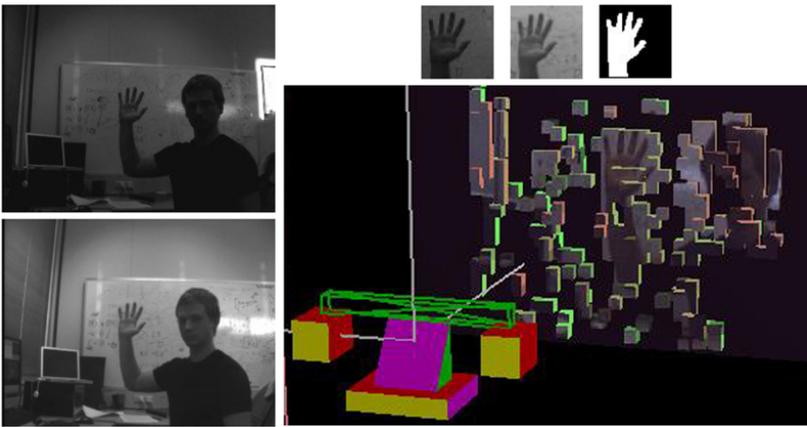


Fig. 8. Bimodal operation. Left: left (top) and right (bottom) input images. Right: Foveal perception (top) and peripheral perception (bottom). Foveal segmentation enhances the coarse perception of mass in the scene.

4 Conclusion

A bimodal active vision system has been presented. The peripheral mode fused actively acquired depth data into a 3D occupancy grid, operating at approximately $20Hz$. The foveal mode provides coordinated stereo fixation upon mass/objects in the scene. It also enables pixel-wise extraction of the object or region of mass upon which fixation occurs using a maximum a-posterior zero disparity filter. The foveal response operates at around $25Hz$. Bimodal perception operates at approximately $15Hz$ on the $3GHz$ single processor PC.

Obtaining a peripheral awareness of the scene and extracting objects within the fovea permits experimentation in fixation and gaze arbitration. Prioritised monitoring of objects in the scene is the next step in our work towards artificial scene awareness.

References

1. J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," in *IEEE Int. Journal on Computer Vision*, 1988.
2. N. Apostoloff and A. Zelinsky, "Vision in and out of vehicles: Integrated driver and road scene monitoring," *IEEE Int. Journal of Robotics Research*, vol. 23, no. 4, 2004.
3. R. Bajczy, "Active perception," in *IEEE Int. Journal on Computer Vision*, 1988.
4. D. Ballard, "Animate vision," in *Artificial Intelligence*, 1991.
5. J. Banks and P. Corke, "Quantitative evaluation of matching methods and validity measures for stereo vision," *IEEE Int. Journal of Robotics Research*, vol. 20, no. 7, 1991.
6. Y. Boykov, O. Veksler, and R. Zabih, "Markov random fields with efficient approximations," Computer Science Department, Cornell University Ithaca, NY 14853, Tech. Rep. TR97-1658, 3 1997.
7. A. Dankers, N. Barnes, and A. Zelinsky, "Active vision - rectification and depth mapping," in *Australian Conf. on Robotics and Automation*, 2004.
8. —, "Active vision for road scene awareness," in *IEEE Intelligent Vehicles Symposium*, 2005.
9. A. Dankers and A. Zelinsky, "Driver assistance: Contemporary road safety," in *Australian Conf. on Robotics and Automation*, 2004.
10. A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Computer Magazine*, 6 1989.
11. L. Fletcher, N. Barnes, and G. Loy, "Robot vision for driver support systems," in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2004.
12. L. Ford and D. Fulkerson, *Flows in Networks*. Princeton University Press, 1962.
13. S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984.
14. G. Grubb, A. Zelinsky, L. Nilsson, and M. Rilbe, "3d vision sensing for improved pedestrian safety," in *IEEE Intelligent Vehicles Symposium*, 2004.
15. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision, Second Edition*. Cambridge University Press, 2004.
16. S. Kagami, K. Okada, M. Inaba, and H. Inoue, "Realtime 3d depth flow generation and its application to track to walking human being," in *IEEE Int. Conf. on Robotics and Automation*, 2000.
17. V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via graph cuts," in *Europuan Conf. on Compupter Vision*, 2002.
18. —, "What energy functions can be minimized via graph cuts?" in *Europuan Conf. on Compupter Vision*, 2002.
19. G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system," in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2004.
20. N. Pettersson and L. Petersson, "Online stereo calibration using fpgas," in *IEEE Intelligent Vehicles Symposium*, 2005.
21. E. Schwartz, "A quantitative model of the functional architecture of human striate cortex with application to visual illusion and cortical texture analysis," in *Biological Cybernetics*, 1980.
22. H. Truong, S. Abdallah, S. Rougeaux, and A. Zelinsky, "A novel mechanism for stereo active vision," in *Australian Conf. on Robotics and Automation*, 2000.

A System for Automatic Marking of Floors in Very Large Spaces

Patric Jensfelt¹, Gunnar Gullstrand¹, and Erik Förell²

¹ Centre for Autonomous Systems, Royal Institute of Technology, SE-100 44 Stockholm, patric@nada.kth.se, gunnar@gullstrand.nu

² Stockholm International Fairs, SE-125 80 Stockholm, Erik.Forell@stofair.se

Summary. This paper describes a system for automatic marking of floors. Such systems can be used for example when marking the positions of stands for a trade fair or exhibition. Achieving a high enough accuracy in such an environment, characterized by very large open spaces, is a major challenge. Environmental features will be much further away then in most other indoor applications and even many outdoor applications.

A SICK LMS 291 laser scanner is used for localization purposes. Experiments show that many of the problems that are typically associated with the large beam width of ultra sonic sensors in normal indoor environments manifest themselves here for the laser because of the long range.

The system that is presented has been in operation for almost two years to date and has been used for every exhibition in the three main exhibition halls at the Stockholm International Fair since then. The system has speeded up the marking process significantly. For example, what used to be a job for two men over eight hours now takes one robot monitored by one man four hours to complete.

Keywords: Floor marking, localization, large space, long range, laser

1 Background

The Stockholm International Fairs (StoFair) [1] is the leading exhibition, fair and congress organizer in Scandinavia. It has 56,500 m² of indoor exhibition area. There are about 70 fairs and over 1000 congresses, conferences and seminars per year. The StoFair uses a completely free layout for the exhibitions, that is, each exhibition can have its own unique layout and there are no restrictions on the shape of the individual stands.

Shortly before each event, the production phase is initiated by marking on the floor where each stand is located. Then the stands are built, the customers move in, the fair runs, the customer move out and the stands are torn down. The next marking takes places and the cycle continues. To maximize the

utility of the space, the time between fairs should be as small as possible. The marking of the stand positions are therefore often performed during the night or at other odd hours.

Traditionally the marking has been carried out manually with tape and a tape measure. It is a very tedious and boring job. For each tape that is placed the person has to bend down all the way to the floor. A large exhibition can have several hundred stands and for each stand several coordinates are marked. This in combination with the odd hours motivates automation of the process.

The automation of the marking process can be realized in several ways and with different levels of autonomy. Most of the time is spent on finding the location to mark and not the marking itself. Clearly, a fully autonomous system provides the largest potential gain. One person could then supervise one or more marking robots and do the job faster than before. It would also relieve the person from the hard labor that the manual process involves.

2 Introduction

To realize an autonomous marking system there are several subproblems that need to be addressed. The robot need to be able to localize accurately, navigate safely through the environment and be able to mark locations on the floor. There are no systems available off the shelf yet for this task to the best knowledge of the authors.

The problem of localization has been thoroughly studied in the robotics literature, see for example [2, 3, 4]. Most of the localization research have been carried out in indoor environments. The main sensor modalities have been the ultra sonic sensors and now lately the laser scanner. Both of these provide range information. In outdoor applications other types of sensors such as GPS and inertial sensors are also common (see e.g. [5]). GPS however, is not available indoor. Different representations have been used where the two main directions are to use features [2, 4] and occupancy grids [6, 3].

Navigation and obstacle avoidance have also attracted a lot of attention over the years. In most indoor environments this is a key component as the distance to obstacle at all times is relatively small. Several methods have been proposed such as the Vector Field Histogram [7], the Dynamic Window Approach [8] and the Nearness Diagram [9]. Also here the sonar and laser scanners have been the most commonly used sensors. In the current application obstacle avoidance is easier then under regular indoor conditions as the environment is very large with few obstacles.

2.1 Outline

The rest of the paper is outlined as follows. Sections 3 lists some of the requirements on the system and presents the overall design. Section 4 discusses

the implementation except for the localization part which is described in Section 5. The results of an experimental evaluation of the system are given in Section 6 and a summary and some conclusions can be found in Section 7.

3 Requirements and Design

3.1 Requirements

This section lists some of the requirements that was put on the system by StoFair.

- The environment is constantly undergoing changes. A system that relies entirely on artificial landmarks, such as retro reflective tapes, being placed throughout the environment would be costly to maintain. It is therefore desirable that the system uses the natural environment as much as possible.
- Since the marks on the floor are the basis for building the walls and placing the carpets they need to have a certain level of accuracy. The StoFair specified 3 cm as an acceptable level.
- It must be easy to maintain the map so that it can be adapted to changes in the environment.
- The system must avoid collisions if there are objects in the way and instead report points as unreachable if blocked.
- The marking is sometimes done several weeks ahead of time. The floor is cleaned with machines between the fairs and each mark that is lost in this process has to be re-measured. This means that the markings have to withstand significant wear for quite some time.
- The system must have the means to notify an operator if there is a problem, such as the batteries being low, the robot is stuck, etc.
- The robot must be able to report what coordinates where marked and which failed. Based on such a report the system should also be able to continue a mission if it was interrupted.
- It would be desirable to add information besides the location of coordinates to the markings.

3.2 Design Decisions

It is clear that the sensor used for localization must have a long range since the environment is very large and the features are sparsely spaced. With GPS being ruled out, radio localization systems not yet being accurate enough, sonar sensor not having the necessary range, the remaining candidate sensors are cameras and laser scanners. Vision was also ruled out since the lighting conditions are often quite bad and the uncertainty was too large whether or not the necessary accuracy could be reached. The choice was therefore to use a laser scanner.

As mentioned in Section 2 there are several ways to represent the environment in a map. Depending on the application one will be better suited than the other. Here, the requirement that a user should be able to maintain the map tipped the scale in favor of the feature based representations. Another reason for this was that a CAD model of the environment already existed which would be easier to exploit in a feature map setting.

To get a head start in the project it was decided to use a commercially available platform. Building a custom made platform would take too much time and cost too much money in the initial phase of the project.

A four wheel platform would have the advantage of handling uneven floors well and be able to traverse small obstacles such as pieces of wood that had been left on the floor. The downside was that it would have significantly worse odometric performance and might not even be able to turn on a carpet for example. A conventional two wheeled, differential drive system was therefore chosen.

4 Implementation

4.1 Platform

The platform of choice for the unit was a Pioneer2-DXE from ActivMedia [10] with an on-board 800MHz computer running Linux. Figure 1 shows the platform equipped with everything that is needed to carry out a marking mission. The standard batteries of the Pioneer robots were later replaced by packs of Ni-MH cells to boost the autonomy from about 2h up to 6h. The laser scanner is the main sensor for localization and navigation. In addition, sonar and bumpers are used as a compliment for close range obstacle avoidance.

Regarding the laser scanner a more thorough investigation should have been made. Now the choice was biased by already being familiar with the SICK family of laser scanners. A SICK LMS 291 was chosen, as it has better range than the indoor SICK LMS 200.

4.2 Map

A map of the environment already existed in the CAD system that is used for planning the layout of the fairs and many other tasks. The CAD system is already familiar to the staff and offers an easy to use interface for editing a map. It was therefore decided that the map format for the robot should be CAD compatible. The DXF format was chosen because it is text based and easy to parse. Changes to the environment made in the CAD system can thus be carried over to the robot effortlessly. Figure 2 shows the features available to the robot for localization in HallC at StoFair. This hall is about 80m by 270m. Looking at the figure it seems to be dense with features, but pillars in the middle are 25m apart.

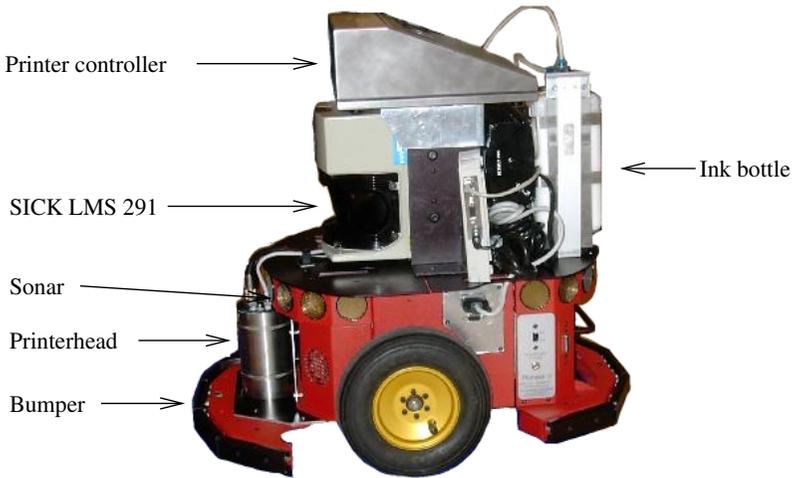


Fig. 1. The platform base is a Pioneer-2DXE. A SICK LMS 291 is the main sensor. For the marking it has been equipped with a industrial ink-jet printing system consisting of a printer head, an ink bottle, hoses and a printer control unit.

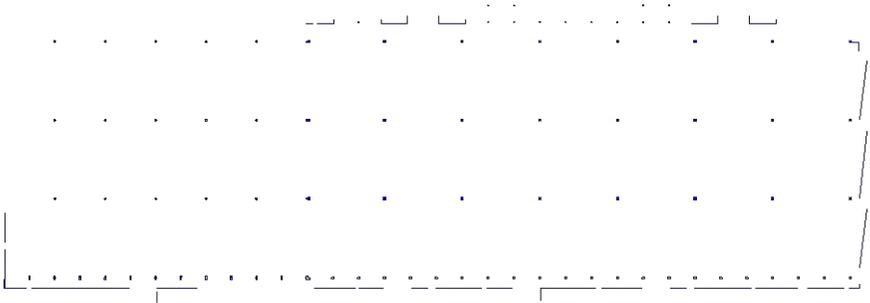


Fig. 2. The wall and pillar features in HallC which is about 80m by 270m large.

4.3 Marking Device

Clearly the system would not be functional without any means to mark locations on the floor. The requirement that the markings must be sturdy enough to withstand running a cleaning machine over them and to allow addition information to be added to the marking led us to use an industrial ink-jet printer from EBS [11]. Such systems are typically used to print on for example cardboard boxes. In its normal application the printer head is fixed and the boxes move past the head on a conveyer belt. Here the printer head is moved by the robot over the floor to create the equivalent of a gigantic plotter. This has given the first platform its name, Harry Plotter.

4.4 Navigation

The navigation system have three main parts that are responsible for i) local planning to make sure that obstacles are avoided, ii) marking of points and iii) global path planning to select which order to traverse the coordinates. The navigation system operates at 20Hz, driven by the rate of the odometric data.

Obstacle Avoidance

Obstacle avoidance in exhibition hall environments is often not as challenging as in a typically indoor environment as there is so much space to maneuver on. However, the system still needs to be able to handle situations with heavy clutter not to get stuck when an area has not been properly cleaned. The system switches between two modes depending on the amount of clutter. In the normal mode a controller similar to the one used in the High Safety situations of the Nearness Diagram [9] is used. In situations with heavy clutter the Global Dynamic Window Approach [12] is used. The speed had to be limited to 0.6m/s as the platform was unable to maintain a straight course at higher speed with all the extra equipment that added to the weight. Clearly, a custom made platform could have performed better here.

Marking

The printing on the floor is done at constant speed (0.15m/s). The printer is “loaded” with a text string and the printing is started by a signal that is normally generated when the object to print on passes a photodetector. Here this signal is generated by the robot.

Besides marking the location of a certain coordinate the system can add extra information that makes it easier to identify the coordinate and make it easier to see how the carpets or walls should be placed. This extra information is given by two optional text strings.

At a speed of 0.15m/s and a sampling rate of 20Hz the error just from the discrete time control could be in the order of 1cm. In addition, there are no real-time guarantees in standard Linux which can also contribute to errors. Another source of errors is the delay between when the signal is given and the printing starts. This delay is measured and compensated for. To reduce the influence of the first type errors the distance to the mark position is calculated in each step. Instead of signalling to print when the mark has been passed, printing starts when it is predicted to be shorter to the mark location in the current iteration than it will be when scheduled the next time.

Global Path Planning

The global path planning here is similar to the traveling salesman problem, each coordinate has to be visited to mark it. Currently a simplistic approach

is used where the stands are completed sequentially. The advantage of this approach, besides being computationally attractive, is that construction of the stands in principal could begin immediately after the first stand has been marked. Planning is performed once before each mission and results in an XML-file with the information needed for each coordinate.

The direction of the text information is defined by the direction of the line that connects the current coordinate with the previous. In cases where there is no previous point, special coordinates which are not marked can be added to the file.

The mission specification file in XML-format can be used directly as a report file. A tag for each coordinate tells if it has been marked or not. A comment regarding the cause of failures can also be added and easily be picked out because of the XML-format.

The robot is unable to mark a certain coordinate in mainly two cases i) the space required by the robot to mark is occupied or ii) it is not sure enough about its position.

4.5 User Interface

The user interface is graphical and has been written in JAVA. It allows the operator to monitor the progress of the robot continuously. The planned path of the robot is displayed and coordinates that have been successfully marked are faded out. Points that the robot was unable to mark are highlighted. In cases where the operator knows that certain parts of a hall are blocked these areas can be de-selected graphically. The graphical user interface also provides information about the estimate time left to complete the mission and the status of the batteries.

As an extra security measure the robot is equipped with a GSM module so that it can send text messages (SMS) to the operator. Such messages are sent when the batteries need to be changed, if the robot is stuck or lost somewhere.

5 Localization

Because of the nature of the task the localization component in the system was one of the keys to success. It has been shown by many that indoor localization using a feature representation can be made highly accurate [4]. The challenge in this application is that the environment is much larger, the floor is not as even and the map might not be completely correct and complete. As was seen in Figure 2 the distance between features is large, in the order of 25m. The surface of the floor varies between relatively smooth concrete to uneven asphalt where the robot can almost get stuck in places.

Two types of features are used in the map, i) lines which correspond to walls and ii) pillars. A wall provides accurate information about the distance and orientation relative to it. However, it would not be possible to rely only

on the walls as they are not visible at all from some poses in the large hall. Notice also that the uneven floor limits the maximum usable range as well. The different features are kept in different layers in the CAD map so that they can be distinguished easily. The pillars come in many different shapes and are defined as sets of lines and arcs. An Extended Kalman Filter (EKF) is used to fuse the information from the measurements of the features and the odometry.

5.1 Lines

There are many examples of using lines for localization in the literature. What makes this case special is the large distances. At typical indoor distance ($< 10\text{m}$) the foot print of the laser scanner is often small enough to neglect. However, in the large halls this will lead to large errors. The beam width of the SICK LMS 291 is in the order of $0.01\text{rad} \approx 0.60^\circ$ [13]. The footprint can easily be in the order of a meter in some cases. When performing line extraction for the localization this must be taken into account. Figure 3 shows how the compensated point can be calculated assuming that the reflecting surface is rough.

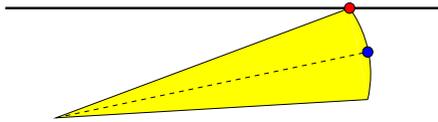


Fig. 3. The center point (blue) is reported by the laser scanner and the one at the end of the arc (red) is an approximation of the true reflection point

The problem with this model is that the effective beam width, depends (at least) both on distance and on the reflecting material. In the environment under consideration some walls are very smooth, whereas some are rough. Just as ignoring the beam width is bad, erroneously compensating for it is equally bad. Therefore, to avoid having to augment the map with reflectivity information for the walls another approach is taken. Only wall segments that can be observed close to perpendicular are used for localization. Because of the layout of the halls this is generally not a limitation.

5.2 Pillars

The description of the pillars gives a flexible representation that can account for the many types of pillars that are present in the environment. When matching scan points to the contour of the pillars the ICP algorithm [14] is used. The contour is sampled to generate the reference point to match the scans against (see Figure 4). The pillars all have similar painted surfaces and thus

the compensation illustrated in Figure 3 can be used. Only the sample points that correspond to model segments that are visible from the current pose are used for the different pillars. After associating scan points to pillars based on their position the ICP algorithm is run to associate the points with the different segments on the pillars. The position of each scan point is then compensated for (see Figure 3), according to the associated pillar segment. After this a second turn of the ICP algorithm is run to make adjustments caused by the beam width compensation.

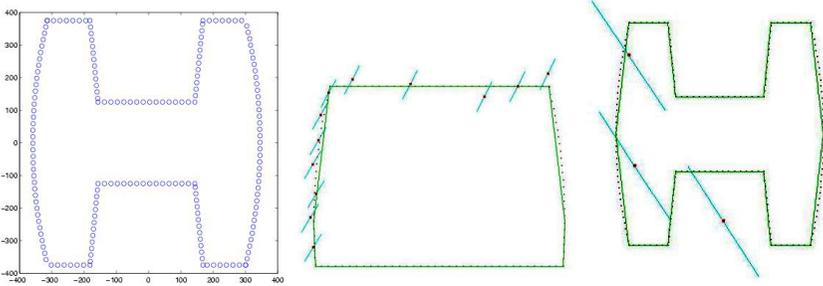


Fig. 4. Left: Sampled contour as reference points for the ICP algorithm. Mid: Close range observation. Right: Long range observation.

6 Experimental Evaluation and Modifications

At the end of the project, in February 2003, a few month before the system was put into operation, a full scale test of the system was performed. The total number of coordinates was 722, out of which 518 should be marked³. The length of the trajectory was 4374m and contained 246 stands corresponding to a total stand area of 10150m². The total time for the task was 4.5h and the robot succeeded to mark 492 out of the 518 points.

To evaluate the marking accuracy, the 492 marked coordinates were hand measured. Figure 5 shows the error distribution both in terms of the absolute error and separated into the x - and y -components. The standard deviation for the absolute error was 18mm and the average error was 28mm.

There are many sources of errors. One was mentioned before dealing with the printer. Furthermore, the CAD map that is the basis for the localization is accurate to about 1-2cm. The laser sensor also has a systematic error in the order of centimeters [13]. Finally, the hand measured positions also have an error associated with them. Put together these errors can explain most of the fluctuations in the accuracy. The really large errors (≥ 75 mm), which

³ The rest defined directions

represents less than 3% of the total number points, are the result of data association errors. Many of these are caused by unmodeled objects being in the scene. For example, the pillars are often used to lean all sorts of things against to get them out of the way. At large distances it is very difficult to detect such errors.

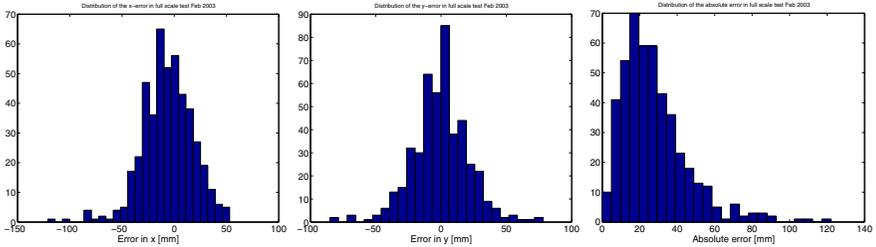


Fig. 5. Error distribution for the marked point for the first full scale test.

There were two causes for the 26 coordinates that could not be marked.

- In some places the trajectory was planned such that only a single wall was visible, i.e. the robot was driving close to and toward the wall. In these cases the pose of the robot could not be determined with enough accuracy and these coordinates were skipped.
- The robot is unable to mark coordinates that are too close to obstacles. Furthermore, in the current implementation the robot also wants to be able to drive straight for a distance before the mark is made to stabilize everything and reach the right printing speed.

6.1 Modifications

Some modifications were made to the system before it was put into operation in August 2003.

A method based on using temporary localization stick landmarks was developed. The halls at the StoFair all have some coordinates permanently marked on the floor to assist in the manual marking process. Whenever such a point comes into view the system searches for a stick there. If it is found very close to the predicted position it is added to the map and henceforth used in the localization process. The sticks have their own layer in the CAD map.

In many cases where the robot is uncertain of its pose for a mark, the pose can be determined better if the robot is reversing the mark direction. Therefore, the robot attempts to mark a point in the opposite direction before reporting a failure due to insufficient accuracy. The result of this modification is that the robot is able to mark all positions free from obstacles under normal conditions.

6.2 Re-evaluation

The same fair that was evaluated in 2003 was evaluated again in 2005. This time a total of 194 coordinates spread out over the hall were examined manually. Out of these, 188 points were marked and 6 had been skipped because they were too close to obstacles (pillars). The ability to turn around and mark in the other direction reduced the number of failures. The marked coordinates had the same average error and standard deviation as in the evaluation two years earlier.

The system has now become a natural part of the marking process. It has been used for marking every fair in the three main halls since it was taken into service in 2003. The robot operators are from the same group of people that used to perform the manual marking. They do not have any special computer or robotics training which indicates that the system is easy enough to use by people that have no robotics background.

7 Summary and Conclusion

To summarize we have presented a system for automatic markings of floors. It has been used to mark every fair in the three main halls at StoFair since August 2003. The system has shown that high accuracy can be achieved in very large areas, but that the laser scanner has to be modelled better than in typical indoor applications.

Figure 6 shows an example of how it might look when two fairs have been marked in different colors. The dashed lines have been overlayed graphically to highlight the location of the walls for the two stands. Notice how one of the two texts will be visible even after the construction as it ends up outside the wall.

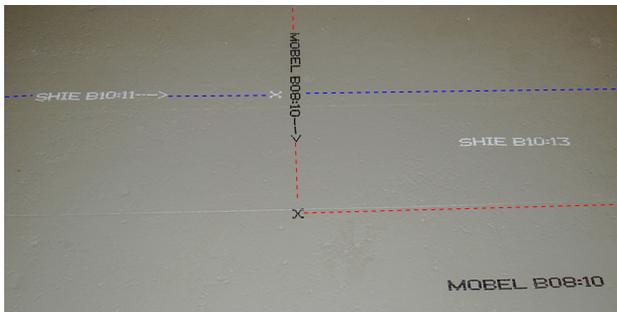


Fig. 6. An example of how it looks on the floor when two fairs have been marked

The most important indicators of success in this project is the fact that the system has been used now for almost two year continuously and is greatly

appreciated by the staff and that a second platform has been built. At first many considered it to be mysterious and some were sceptical but today it is taken for granted. As a final remark the time to mark a typical fair has been cut from 8h with two men to about 4h with one robot and one man that potentially can run more than one robot at a time.

References

1. Stockholm International Fairs, SE-125 80 Stockholm, SWEDEN. <http://www.stofair.se/>.
2. J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, 1991.
3. W. Burgard, D. Fox, D. Henning, and T. Schmidt, "Estimating the absolute position of a mobile robot using position probability grids," in *Proc. of the National Conference on Artificial Intelligence (AAAI-96)*, (Portland, Oregon, USA), pp. 896–901, Aug. 1996.
4. K. Arras and N. Tomatis, "Improving robustness and precision in mobile robot localization by using laser range finding and monocular vision," in *Proc. of the 3rd European Workshop on Advanced Mobile Robots (Eurobot'99)*, (Zürich, Switzerland), pp. 177–185, Sept. 6–9, 1999.
5. E. Nebot, S. Sukkarieh, and H. Durrant-Whyte, "Inertial navigation aided with gps information," in *Proc. of Fourth Annual Conference on Mechatronics and Machine Vision in Practice*, pp. 169–174, Sept. 1997. Describes difference between direct and indirect kalman filter.
6. A. Elfes, "A sonar-based mapping and navigation system," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'86)*, vol. 3, pp. 1151–1156, Apr. 1986.
7. J. Borenstein and Y. Koren, "The vector field histogram - fast mobile obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 278–288, June 1991.
8. D. Fox, W. Burgard, S. Thrun, and A. B. Cremers, "A hybrid collision avoidance method for mobile robots," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA '98)*, vol. 2, (Leuven, Belgium), pp. 1238–1243, May 1998.
9. J. Minguez and L. Montano, "Nearness diagram navigation (nd): A new real time collision avoidance approach," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, pp. 2094–2100, 2000.
10. ActivMedia Robotics. <http://www.activmedia.com/>.
11. EBS Ink-Jet Systems GmbH. <http://www.ebs-inkjet.de/>.
12. O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'99)*, vol. 1, pp. 341–346, May 1999.
13. SICK, *Laser Measurement Systems, Technical Description*.
14. P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

Development of an Angular Characterisation System for Cooperative UAV / UGV Applications

Paul Thompson and Salah Sukkarieh

ARC Centre of Excellence for Autonomous Systems,
Australian Centre for Field Robotics,
The Rose St. Building, J04, University of Sydney, NSW, Australia 2006
{p.thompson,salah}@acfr.usyd.edu.au

Summary. This paper develops a theory for the decentralised estimation of angular profiles for point characterisation. Angular profiles are an example of spatially distributed characterisation which may be further processed for classification and identification. This paper also describes the development of a vision system for integration into a decentralised data fusion system for tracking and characterisation. Visual results are presented showing the field setup and sensor observations.

Keywords: characterisation, decentralised estimation, vision, angular profile, information theoretic properties

1 Introduction

This project extends our previous work on multi-air vehicle decentralised data fusion (DDF), which used point feature detection for object tracking, localisation and mapping [1]. This paper discusses our developments in the incorporation of point characterisation for detecting and identifying low signature, partially obscured or hidden objects.

Multiangular image processing is common in geoscience applications for land classification [2]. A wide variety of application specific domains implement angular characterisation, but generally not in a recursive estimation framework, let alone in a method amenable to probabilistic decentralised data fusion.

In SLAM, tracking and structure from motion, point features are not generally considered to possess interesting properties as a function of viewing angle. Multiangular characterisation in a decentralised, real time object tracking environment is a new approach.

The rest of this paper is laid out as follows: Section 2 introduces our approach, explaining the purpose of characterisation and angular profiles in particular.

Section 3 describes the setup of our vision system and field tests. Section 4 shows visual results from the field tests. Section 5 develops a technique for the decentralised estimation of angular profiles, with simulated demonstrations. Section 6 outlines directions for future development.

2 Approach

Our aim is to incorporate identity and class of objects into our existing sensing and decentralised data fusion system [1, 3] and to include this in the development of an information theoretic control layer.

In general, we are proposing to perform feature characterisation from sensor data prior to performing identification upon the characterisation results. These techniques incorporate spatially distributed observations and build spatially distributed representations of objects to provide a rich basis for classification.

Angular dependence of properties is one particular dimension we are considering for characterisation. This is based on the challenges of feature identification and classification from a single frame or scan, especially given the possibility of hidden or obscured objects.

Characterising angular properties requires observing from a wide range of angles. This behaviour will be useful for characterisation in the form of angular profiles described here and also for other characterisations such as estimating 3D structure [4] and bearing only localisation.

In this paper, recent results are shown from a flight trial involving circling around features of interest.

3 Experimental Setup

The vision system is mounted as a nose module on our UAV, as shown in figure 1(a). The colour camera is a Sony XCD-X710CR. This was chosen for its high resolution (1024×768) and high frame rate (up to 30 fps at full size). The colour camera interfaces via a standard firewire device for commanding and image transfers. The trigger input on the camera is driven from a parallel port output signal driven from a custom Linux kernel module for periodic events and timestamping.

Both cameras are mounted in fixed positions, along the lateral axis of the UAV. This geometry is consistent with the need for viewing ground features from many angles and consistent with the flight parameters of the UAV. The UAV flies in arc trajectories around the ground features with the bank angle, forward velocity, relative altitude and turn radius matched to ensure that the sensor footprint covers the ground feature and that flight parameters are kept within safe limits. Details of the UAV control architecture are given in [5].

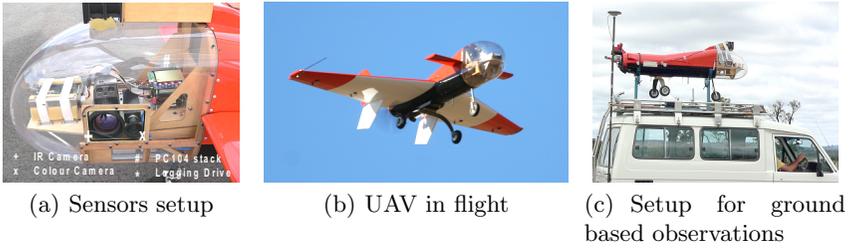


Fig. 1. Sensors, flight vehicle and ground vehicle setup

Ground vehicle observations were obtained using the same vision system, as shown in figure 1(c).

Flight tests were carried out in a rural environment consisting of natural and pre-existing man-made features. Some typical features can be seen in figure 2. These include isolated trees and vegetation patches, live animals (cattle), long straight fences with junctions and corners, straight and curved roads (including the airstrip), a continuous curved river, isolated dams and water tanks, sheds and other buildings.

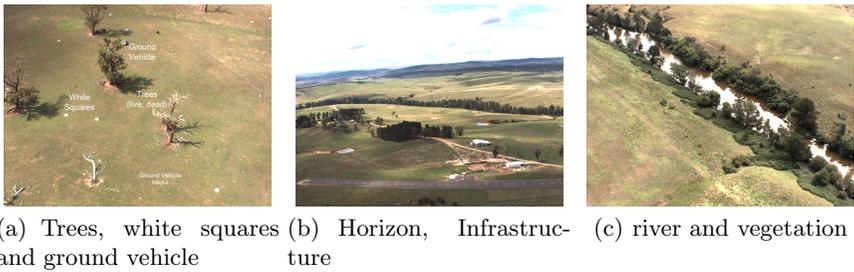
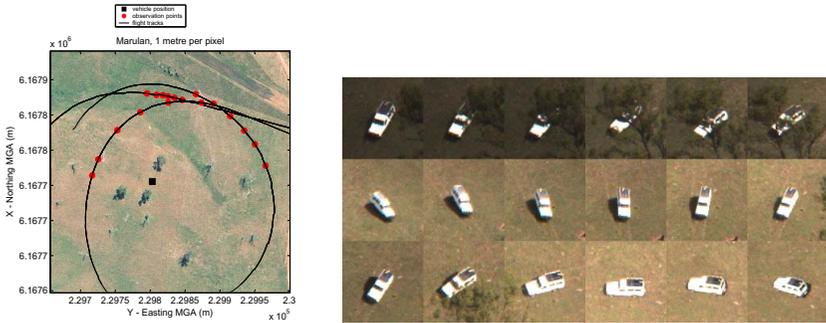


Fig. 2. Environmental setup

Various artificial features were introduced into the environment; white squares, patterned cylinders and a vehicle. The white squares are 1×1 metre plastic squares which are highly visible from the air and ground. These were surveyed in position using DGPS to act as fiducial markers for the confirmation of tracking and image registration results (of their own position and of nearby natural features). White squares are the most basic object for tracking purposes. The patterned cylinders are 2.5 metres high by 1.5 metres diameter cylinders with a printed pattern wrapped around the circumference. The vehicle acts as a realistic test case for tracking and classification, (see figure 3(b)).

4 Flight Imagery

Results are shown here for the images obtained from the UAV mounted vision system. Images will be subjected to feature extraction and used for construction of angular profiles of the various natural and artificial features. Figure 3 shows the ground vehicle as viewed from multiple positions during flight. Images in figure 2 were also obtained from the UAV vision sensor.



(a) A sample of observation positions. Dots indicate the UAV vehicle positions for each observation shown below.

(b) A selection of views of the vehicle. Each row is obtained from a single pass of the UAV past the site.

Fig. 3. Images from the flight data

5 Estimation of Angular Profiles

This section describes an approach to the estimation of angular profiles. Angular profiles will be important to the development of identification and classification algorithms (for example, [2]) and will play a role in directing information theoretic control algorithms into circling features of interest. Angular profiles can be considered as a container for observables which vary in angle. It is a subject of future development to incorporate the extraction of such observables from images, however the angular profiling technique is demonstrated here with preliminary observations. Possible observables include geometry, colour, texture and outputs from single-view image processing techniques. Angular profiles can be considered as a useful intermediate between image fusion and three dimensional object estimation on the one hand, and the fusion of single aspect classification results on the other [6].

Angular profiles are cast in an information filter (inverse covariance) form [7]. Recent developments from the SLAM community in large state space management in information form will be beneficial in the implementation [8].

State Description. An angular profile of an object is a function $\mathbf{x}(\theta)$. The function $\mathbf{x}(\theta)$ is discretised into a finite set of angles $[\theta_1 \cdots \theta_N]$ covering all

angles $0 \leq \theta_i < 2\pi$. Each $\mathbf{x}_i = \mathbf{x}(\theta_i)$ is a continuous scalar, $\mathbf{x}_i \in \mathbb{R}$. The state space for estimation consists of the state vector $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_N]^T$.

An angular profile covering two dimensions of angle is similarly a function $\mathbf{x}(\theta, \phi)$ discretised into a finite set covering $0 \leq \theta_i < 2\pi$ and $-\pi \leq \phi_i < \pi$.

Problem Description. The challenge in the theory behind estimation of angular profiles lies in describing the correlations between angular states such that the profile behaves well during estimation. In particular, it is desired to apply a consistency model, to reflect the fact that there can not be sudden jumps in the profile across small changes in angle. This consistency model plays the role of a process model in ordinary scalar estimation, in that it serves to smooth out the noise presented by observations.

Consistency models are required in order to smooth out observation noise for observations from multiple angles. By analogy with conventional scalar estimation, process models smooth out observation noise for observations from multiple points in time. Consistency models are required so that correlations can be propagated around the profile such that observations at one angle affect the estimates at all other angles. By analogy with conventional scalar estimation, process models describe correlations such that observations affect future estimates.

This means that consistency models are as fundamentally important to the estimation as process models are to conventional estimation and that consistency models play the same role in the estimation as conventional process models.

5.1 A Differential Observation Technique for Angular Consistency Models

This section describes a technique for constructing the correlations between the angular profile states. This section is described with reference to a one-dimensional angular profile. The extension to two dimensions is described in section 5.3

A consistency model for the angular profile is assumed, taking the form given in equation 1 and re-arranged in terms of all states as in equation 2. The consistency model in examples and demonstrations here is a constant value model, $\mathbf{F} = \mathbf{I}$. This consistency model is an observation of the difference between successive states. The true difference is \mathbf{w} , which is modelled as zero mean Gaussian noise of covariance \mathbf{R}_c . \mathbf{R}_c , \mathbf{Z}_c and \mathbf{H}_c refer to the parameters of the consistency model update.

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w} \quad (1)$$

$$(\mathbf{x}_k - \mathbf{F}\mathbf{x}_{k-1}) = \mathbf{w}$$

$$[\mathbf{0} \cdots -\mathbf{F} \quad \mathbf{I} \cdots \mathbf{0}] [\mathbf{x}_0 \cdots \mathbf{x}_{k-1} \quad \mathbf{x}_k \cdots \mathbf{x}_N]^T = \mathbf{w} \quad (2)$$

Equation 2 is of the form given in equation 3

$$\mathbf{Z}_c = \mathbf{H}_c \mathbf{X} + \mathbf{v} \quad (3)$$

$$\begin{aligned} \mathbf{Z}_c &= [\mathbf{0}] & \mathbf{H}_c &= [\mathbf{0} \cdots -\mathbf{F} \mathbf{I} \cdots \mathbf{0}] \\ \mathbf{v} &= \mathbf{w} & E[\mathbf{w}] &= \mathbf{0} & E[\mathbf{w}\mathbf{w}^T] &= \mathbf{R}_c \end{aligned}$$

The consistency model is applied to each pair of successive states, resulting in a concatenated \mathbf{H}_c matrix, shown in equation 4 for an example with five states.

$$\mathbf{Z}_c = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad \mathbf{H}_c = \begin{bmatrix} -\mathbf{F} & \mathbf{I} & & & \\ & -\mathbf{F} & \mathbf{I} & & \\ & & -\mathbf{F} & \mathbf{I} & \\ & & & -\mathbf{F} & \mathbf{I} \\ \mathbf{I} & & & & -\mathbf{F} \end{bmatrix} \quad (4)$$

The final row in equation 4 is a periodic boundary condition which observes the difference between the first (\mathbf{x}_1) and last (\mathbf{x}_N) state elements¹. The angular profile must be periodic by definition, i.e. $\mathbf{x}(0) = \mathbf{x}(2\pi)$.

This consistency model observation can be applied as an observation update. Here we assume that an Information filter (inverse-covariance) representation of the state variables is being used. The Fisher Information matrix, \mathbf{Y} , is initialised to zero to represent infinite uncertainty. The consistency model observation becomes an addition to \mathbf{Y} , as shown in equation 5

$$\mathbf{Y} = \mathbf{0} + \mathbf{H}_c^T \mathbf{R}_c^{-1} \mathbf{H}_c \quad (5)$$

The resulting \mathbf{Y} information matrix is shown by example in equation 6, with $\mathbf{R}_c^{-1} = \mathbf{I}$ and $\mathbf{F} = \mathbf{I}$.

$$\mathbf{Y} = \begin{bmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{bmatrix} \quad (6)$$

Note that the information matrix is sparse. The number of non-zeros is given by $3Nn$ where N is the number of angles in the discretisation and n is the size of the state at a given angle. Note that at this point $\det(\mathbf{Y}) = 0$, as there have been no genuine observations. The rank of \mathbf{Y} is $N - 1$. There is zero information along the *sum of all states* eigenvector but all others are nonzero.

¹ Note that $\mathbf{x}_1 \neq \mathbf{x}_N$, since $\theta_1 \neq \theta_N$ (they are separated by one interval)

Observations. The most basic observation simply observes the value at one angle θ_i , updating \mathbf{x}_i . In this case the observation \mathbf{H} matrix selects one particular angle state eg: $\mathbf{H} = (1 \ 0 \ \cdots \ 0)$. Note that there is a data association problem involved in choosing which angle state to associate an observation with.

Using the Information filter observation update equation, observations of \mathbf{x} at particular angles are made by information addition:

$$\mathbf{Y}^+ = \mathbf{Y}^- + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \quad (7)$$

The form of the information update results in additions to the diagonal of \mathbf{Y} , retaining the sparse layout regardless of the number of observations.

This technique inherits the scalability and decentralisation properties of the information filter. Multiple observations are fused recursively without increase in the size of the representation. Furthermore, the information matrix is always highly sparse, in a banded pattern. By arriving at an information form description of the angular profile, the technique is readily applicable to incorporation into our existing decentralised data fusion architecture [1].

5.2 Angular Profile Simulations

The following simulations demonstrate the theory. A true angular profile was created from an arbitrary smooth function. Observations were taken from the true profile with zero mean Gaussian noise consistent with the observation model. To plot the angular profile, the information matrix and vector are converted to covariance and estimate. The estimated angular profile is then given by \mathbf{x} . Error bounds at each angle are given by the diagonals of \mathbf{P} .

Figure 4(a) shows the angular profile after two observations. There is no direction imposed on the propagation of uncertainty around the profile; it is symmetrical in both directions. Uncertainty grows as one moves away from angles where an observation was made. This is reflected in the structure of the covariance matrix, Figure 4(c). Note that correlations exist broadly across regions without observations, yet the information matrix remains sparse and banded. Note that the individual observations become simple additions onto the diagonal with almost no processing required for successive observations (especially not a matrix inversion). The profile construction problem is essentially deferred until the information matrix is inverted.

5.3 A Two Dimensional Angular Consistency Model

Section 5.1 described the development of the consistency model for a circular (one dimensional) angular profile. This section extends the structure of the consistency model to cover two dimensions, forming a surface.

Equations 1 to 3 refer to the application of a consistency model on pairs of elements according to their adjacency on the profile, including a periodic boundary condition. This technique applies to pairs of states in no special order, therefore it is possible to apply the consistency model to states which are adjacent in an undirected graph of arbitrary structure. The undirected

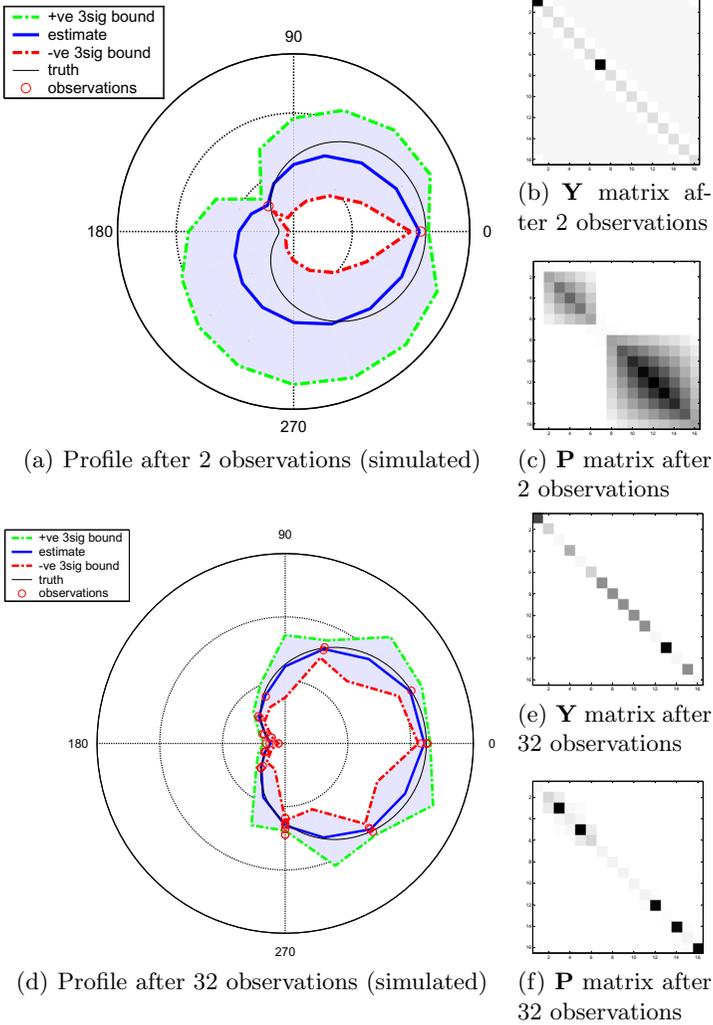


Fig. 4. Simulated Angular profiles.

graph structure for the one dimensional profile is a simple ring. The undirected graph structure for the two dimensional profile is a spherical mesh.

The spherical mesh is generated from DistMesh [9]. The mesh is uniform overall but not perfectly regular. The positions of the nodes and the mesh connectivity is constant throughout the estimation process. In future revisions the irregular mesh will be replaced by a regular mesh with identical edge lengths and number of edges per node in order to ensure symmetry and uniformity.

The consistency model is described by an \mathbf{H}_c matrix equivalent to equation 4. This matrix is formed by iterating through each of the graph edges of the mesh structure, where each edge yields one row of \mathbf{H}_c . For each edge, where the edge is between nodes i and j :

$$\mathbf{H}_{c_{row,i}} = \mathbf{I} \quad \mathbf{H}_{c_{row,j}} = -\mathbf{I} \quad (8)$$

The consistency model observation becomes an addition to \mathbf{Y} , as in equation 5.

5.4 Two Dimensional Angular Profiles Demonstration

Figure 5 shows a demonstration of angular profiles from our flight vehicle and ground vehicle. The patterned cylinder object was characterised according to the metric area of the object as viewed from the (air or ground) borne image sensor. This is a preliminary observable for demonstration of the estimation structure. Figure 5(d) shows the separate contributions from the air and ground vehicle, which are separate due simply to their differing angles of elevation. The fusion of information from air and ground is simplified by the use of angular profiles because they allow explicit differences in value at viewing angles. Hence it is not required that features be absolutely identical from air and ground. Figures 5(a) and 5(b) are shown at the same orientation. The peaks in profile information correspond to the groups of observation points where multiple observations have been fused. Regions without observations take on an estimate obtained through the network of consistency models, causing those regions to have non-zero information.

5.5 Information Theoretic Properties of Two Dimensional Angular Profiles

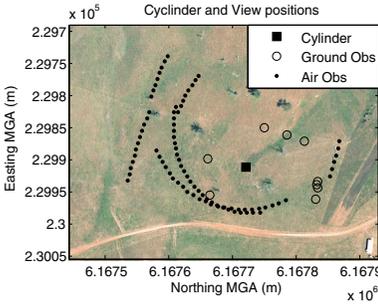
One application of angular profiles is in causing information theoretic control schemes [10] to explore multiple viewing angles of point features (in addition to spatial exploration over multiple features). This section describes the properties of the determinants of the information matrices of angular profiles.

The entropic information i of an n -dimensional Gaussian variable with Fisher information, \mathbf{Y} and the mutual information I between two alternate information matrices \mathbf{Y}_a and \mathbf{Y}_b are given by:

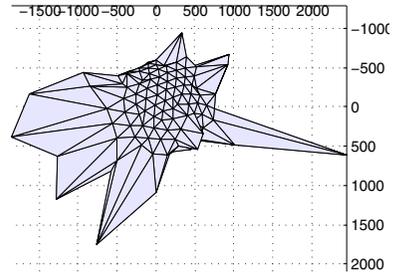
$$i = \frac{1}{2} \log [(2\pi e)^n |\mathbf{Y}|] \quad I = \frac{1}{2} \log \left[\frac{|\mathbf{Y}_a|}{|\mathbf{Y}_b|} \right] \quad (9)$$

Angular profiles determinants have the following properties:

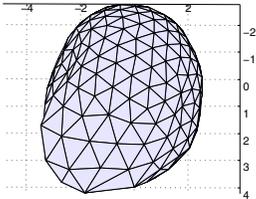
- After application of the consistency model but before observations, $|\mathbf{Y}| = 0$. This means that \mathbf{Y} retains the properties of a non-informative prior after application of the consistency model.
- A single observation causes the determinant, $|\mathbf{Y}_b|$, to be non-zero.



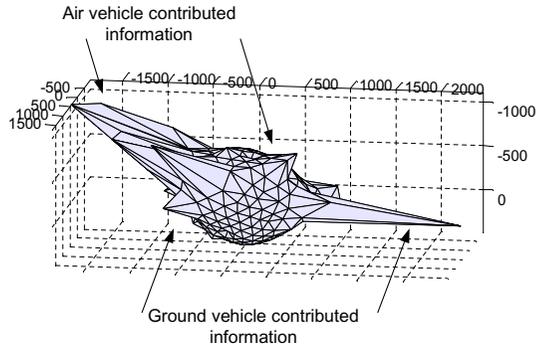
(a) Cylinder Object location and air and ground viewing positions



(b) Radius represents the profile information (inverse covariance). The orientation matches that of 5(a)



(c) Radius represents the profile estimate (Projected area of the object, m^2)



(d) Radius represents the profile information (inverse covariance). Information peaks correspond to air and ground observations

Fig. 5. Angular Profiles Demonstration

The mutual information properties of angular profiles are distinct from those of three dimensional bearing only point localisation [10]. Given a single observation, the next observation to maximise information gain should be 180 degrees around the profile. A sequence of adjacent observations optimised for information gain explores all angles.

5.6 Other Applications and Extensions

The method of interpreting prediction models as differential observations used here to develop a technique for developing the consistency models for angular profiles can be applied to other problems in the estimation of spatially distributed states. In particular, it could be applied to the estimation of the trajectory of near-linear features such as fences, roads and rivers presented by our field site.

Interpreting prediction models as differential observations can also be applied to temporal estimation. It is a subject of future investigation to compare this to other treatments of delayed and asequent data handling [11] and to other smoothing formulations of estimation. [12]

Interpreting spatial consistency models and temporal prediction models as differential observations (in space a time respectively) allows one to describe consistency in space and time simultaneously. This provides a method for simultaneously estimating spatially distributed random fields and providing temporal smoothing (spatio-temporal estimation). This can be compared to the spatial Kalman filtering described in [13] and [14].

It will be necessary to describe temporal process models for the angular profiles, primarily to introduce uncertainty over time.

There are difficulties involved with handling observations of the angular profile from uncertain angles. As described, the technique treats the angular states as fixed on a set of angles around the object and so observations must be subject to data association to choose the angle to update.

6 Conclusion and Future Work

In this paper we introduced our project and approach, described the vision system and environment. We introduced a theory for the estimation of angular profiles with demonstrations from simulation and field data.

In future developments we will be incorporating the image processing algorithms and observation models necessary to observe angular profiles as described here. We will be revising the decentralised data fusion system to allow greater flexibility in the choice of states associated with each feature in order to support the communication and fusion of angular profiles.

Feedback from the angular profile states and localisation states will need to be used simultaneously for information theoretic decentralised control. As discussed in section 5.5, an angular profile of a single feature has well behaved properties in entropy and mutual information, causing decentralised control algorithms to explore not only different positions in space but different angles of view. However, implementing angular profiling alongside localisation presents many challenges.

The technique of angular profiling is limited by the choice of the profiled observable. For general vision based applications it may be preferable to focus on methods for estimating the three dimensional geometric structure and colour or intensity of regions, rather than relying upon low dimensional remote observables. However, the ability of angular profiles to provide an entropic measure of angular information coverage is a relevant and beneficial feature.

Acknowledgments

This project is supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government. This project is supported by BAE Systems, Bristol, UK.

References

1. Salah Sukkarieh, Eric Nettleton, Jong-Hyuk Kim, Matthew Ridley, Ali Goktogan, and Hugh Durrant-Whyte. The ANSER project: Data fusion across multiple uninhabited air vehicles. *The International Journal of Robotics Research*, 22(7-8):505–539, 2003.
2. Nadine Gobron, Bernard Pinty, Michel M Verstraete, Jean-Luc Widlowski, and David J. Diner. Uniqueness of multiangular measurements. *IEEE Transactions on Geoscience and Remote Sensing*, 40(7):1574 – 1592, 2002.
3. Eric Nettleton. *Decentralised Architectures for Tracking and Navigation with Multiple Flight Vehicles*. PhD thesis, The University of Sydney, 2003.
4. Frank Dellaert, Steven M. Seitz, Charles E. Thorpe, and Sebastian Thrun. Structure from motion without correspondence. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:557 – 564, 2000.
5. D.T. Cole, S. Sukkarieh, A.H. Goktogan, H. Stone, and R. Hardwick-Jones. The development of a real-time modular architecture for the control of uav teams. In *The 5th International Conference on Field and Service Robotics*, July 2005.
6. Ed Waltz. *Handbook of Multisensor Data Fusion. The Principles and Practice of Image and Spatial Data Fusion*. CRC Press, 2001.
7. Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 1 of *Mathematics in Science and Engineering*. 1979.
8. Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y. Ng, Zoubin Ghahramani, and Hugh Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7-8):693–716, 2004.
9. Pen-Olof Persson and Gilbert Strang. A simple mesh generator in matlab. *SIAM Review*, 46(2):329 – 345, 2004.
10. Ben Grocholsky. *Information-Theoretic Control of Multiple Sensor Platforms*. PhD thesis, Australian Centre for Field Robotics Department of Aerospace, Mechatronic and Mechanical Engineering The University of Sydney, 2002.
11. Eric W. Nettleton and Hugh F. Durrant-Whyte. Delayed and asequent data in decentralised sensing networks. *Proceedings of SPIE - The International Society for Optical Engineering*, 4571:1 – 9, 2001. Decentralised sensing networks.
12. Robert F. Stengel. *Optimal Control and Estimation*. Dover, 1994.
13. K.V. Mardia, C. Goodall, E.J. Redfern, and F.J. Alonso. The kriged kalman filter. *Test (Trabajos de Estadstica)*, 7(2):217–285, December 1998.
14. Noel Cressie and Christopher K. Wikle. Space time kalman filter. *Encyclopedia of Environmetrics*, 2002.

Topological Global Localization for Subterranean Voids

David Silver, Joseph Carsten, and Scott Thayer

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
{dsilver,jcarsten,sthayer}@ri.cmu.edu

Summary. The need for reliable maps of subterranean spaces too hazardous for humans to occupy has motivated the development of robotic mapping tools. For such systems to be fully autonomous, they must be able to deal with all varieties of subterranean environments, including those containing loops. This paper presents an approach for an autonomous mobile robot to determine if the area currently being explored has been previously visited. Combined with other techniques in topological mapping, this approach will allow for the fully autonomous general exploration of subterranean spaces. Data collected from a research coal mine is used to experimentally verify our approach.

1 Introduction

In many parts of the world, abandoned mines present a significant environmental hazard. Toxic runoff, landslides, and subsidence are just some of the dangers presented by these structures. In the U.S. alone, there are tens of thousands of abandoned mines [3] that threaten nearby surface and subterranean operations. The first step towards combating this problem is to obtain an accurate metric survey of the mine structure. Unfortunately, in most cases an accurate survey of the mine has either been lost or never existed. Taking a new survey of the structure is often limited to inspections via boreholes, as abandoned mines are usually too dangerous for people to enter. For this reason, robots have been proposed as a method for mapping abandoned mines.

The Carnegie Mellon Subterranean Robotics group has undertaken the task of developing robotic systems that can autonomously explore abandoned mines or other hazardous subterranean voids. The initial effort led to the development of a system that can autonomously navigate and explore long stretches of a single mine portal [2]. More recent work has focussed on expanding mission profiles to include general exploration of multiple intersecting corridors. This led to a system which can detect and traverse multiple corridors [13], but can not determine when it has returned to a previously

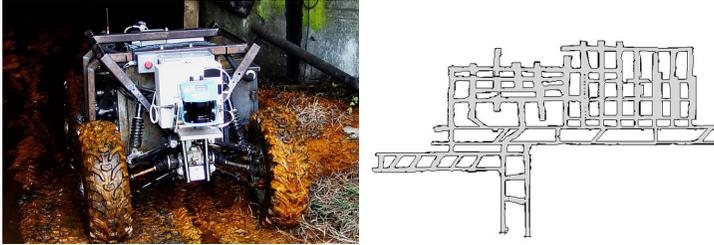


Fig. 1. Left: Groundhog, the current robotic platform of the mine mapping project. **Right:** This map was generated from data acquired during experimentation and utilizes offline globally consistent mapping techniques. It shows the highly cyclic nature of room-and-pillar mines.

visited corridor intersection from a different direction. This constraint limited the environments explored in [13] to those which did not contain loops.

This paper presents a method by which an autonomous mobile robot can identify correspondences between intersections in subterranean environments, allowing for autonomous loop closure and more general exploration. Our approach for matching intersections is based on comparisons of both 2D and 3D range data local to each intersection. The results of these comparisons are then fed to a binary classifier, which produces the probability of a match. Such a classifier can then be integrated into a complete system designed to track multiple topological map hypotheses.

The remainder of this paper discusses the relevant details of our approach. Section 2 provides background into subterranean topological exploration. Section 3 describes our technique, with experimental results presented in Section 4. We conclude with a discussion and directions for future work.

2 Subterranean Topological Exploration

2.1 Robotic Platform

Our current mine mapping platform is **Groundhog** (Figure 1), a 700 kg custom-built ATV-type robot that is physically tailored for operation in the harsh conditions of abandoned mines. Groundhog’s primary sensing consists of 2 SICK LMS-200 laser range finders mounted in front and back. Each has a 180° field of view, and is mounted on a tilt mechanism with a 60° range. Tilting each laser allows for the acquisition of 3D range data. Groundhog has been used extensively in both test and abandoned mine environments, accruing hundreds of hours of mine navigation, including 8 successful portal entry experiments in the abandoned Mathies mine outside of Pittsburgh, PA. Offline techniques have been used to generate globally consistent, large-scale maps based on log data from these experiments. For a thorough overview of the Groundhog system, see [2].

2.2 Topological Representations

Topological representations coincide nicely with the inherent structure of room-and-pillar mines, which consist almost exclusively of narrow corridors and corridor intersections (see Figure 1). A topological map is a graph representation of an environment. The nodes of the graph correspond to distinct locations in the environment, and the edges correspond to direct paths between two such locations. For mines, nodes and edges correspond to intersections and corridors, respectively. This approach was used in [10] to allow a robot to traverse known mine environments. Topological maps have also proven useful in robotic exploration tasks of unknown environments [9]. Unexplored edges in a topological map correspond to unexplored regions of the environment, thus providing a mechanism for determining which region of the environment to explore next.

The key components of a system designed for autonomous topological exploration are:

- A method for traversing an edge in the environment until a node is reached.
- A method for detecting a node and its associated edges in the environment.
- A method for determining whether the currently sensed node has been visited before, and if so which previously visited node it corresponds to (this is the problem our current work strives to solve).
- A representation of the topological map and its associated uncertainty.

The first two components have been previously developed and tested in subterranean environments, as described in the following sections.

2.3 Edge Traversal

Edge traversal is the first necessary component for autonomous topological exploration. While traversing a single corridor, Groundhog utilizes the Sense-Plan-Act (SPA) framework. While stationary, Groundhog tilts one of its lasers to accumulate 3D range data from the space in front of it. This 3D point cloud is used to generate a 2.5D cost map. Next, a goal pose is chosen that will further Groundhog's progress down the corridor (or turn it into a new corridor). A path is planned to the goal pose by feeding the cost map into a nonholonomic motion planner described in [13]. The planned path is then traversed by Groundhog, and the whole process repeated. For a more detailed description, see [2, 13].

2.4 Node Detection

A method for node detection is also critical to topological exploration. Groundhog detects intersections in its environment by searching for nodes of the *generalized Voronoi diagram* (GVD) [6]. Edges of the GVD represent sets of points equidistant from 2 objects. Nodes of the GVD represent points

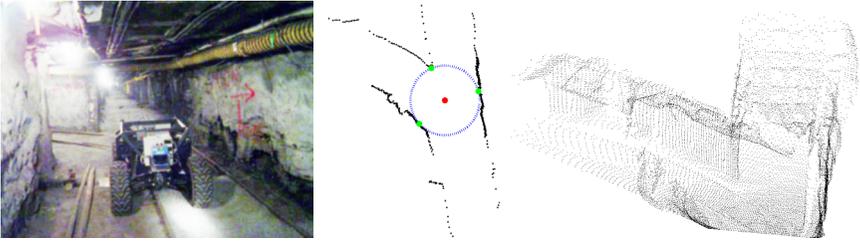


Fig. 2. The data collected at each node. **Left:** Groundhog approaching an intersection. **Center:** the 2D range data collected, as well as the detected node location and radius. **Right:** the 3D range data collected.

equidistant from 3 objects. While traversing an edge, potential GVD nodes are detected using a procedure described in [15]. Each potential node is then tracked until Groundhog drives through the intersection to which the node corresponds. The purpose of this extra traverse is to obtain a 2D map of the environment around the node with a full 360° coverage, as opposed to the 180° field of view of Groundhog's lasers. Such coverage is achieved by combining multiple laser scans from different vantage points. This 360° coverage is necessary to determine whether the intersection just traversed is worth exploring; if the end of a corridor is already within sensor range from the intersection itself, it may not be worth further exploration. This procedure also eliminates large concavities that can appear as intersections when first detected. After a node has been detected and verified, a 3D scan of the intersection is taken, and Groundhog continues its exploration. The Voronoi radius (equidistance value between the node and the objects that formed it), 2D map, and 3D scan (Figure 2) are all stored for later use.

2.5 Framework for Topological Uncertainty

For successful topological exploration, a robot must be able to determine if a given node has been previously visited. This determination can be made based purely on the local topology [7], or by combining topological information with range data or data on nearby features. The techniques described in this paper follow the latter approach.

Regardless of the specifics of the node matching approach, its output will be uncertain. There may be multiple previous nodes which match the current node closely enough to be considered a possible match, and the fact that the node may never have been previously visited adds additional uncertainty. A framework is necessary for dealing with this uncertainty until the ambiguity can be removed. A widely adopted approach is to maintain multiple hypotheses as to the correct topology of the environment [8, 11, 16]. The robot can then either take actions designed to explicitly remove the ambiguity, or maintain multiple hypotheses until the natural exploration behavior of the robot

produces enough additional information. In either case, the correct framework can add additional robustness on top of the chosen node matching scheme.

3 Subterranean Node Matching

We approach node matching as a topological global localization problem. When a robot arrives at a node N_i along edge E_i , it can localize itself to a discrete subset of all possible states in the world (the set of states located at a node, oriented along an edge). If the robot can properly match N_i and E_i to a previously visited N_j and E_j , then it will have relocalized itself. If the robot can properly determine that N_i has not been visited before, it will still have localized itself to the correct state, albeit a state that has not previously been visited.

To determine whether the current node N_i matches a previous node N_j , we use a hybrid approach based on both local topology and range data (Figure 3). Local topological data is rarely descriptive enough to determine explicitly whether two nodes match. However, it requires essentially no preprocessing: it is computationally inexpensive to determine whether N_i and N_j are of the same degree. For this reason, local topological data is used to pare down the number of prospective matches.

For similar reasons, 2D as well as 3D range data is used. While 2D range data is usually not descriptive enough to make an explicit determination, it is much cheaper to process than the full 3D point cloud, and can further pare down the number of prospective matches. 2D data has another advantage under our current setup: as described in Section 2.4, 2D information is collected a full 360° around the intersection. The additional coverage offered by 2D data often proves quite useful in determining final matches.

A common approach for determining whether a robot is revisiting a location is to explicitly search for features in the local environment, and try to match these features to those that have been previously detected. However, subterranean spaces provide a unique challenge for feature extraction. While such spaces are often feature rich, it is hard to characterize the features exhibited. Features can vary greatly in both type and scale, and so a more robust approach is needed. For this reason, our approach compares nodes in a manner which does not require explicit extraction of predetermined features.

3.1 Comparison of Topological Properties

The first step of our node matching scheme is to use the topological properties of the detected node N_i to eliminate as many nonmatching nodes N_j as possible. These topological properties are the degree of the node and its associated Voronoi radius. Another property we explored was the relative orientations of the edges associated with the node. Previous work [12] has shown these relative orientations to be quite susceptible to noise. This lack of robustness

```

CompareNodes( $N_i, N_j$ ):
if  $N_i$ .degree  $\neq N_j$ .degree then return 0
 $d \leftarrow N_i$ .degree
if  $|N_i$ .vRadius  $- N_j$ .vRadius  $| > T_r^d$  then return 0
 $P_2 \leftarrow$  PositionOffsetBetweenNodes( $N_i, N_j$ )
 $R_2 \leftarrow$  MinimumErrorRotation( $N_i, N_j, P_2$ )
( $MSE_{2D}, P_2, R_2$ )  $\leftarrow$  TrICP2D( $N_i$ .2D,  $N_j$ .2D,  $P_2, R_2$ )
if  $MSE_{2D} > T_e^d$  then return 0
( $MSE_{3D}, P_3, R_3$ )  $\leftarrow$  TrICP3D( $N_i$ .3D,  $N_j$ .3D,  $P_2, R_2$ )
 $E \leftarrow$  FormErrorVector( $N_i, N_j, P_3, R_3$ )
return LogisticRegression( $E, d$ )

```

Fig. 3. Pseudocode for our node matching procedure

was also observed in our own experiments, and therefore this property was not used. Instead, if N_j has a different degree than N_i , or the difference in observed radii is more than a threshold T_r , N_j is eliminated as a candidate match. T_r is set relatively high, so as to ensure that no correct matches are ever thrown out, while eliminating as many incorrect matches as possible in a computationally inexpensive manner.

3.2 2D Map Matching

The next phase of node matching is to compare each node’s 2D local map. Before the 2D maps can be compared, they must be properly aligned. Alignment of 2D point sets can be achieved using the Iterative Closest Point (ICP) algorithm [4]. ICP assumes that each point in the data set corresponds to the closest point in the model set. These correspondences are used to compute the transformation between the two sets that minimizes the Mean Squared Error (MSE). The correspondences are then recomputed, and the process iterates until convergence.

Due to the manner in which our 2D maps are constructed, the assumption that every point in the data set has a corresponding point in the model set is often violated to a degree that degrades performance. Therefore, the Trimmed Iterative Closest Point algorithm (TrICP) [5] is used instead. The key difference between ICP and TrICP is that TrICP assumes that only a proportion ξ of the points in the data set correspond to points in the model set. At each iteration, only ξK of the K points in the data set are used. The ξK points used are those with the smallest squared distance to their corresponding point in the model set. When unknown beforehand, ξ can be automatically set by minimizing the function

$$\psi(\xi) = MSE(\xi)\xi^{-(1+\lambda)} \quad (1)$$

where $MSE(\xi)$ is the MSE of the ξK points with the smallest squared distance to their corresponding point in the model set. The parameter λ balances the tradeoff between using more points and increasing MSE. In [5], $\lambda = 2$.

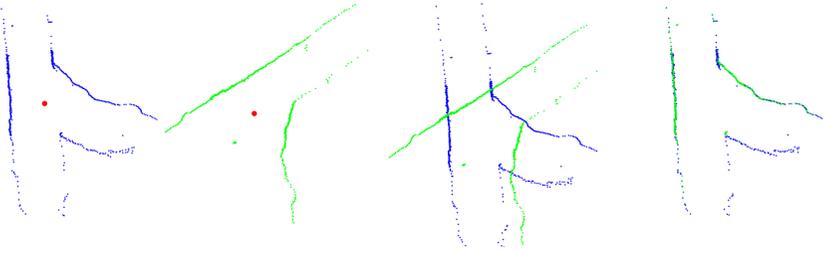
Both ICP and TrICP require a fairly accurate initial alignment in order to converge correctly. By framing node matching as a global localization problem, it is assumed that there does not exist a good long term estimate of metric position. In practice, this is usually the case, as Groundhog's online position estimation is not stable over long distances (accurate metric maps are produced offline). Since Groundhog's perceived metric position can not be used for an initial alignment, the locations of the nodes themselves are used. Since each node is embedded into the environment, if the two local maps are really of the same intersection, then the location of the Voronoi node in each map corresponds to the same point in space, represented in different coordinate frames. Setting the origin of each local map to be the corresponding Voronoi point thus produces an initial alignment in position. However, the orientation of each map relative to the node is still unknown. To fix the orientation, TrICP is run 8 times, with the initial orientation of one map relative to the other equally spaced at 45° intervals. TrICP is able to overcome such large errors in initial orientation because the error in initial position is small. The final alignment that results in the smallest MSE is selected as the correct 2D alignment (Figure 4(a)).

The MSE of the final alignment (after recomputing ξ) is compared against a threshold T_e . Just as with T_r , T_e is set to eliminate as many false matches as possible, while not eliminating any correct matches.

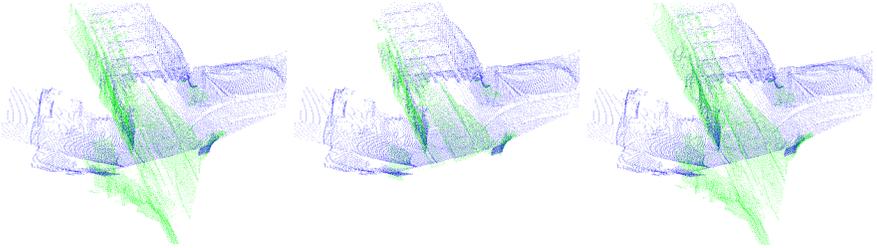
3.3 3D Map Matching

The last phase of node matching uses the 3D range data gathered after each node is detected. As with the 2D data, the 3D data must first be properly aligned. The 3D alignment is also achieved using TrICP. The initial 3D alignment used for TrICP is based on the final 2D alignment. Using the 2D alignment between the candidate nodes, and the known position of each node relative to the origin of the 3D scan, an initial 3D alignment is computed that is fairly accurate in x , y and yaw. Just as running TrICP with only an initial x and y allows the 2D alignment to converge to the correct orientation, running TrICP with an initial x , y and yaw allows the 3D alignment to converge to the correct z , roll, and pitch (Figure 4(b)).

In this phase, TrICP is run with one modification. Normally, ξ is computed according to (1) once during the first iteration. Thus, ξ depends heavily on the initial alignment. Since the initial alignment could have significant error in 3 of the 6 degrees of freedom, ξ must be occasionally recomputed. For this purpose, an additional loop is added around TrICP. After TrICP successfully converges, ξ is recomputed based on the final alignment. The final alignment is then fed back into TrICP as the new initial alignment. This process repeats until the value of ξ converges.



(a) 2D alignment: Each map is centered around its Voronoi node, and then one map is rotated relative to the other to find the minimum MSE alignment.



(b) 3D alignment: the 2D alignment is used as the initial 3D alignment (**left**). The ξN_d closest points (**center**) are then used to find the final alignment (**right**).

Fig. 4. 2D and 3D alignment of range data at an intersection

After each 3D alignment is complete, an error vector $E = \{e_1, \dots, e_n\}$ is produced for each prospective match $N_i \leftrightarrow N_j$. The error vector consists of both 2D and 3D error measures. The 2D metrics are used despite the availability of 3D metrics, due to the 360° coverage of 2D data. In addition to MSE, additional error metrics based on the normal vectors of the 3D range data are used. This error metric is especially useful for classifying potential matches with a small ξ . The specific error vector used is described in Section 4.

3.4 Classification

After an error vector has been produced, the final task is to determine as accurately as possible whether or not N_i matches N_j . This can be viewed as a binary classification problem, with matching and non-matching classes. One approach to binary classification is *logistic regression* [1]. Under this approach, the probability of a match is computed from the error vector E as

Table 1. The results of each phase of node matching

| Stage of Comparison | # of Incorrect Matches Remaining | # of Correct Matches Remaining |
|---------------------|----------------------------------|--------------------------------|
| Original Dataset | 1962 | 108 |
| Degree Matching | 1002 | 108 |
| Radii Difference | 588 | 108 |
| 2D MSE | 173 | 108 |
| Logistic Regression | 23 | 108 |

$$P(N_i \leftrightarrow N_j | E = \{e_1, \dots, e_n\}) = \frac{1}{1 + \exp(-z)} \quad (2)$$

$$z = w_0 + w_1\Phi_1(e_1) + w_2\Phi_2(e_2) + \dots + w_n\Phi_n(e_n) \quad (3)$$

W is vector of weights $\{w_0, \dots, w_n\}$, computed from training data using a maximum likelihood formulation. Each Φ_i is constructed as a classifier based on an individual element of the error vector. Our approach constructs each Φ_i as a Gaussian classifier of the i^{th} element of the error vector

$$\Phi_i(e_i) = \frac{\mathcal{N}(e_i, \mu_i^+, \sigma_i^+)}{\mathcal{N}(e_i, \mu_i^+, \sigma_i^+) + \mathcal{N}(e_i, \mu_i^-, \sigma_i^-)} \quad (4)$$

where μ_i^+ and σ_i^+ are the mean and standard deviation of the i^{th} element of E over matches, μ_i^- and σ_i^- are the mean and standard deviation over non-matches, and $\mathcal{N}(e, \mu, \sigma)$ is the Gaussian probability density function.

4 Experimental Results

4.1 Data Collection

To test our node matching approach, data was collected from the Bruceton research coal mine near Pittsburgh, PA. The dataset consists of the same topological, 2D, and 3D data that would be collected during autonomous exploration and intersection detection. 3D range data was downsampled to one point per 5cm voxel [14], to ensure equivalent resolution from multiple vantage points and to provide a significant decrease in computation. For each intersection, data was collected from each corridor leading into the intersection. Data was gathered from 46 different intersection/corridor combinations, resulting in 2070 possible matches. Of these, 108 are correct matches. The results of each phase of node matching are shown in Table 1.

4.2 Topological Matching

Of the 2070 possible matches, 960 (46%) can be immediately eliminated, because the degree of one node does not match the degree of the other node.

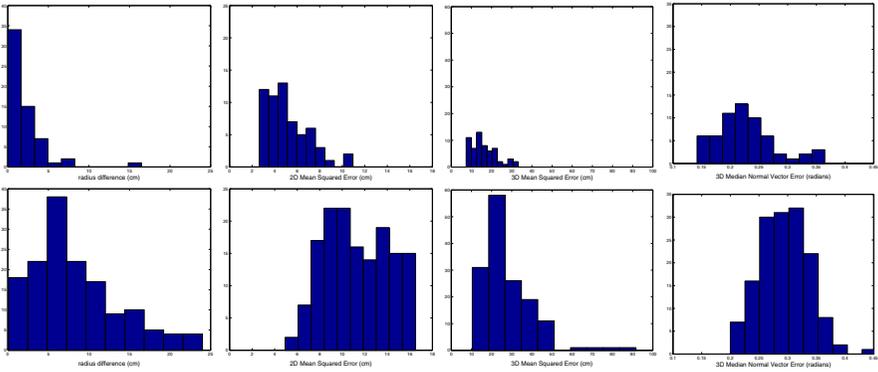


Fig. 5. The distributions of each value of the final error vector over nodes of degree 3. The distributions over matches are shown on top, and non-matches on the bottom

Next, matches are eliminated based on the Voronoi radius. To make it as unlikely as possible that any correct matches are eliminated in this phase, T_r is set at 1.5 times the maximum difference in Voronoi radii observed in a correct match. To take into account the differences in various types of intersections, a different threshold T_r^d is chosen based on the degree d of the node. Solely based on radii thresholding, 1219 (59%) of the possible matches can be immediately eliminated. Combining radii thresholding with the enforcement of degree equality eliminates 1374 (66%) of the possible matches. Thus, approximately $2/3$ of prospective matches are eliminated almost immediately.

4.3 2D Matching

After thresholding on topological properties, the next phase is to align the 2D range data, and compare the MSE against a threshold T_e . For 2D TrICP, a λ value of 2 was used. As with radius thresholding, a different T_e^d is used for each node degree d , and each T_e^d is set at 1.5 the maximum observed MSE in a correct match. Of the 2070 possible matches, 1573 (76%) can be eliminated solely based on 2D MSE thresholding. By also only considering matches that passed the topological matching phase, 1789 (86%) matches are eliminated. Thus, the relatively inexpensive topological and 2D matching phases are able to quickly eliminate all but about 14% of the possible matches.

4.4 3D Matching

Next, the 3D range data associated with the remaining prospective matches is aligned. For 3D TrICP, a λ value of 1.5 was used. After 3D alignment is completed, the final error vector E is formed. An error vector consisting of the following fields has so far produced the best results:

- The difference in Voronoi Radius

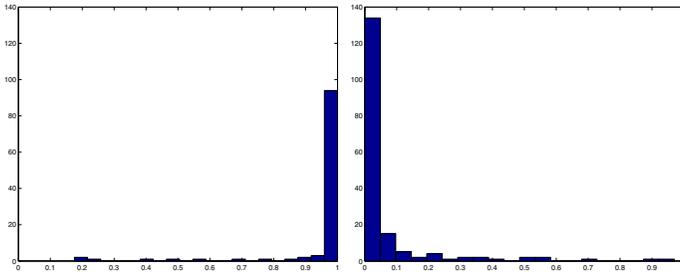


Fig. 6. Output of the final classifier on matches (**left**) and non-matches (**right**)

- The 2D MSE of the $\xi_2 K_2$ points with the smallest error
- The 3D MSE of the $\xi_3 K_3$ points with the smallest error
- The median angle between normal vectors of the $\xi_3 K_3$ points with the smallest error

Example distributions of these 4 elements over both correct and incorrect matches are shown in Figure 5.

4.5 Final Classification

After the error vector has been computed, it is fed into the classifier to compute a final match probability. For this experiment, the classifier was trained over the set of all matches that were not eliminated by thresholding tests. To help reduce the chance of a false negative, correct matches were weighted twice as heavily as incorrect matches during training. As with all other phases, a separate classifier is used for each possible node degree.

The distributions of the final probabilities over both correct and incorrect matches are shown in Figure 6. Thresholding the final probability at 0.1 results in all 108 correct matches still being considered, with only 23 remaining false positives. This accuracy is more than sufficient for use within a multi-hypotheses topological framework.

5 Conclusion

In this paper, we have presented a method for approximating the probability that two corridor intersections in a subterranean void match. Such an approach can be used by an autonomous mine mapping robot to determine when it is revisiting an intersection. This approach, in conjunction with other topological techniques, will allow for the full autonomous exploration of mine environments, including autonomous loop closure.

Future work will focus on making our node matching technique robust to the point that multi-hypotheses tracking will almost never be necessary. One method for achieving this would be to use multiple 3D scans from each visit

to a node to provide the same 360° coverage that the 2D scans achieve. Also, more intelligent means of computing T_r and T_e will be explored. Further, the possibility of more descriptive 3D error metrics will be investigated.

References

1. A. Agresti. *Categorical Data Analysis*. Wiley-Interscience, 2002.
2. C. Baker, A. Morris, D. Ferguson, S. Thayer, C. Whittaker, Z. Omohundro, C. Reverte, W. Whittaker, D. Hähnel, and S. Thrun. A Campaign in Autonomous Mine Mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, 2004.
3. J. Belwood and R. Waugh. Bats and mines: Abandoned does not always mean empty. *Bats*, 9(3), 1991.
4. P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.
5. D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *Proc. Int. Conf. on Pattern Recognition*, 2002.
6. H. Choset and J. Burdick. Sensor based planning, part II: Incremental construction of the generalized voronoi graph. In *Proc. of IEEE Conference on Robotics and Automation*, pages 1643 – 1648, Nagoya, Japan, May 1995. IEEE Press.
7. H. Choset and K. Nagatani. Topological simultaneous localization and mapping (slam): towards exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, Apr. 2001.
8. G. Dudek, P. Freedman, and S. Hadjres. Using local information in a non-local way for mapping graph-like worlds. In *Proc. of the 13th International Joint Conference on Artificial Intelligence*, 1993.
9. G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Robotic exploration as graph construction. *Trans. on Robotics and Automation*, 7(6):859–865, Dec. 1991.
10. E. Duff, J. Roberts, and P. Corke. Automation of an underground mining vehicle using reactive navigation and opportunistic localization. In *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2003.
11. B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *IEEE International Conference on Robotics and Automation*, 2004.
12. B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset. Hierarchical simultaneous localization and mapping. In *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, volume 1, pages 448–453, Oct. 2003.
13. A. Morris, D. Silver, D. Ferguson, and S. Thayer. Towards topological exploration of abandoned mines. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.
14. J. Rossignac and P. Borrel. *Multi-Resolution 3D Approximations for Rendering Complex Scenes.*, pages 455–465. Springer-Verlag, 1993.
15. D. Silver, D. Ferguson, A. Morris, and S. Thayer. Feature extraction for topological mine maps. In *IEEE/RSJ Conf. on Intelligent Robots and Systems*, 2004.
16. N. Tomatis, I. Nourbakhsh, and R. Siegwart. Hybrid simultaneous localization and map building: Closing the loop with multi-hypotheses tracking. In *IEEE International Conference on Robotics and Automation*, 2002.

A Navigation System for Automated Loaders in Underground Mines

Johan Larsson^{1,2}, Mathias Broxvall², and Alessandro Saffiotti²

¹ Atlas Copco, Örebro, Sweden

johan.larsson@tech.oru.se

² Center for Applied Autonomous Sensor Systems, Örebro University, Örebro, Sweden

{mbl, asaffio}@aass.oru.se

Summary. For underground mining operations human operated LHD vehicles are typically used for transporting ore. Because of security issues and of the cost of human operators, alternative solutions such as tele-operated vehicles are often in use. Tele-operation, however, leads to reduced efficiency, and it is not an ideal solution. Full automation of the LHD vehicles is a challenging task, which is expected to result in increased operational efficiency, cost efficiency, and safety. In this paper, we present our approach to a fully automated solution currently under development. We use a fuzzy behavior-based approach for navigation, and develop a cheap and robust localization technique based on the deployment of inexpensive passive radio frequency identification (RFID) tags at key points in the mine.

Keywords: Mining vehicles, fuzzy logic, hybrid maps, behavior-based navigation, autonomous robots, RFID

1 Introduction

In underground mining, LHD (Load-Haul-Dump) vehicles are typically used to transport ore from the stope or muck-pile to a dumping point. A number of reasons have led to the desire to automate the operation of LHD vehicles, thus removing the need to have a human operator constantly on-board the vehicle. First, a mine is generally not offering the best environment conditions for humans. Second, the nature of this task is such that the vehicle and its operator are continuously subject to the risk of being hit or buried by falling rocks, since the load operation is performed in unsecured areas. Third, an automated LHD vehicle could allow reduced operation costs and increased productivity. Fourth, automatic control of the LHD vehicle could lead to less mechanical strain, which would in turn reduce the maintenance costs.

In some mines, tele-operation of LHDs is used to gain safety, but this often leads to reduced productivity since a remote operator is not able to drive the vehicle as fast as an on-board operator. In addition the maintenance cost of the vehicles tends to increase with tele-operation. These facts have led to the desire to automate the whole

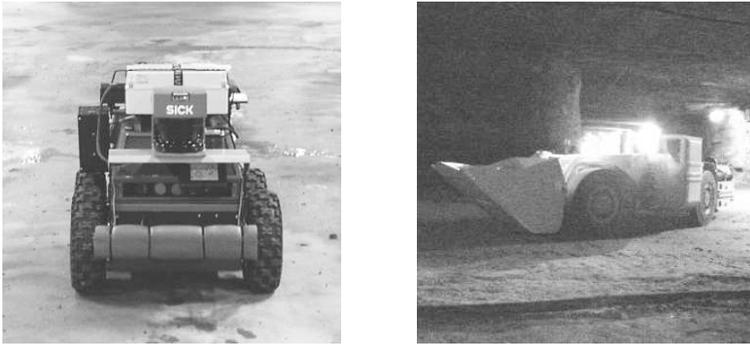


Fig. 1. Left: The ATRV-Jr research robot, carrying the two main sensors used in our experiments, the SICK laser scanner and the RFID tag reader (white box). Right: An LHD vehicle.

tasks performed by the LHD vehicles, or to use a combination of performing some tasks autonomously and others by tele-operation. Since the greatest part of the time in the work-cycle is spent tramping (moving or Hauling), this is the part that is most desirable to automate. This paper addresses the development of a control system that allows the autonomous navigation of an LHD vehicle in a mining environment.

In order to be commercially viable, any solution for the autonomous navigation of LHD vehicles should meet a number of requirements. The solution should require minimal setup and maintenance effort. It should require only little additional infrastructure on the mine, or possibly none at all. It should not require that an accurate geometric map of the mine is hand-coded into the system. It should afford navigation speeds comparable to the ones reached through a human operator (approximately 30 Km/h). Finally, it should guarantee extremely high safety and reliability, that is, faults should have low probability, and there should be mechanisms able to detect these faults and to stop the vehicle.

This paper, present our steps toward the development of a system for automated navigation of LHD vehicles in underground mines that satisfies the above requirements. Our system uses a coarse topological map to represent the mine, and a behavior-based approach to navigate inside the mine using a sequence of reactive follow-tunnel behaviors. No global metric localization is required. Instead, the vehicle uses data from a laser range scanner to maintain its relative position and orientation inside each tunnel, and an intersection recognizer to assess its topological position in the map. Intersections are recognized by a combination of odometry, laser signature, topological structure, and RFID tags. The two main features of this approach are: (1) small setup and maintenance costs, since it only requires to place a passive RFID tag at each tunnel intersection; and (2) high reliability, thanks to the redundancy of the information used.

Our development methodology is in two phases. In the first phase, with focus on localization, we develop our techniques and algorithms on a small research outdoor robot, starting from an existing framework for autonomous navigation [11]. The experiments in this phase are performed in long corridors inside a building, and in a

test underground mine. In the second phase, we will port the developed algorithms to a real 30 ton LHD vehicle manufactured by Atlas Copco, and run experiments in the test mine. Figure 1 shows the two experimental vehicles used in our development. This paper reports about the first phase; the second phase will start in the next few months.

The rest of this paper is structured as follows. In the next section, we briefly overview some related systems for autonomous navigation in underground mines. In Sections 3 and 4, we discuss our approach to localization and to navigation, respectively. In Section 5 we present some preliminary experiments performed on the research robot in both the indoor environment and the test mine. Section 6 concludes.

2 Related Work

Several solutions have been suggested and evaluated for automation of the tramping (movement) of the LHDs. Some of these have been in use for quite some time now, while others have recently emerged pushed by the research in the area of mobile robotics.

2.1 Older Solutions

Several solutions to autonomous tramping have been used in mines around the world for quite some time now. These have all been based on some infrastructure that guides the vehicle. Independent of the type all infrastructure based guidance solutions have several drawbacks, such as installation cost, maintenance cost, and inflexibility. Examples of what has been used are inductive wires [5], light ropes and reflexive tape. Common to these examples are the time and cost of installation, while the light rope also suffer from maintenance cost, and unavailability due to damages created by blasting nearby.

These systems also suffers from another major drawback: none of them allows high speed tramping. A manual operator drives the vehicle at its top speed, which is usually somewhere between 20–30 km/h, while the guidance solutions above rarely or never provide possibilities to travel faster than fractions of the top speed. This is due to the fact that all of the line following systems have difficulty of gaining significant look-ahead, since they only sense the line at the current position of the vehicle or slightly ahead of the vehicle.

Experiments with infrastructure-less guidance using ultrasonic sensors to detect the tunnel walls have been performed successful at low speed [12], [10], but the difficulties to get the necessary high resolution look-ahead prevented this system from being able to do any high-speed navigation.

Finally none of the systems above utilizes any form of obstacle detection, which is another drawback in a sometimes unpredictable mining environment.

2.2 Current Products and Recent Solutions

A more flexible system of infrastructure based guidance is used in the LKAB mine in Kiruna, Sweden [13]. This system is based on a bearing only laser scanner that measures the angle to reflexive tapes on the tunnel walls, and allows the vehicle to operate at full speed. The drawback of this robust and highly reliable system is the need to install the reflexive tapes, and to measure the position of the same to be able to integrate them into the guidance map. This system is more flexible than the ones mentioned earlier since once the reflexive tapes are installed and integrated in the guidance map, the path to be followed by the vehicle can be changed in software.

An infrastructure-less guidance system is described in [8]. This system solely depends on dead reckoning, angle/distance laser scanner and the natural landmarks in the mine. During automatic tramping a five-meter section of the scanned tunnel profile closest to the vehicle is compared to a map with known profiles and the position can thus be established. The map, which is a polyline representation of the tunnel wall, on a specific height above the floor (the height the laser scanner is mounted on the vehicle) is created by a teaching procedure. During the teaching the vehicle is driven manually in the tunnel allowing the laser scanners to register the profile of the tunnel wall on each side of the vehicle. The scanner produces 181 measurements per scan, one each degree, but only the ten left- and rightmost are taken into account. These measurements are then fused into a polyline representation of the tunnel wall with the average distance of 10 cm between the points. With this system tramping velocity comparable with human drivers has been achieved with LHD and velocities up to 40 km/h have been tested on mine trucks. Although this system does not need any extra infrastructure for the navigation, it has the drawback that the vehicle has to be driven manually through every path, before it can run there autonomously.

In [4] and [9] an experimental setup of a test track, a mine created by shade cloth, is described and used to evaluate a reactive guidance and navigation system of a LHD. The guidance system utilizes laser range scanners and dead reckoning, together with a nodal map representation of the test track. The 300 m long test track consists of sharp corners, intersections, a hall, and a loop. No extra infrastructure to guide the vehicle is installed. The results of the experiments show that the combination laser range scanner and reactive guidance is a feasible way to perform mine navigation. The test vehicle successfully navigated through the test track for up to one hour at a time without human interaction. Regarding the important issue of speed the experiments showed that the control system is able to run the vehicle at the same speed as an experienced human driver. With this particular LHD the maximum velocity of 18 km/h was utilized at parts of the test track. The only situation in which the control system did not manage to equal the human driver was encountered at sharp intersections, where the control system could not see around the corner. Neither can the human operator, but after a few test runs the driver remembered what the tunnel looked like around the corner, and therefore could approach the corner more aggressively. This approach can obviously be implemented in the control system as well by adding driving hints to the map.

3 Localization

In order to fulfill its navigation task the autonomous vehicle needs some form of map, as well as some means of localizing itself within this map. Because of the cost and accuracy problems with a full metric map we choose to use a hybrid map which augments a topological map with some metric information [1]. This topological map consists of a number of nodes (junctions and positions in tunnels) and edges (traversable paths between the nodes). This topological map can also be augmented with some metric information such as approximate tunnel width and length *when available* but the system functions also without such information. For an example of such a topological map augmented with tunnel lengths see Figure 2a.

One of the strengths of using only a topological map is that it can be constructed at little cost and it can easily be updated when the environment changes. By only providing a topological description there is no constraint on the actual layout of the environment: the map provided in Figure 2a could just as well consist of nodes and tunnels through multiple levels of a mine.

The localization used within the loaders consists of two parts, a topological localization which gives information about which edge is currently being traversed or which node has just been reached, and a metric localization which indicates where the vehicle is positioned within the current tunnel. The purpose of the latter is primarily for providing the needed parameters to the reactive behaviors used for tunnel traversal.

3.1 Metric Localization

We compute three types of metric information: longitudinal position along the tunnel, lateral position inside the tunnel, and orientation with respect to the tunnel. The former is used to increase the robustness of the topological localization; the latter are needed by the “FollowTunnel” reactive navigation behavior.

For the lateral localization and orientation within a tunnel we use a laser range scanner. The scanner produces 181 measurements per scan, one per degree and is mounted in the front of the vehicle. Our algorithm processes these scans to provide the rest of the system with the parameters of the detected tunnel segments, together with a certainty factor that depends on the number of reflected laser readings. In addition, our algorithm uses the laser data to detect obstacles for collision avoidance. Our target sampling rate for tunnel and obstacle detection is 75 Hz.

In order to achieve this rate, we have used a modified Hough transform [7] on the 1D laser data to identify pairs of line segments. By allowing some flexibility in the line segments it is also possible to operate in curved tunnels. By only checking for pairs of lines separated by 180 degrees and with a certain minimum and maximum separation it is possible to accurately identify tunnels around the vehicle. Our implementation yields execution times of about 2 ms, well within the requirements for fast navigation. The low execution time is achieved mainly by discarding irrelevant laser points before the Hough transformation is made, but the fact that we do not have to search the entire Hough space for tunnel walls also contributes. Apart from providing

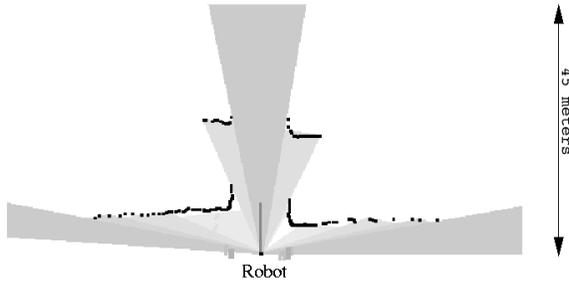


Fig. 3. Identifying open areas and tunnels with a laser range finder

information about the currently traversed tunnel these transformed laser readings are also used for identifying side tunnels which are used in the topological localization.

Figure 3 gives an example of extraction of the edges and direction of a tunnel from laser range data. The data refers to a situation in the test mine, where the robot was about to enter a new tunnel. In the figure, the robot is seen from the top, placed at the center bottom and pointing upward. Laser measurements shorter than the maximum range (80 m) are indicated by black dots. The light gray cones show the identified open areas. The dark gray line indicates the direction of the tunnel segment in front of the robot, found by our algorithm. Notice that the tunnel could be correctly identified even though its walls are interrupted by the entrances of many side tunnels.

The longitudinal position along the tunnels is computed by odometric update, where odometry is given by a combination of scan matching and wheel encoders. The encoder data are very imprecise since the wheel diameter can change by a large amount depending on tire pressure, loaded weight, and tire consumption. However, the combination with the topological localization gives sufficient accuracy for our purposes.

3.2 Topological Localization

The main input to topological localization is node detection and identification: this tells us that we have completed the traversal of one edge and arrived at a node.

To do node detection, we use a redundant combination of four sources of information: (1) longitudinal metric localization inside the tunnel, that tells us when we are near or past the next junction; (2) recognition of the laser signature of a junction from the laser data; (3) recognition of the topological structures, e.g., counting number of side tunnels; and (4) detection of an RFID tag. The latter also gives us the unique ID of the junction, which should match the one in the topological map.

For the first two sources of information (1), (2) we use standard robotic techniques with the normal caveats regarding robustness and deployment. Although by

themselves these are not sufficient for our application we use them as a supplementary source of localization information to further increase the robustness of the two other techniques outlined below.

The third (3) source of information can be useful in areas in which the density of intersections is high. In practice, we identify the side tunnels through the laser system and compare the number of observed side tunnels with the topological map, much like a human driver would given the description “take the second turn on the left”. Note that failures may occur, e.g., if the entrance of a side tunnel is temporarily obstructed by another vehicle.

Perhaps the most peculiar of the above components is the use of RFID tags which is used in the last information source (4). This is a flexible and low cost solution for marking up the environment with standardized radio frequency identification tags.

These tags are a low cost, standardized solution for storing and retrieving data remotely in small tags that have found uses in various fields e.g., inventory tracking, automobile locks, animal tracking and quality control. There exists many different forms of RFID tags with sizes varying from 0.4mm square and up, having reading ranges in the order of a few centimeters up to 8m for passive tags. Battery powered (active) tags have reading ranges in the order of hundreds of meters and typical life lengths of a couple of years. Tags are available for as little as 0.40 USD and expect to drop in price to as little as 0.05 USD as the use of RFID tagging is growing in the industry.

For the application of autonomous navigation in mines we use passive RFID tags in key junctions and equip the LHD with a tag reader allowing us to verify the localization at key points. The deployment of tags can easily be done by untrained staff and noting the position of the tags in the nodes of a simple topological map of the environment is easy.

4 Navigation

The navigation system is organized in the three-layer hierarchical structure represented in Figure 4. The main idea here is to use a coarse topological planner to decide a sequence of tunnels and junctions to traverse, and a set of fuzzy behaviors to perform fast and robust reactive navigation within each tunnel segment.

The bottom layer includes the low-level control and sensor processing algorithms, including the odometry system and the processing of laser data described in Section 3.1.

The middle layer implements a fuzzy behavior-based system. Fuzzy behaviors are easy to define and they provide robustness with respect to sensor noise, and to modeling errors and imprecision [2]. The behaviors that we use were originally developed for indoor, low-speed navigation [11].

The main behavior used in our system is the “FollowTunnel” behavior, which takes as input the parameter (orientation and lateral position) of the tunnel extracted from the laser data as explained above. Other behaviors used in our development

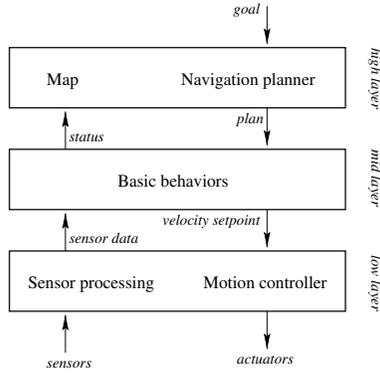


Fig. 4. Hierarchical structure of the control software

include “Avoid” to perform obstacle avoidance, and “Orient” to orient in the direction of the tunnel when entering a new one.

We only needed to modify slightly the original behaviors in order to make them work in our setup and to navigate at our robots topspeed 1.7 m/sec, or about 6 Km/h — the original behaviors were tuned for top speeds of about 0.3 m/sec. Thanks to their qualitative nature, fuzzy behaviors are prone to be transferred from one platform to another with few modifications, see [6]. However, we expect that major changes will be needed when we move to the real LHD vehicle, which is characterized by more complex dynamics and kinematics, less clearance on the sides, and speed up to 30 Km/h.

At the top level, the navigation planner relies on the topological localization described earlier, and decides what sequence of behaviors should be activated in order to reach the given target location. Our planner is based on standard search techniques, and it generates a reactive navigation plan in the form of a set of “situation \rightarrow behavior” rules. These types of plans are called behavioral-plans, or B-Plans [11].

To exemplify the operations of the complete system we consider the topological map from Figure 2. Assume that the vehicle starts at at the junction j_6 facing in the direction of tunnel t_7 and is given the goal to move to j_5 . The topological planner will then generate the following behavioral plan which will be executed:

```

IF obstacle_near                               THEN Avoid()
IF nextNode(j4) AND NOT oriented(t7)          THEN Orient(t7)
IF nextNode(j4)                                THEN Follow(t7)
IF nextNode(j5) AND NOT oriented(t4)          THEN Orient(t4)
IF nextNode(j5) AND oriented(t4)              THEN Follow(t4)
IF nextNode()                                  THEN Still()

```

Avoid, Orient, Follow and Still are fuzzy behaviors, activated according to the fuzzy predicates `obstacle_near`, `nextNode` and `oriented`. j_4 , j_5 , t_4 and t_7 are control system representations of objects in the map, for details see [11].

The laser scanner gives readings which are fed to the avoid-obstacle behavior and used to update the parameters of the current tunnel for the follow behavior. As the vehicle moves these two behaviors make the vehicle follow the center of the tunnel t_4 as long as the topological localization does not signal that the junction j_4 has been reached. When the junction j_4 is detected as described earlier another tunnel is added to the local expectations on the right side, and the laser is used for localizing its exact position and the orient behavior starts up. This behavior uses first odometric information and eventually the laser readings to orient toward tunnel t_4 and when oriented this new tunnel is traversed by the follow behavior.

5 Experiments

5.1 Indoor Trial Runs

In the initial stage we wanted to test the applicability of the system described above when navigating a set of interconnected corridors at a higher speed. For this purpose we staged a number of indoor trial runs using the ATRV-Jr research robot shown in Figure 1 above. These runs were performed in a basement consisting of a number of approximately 2 m wide corridors with a number of junctions, doors, 45 and 90 degree turns as well as a few slightly larger open areas. We started by setting up a simple topological map and placed RFID tags first on the walls and later in the ceiling of the important junctions (see Figure 5a).

Next, a number of runs between a starting point and a target point were made. During the first runs no metrical or RFID information was provided in the topological map, i.e. navigation was solely based on the information extracted from the laser data. This worked well in most corridors and intersections, but in the corridor that included the small open areas the system mistook the open areas as intersections and got lost. After this, two more sets of test runs were made, one using a map with RFID information, the other with coarse metric information added to the map.

By adding RFID information to the topological map the system was able to distinguish between the real junctions and the open areas and reach its target position. However the mounting of the tags turned out to be crucial. All of the runs when the tags were placed in the ceiling were successful and the robot could navigate these corridors at a speed of up to 1.7 m/s. As for the runs with the tags mounted on the walls we experienced a few failures caused by undetected tags since this mounting was outside the specification of the tag/reader combination.

Equal results were achieved with the map with metric information to it, the 150 m long path was travelled as planned in under 180 s despite disturbances as non modelled open doors and recycle paper trailers parked in the corridors.

5.2 Trials from Test Mine

In order to test the techniques in the target environment we have used an abandoned mine, used by Atlas Copco in the testing of LHD vehicles. The mine consists of a



Fig. 5. RFID tags a) in basement, b) on the walls of mine and c) on a stand above center of junction.

number of 10 m wide and 5 m high tunnels with a fairly flat tunnel floor. In the setup of the tests we built a rough topological map consisting of five nodes which should be visited and in doing so a large number of junctions would be passed.³ Since no accurate metric information was available, we placed RFID tags in the junctions either by placing them on the walls of the tunnels or (for practical reasons) using a 3 m high stand simulating placement in the ceiling. See Figures 5b and 5c for a picture of the tags and their stands.

We used the same ATRV-Jr research robot and control program as above for these tests, only parameter changes to enable detection of the much wider tunnels were made. We used odometry to update localization and RFID tags to re-localize and verify that the correct junctions were reached.

By using the corridor localization technique described in Section 3.1 it was possible to localize the tunnels, which was needed to correct from the large odometry errors caused by the uneven surface. As for the localization of the junctions this was achieved through the RFID tags when the robot passed within a radius of 3 m from the center point under the corresponding tag. By placing the tags higher up (in the ceiling) this radius would be increased sufficiently to make it impossible to miss the readings.

6 Conclusions

In underground mining the development of fully automated systems for the navigation of loaders is interesting for a number of reasons, including safety and efficiency. By combining standard robotic techniques such as fuzzy behavior based systems with some application specific techniques the robustness and usability of fully automated navigation systems for autonomous underground vehicles can be improved. In this paper we have investigated a few problems with implementing such navigation systems and presented a solution based on a hybrid metric-topological map with a number of redundant methods for localization which provides greater robustness than any one solution alone. Though still a project under development, the initial tests of this system in realistic environments look promising.

³ From the main tunnels there are side tunnels with a spacing of only about 10 m.

Acknowledgements

This work is partly funded by the Swedish organisation *Robotdalen*, and partly by the Swedish KK Foundation.

References

1. P. Buschka and A. Saffiotti. Some notes on the use of hybrid maps for mobile robots. In *Proc of the 8th Int Conf on Intelligent Autonomous Systems (IAS)*, pages 547–556, Amsterdam, NL, 2004. Online at <http://www.aass.oru.se/~asaffio/>.
2. D. Driankov and A. Saffiotti, editors. *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*. Springer-Verlag, Berlin, Germany, 2001.
3. E. S. Duff and J. M. Roberts. Wall following with constrained active contours. In *4th International Conference on Field and Service Robotics*, July 14–16 2003.
4. E. S. Duff, J. M. Roberts, and P. I. Corke. Automation of an underground mining vehicle using reactive navigation and opportunistic localization. In *Australasian Conference on Robotics and Automation, Auckland*, pages 151 – 156, 27–29 November 2002. <http://www.araa.asn.au/acra/>.
5. G. Eriksson and A. Kitok. Automatic loading and dumping using vehicle guidance in a Swedish mine. In *International Symposium on Mine Mechanisation and Automation, Colorado*, pages 15.33 – 15.40, 1991.
6. J. Huser, H. Surmann, and L. Peters. Automatic behaviour adaption for mobile robots with different kinematics. In *Procs. of the European Congress on Fuzzy and Intelligent Technologies EUFIT*, pages 1095 – 1099, 1996.
7. J. Larsson and M-Broxvall. Fast laser-based feature recognition. In *Proc. of the 3rd Swedish Workshop on Autonomous Robotics*, pages 145–149, Stockholm, Sweden, 1–2 August 2005. Online at <http://www.aass.oru.se/Research/Robots/publications.html>.
8. H. Mäkelä. Overview of LHD navigation without artificial beacons. *Robotics and Autonomous Systems*, pages 21 – 35, 2001.
9. J. M. Roberts, E. S. Duff, P. I. Corke, P. Sikka, G. J. Winstanley, and Jo. Cunningham. Autonomous control of underground mining vehicles using reactive navigation. In *Proceedings of IEEE Int. Conf. on Robotics and Automation, San Francisco, USA*, pages 3790–3795, 2000.
10. T. M. Ruff. Ultrasonic guidance and remote control of a compact loader/tramper. In *International Symposium on Mine Mechanisation and Automation, Colorado*, pages 6.45–6.54, June 1991.
11. A. Saffiotti, K. Konolige, and E. H. Ruspini. A multivalued-logic approach to integrating planning and control. *Artificial Intelligence*, 76(1-2):481–526, 1995. Online at <http://www.aass.oru.se/~asaffio/>.
12. J. P. H. Steele, R. King, and W. Strickland. Modeling and sensor-based control of an autonomous mining machine. In *International Symposium on Mine Mechanisation and Automation, Colorado*, pages 6.55–6.67, June 1991.
13. U. Wiklund, U. Andersson, and K. Hyppä. AGV navigation by angle measurements. In *Proceedings of the 6th International Conference on Automated Guided Vehicle Systems*, pages 199–212, October 1988.

Outdoor Simultaneous Localisation and Mapping Using RatSLAM

David Prasser, Michael Milford, and Gordon Wyeth

School of Information Technology and Electrical Engineering
The University of Queensland
Australia
{prasserd, milford, wyeth}@itee.uq.edu.au

Summary. In this paper an existing method for indoor Simultaneous Localisation and Mapping (SLAM) is extended to operate in large outdoor environments using an omnidirectional camera as its principal external sensor. The method, RatSLAM, is based upon computational models of the area in the rat brain that maintains the rodent's idea of its position in the world. The system uses the visual appearance of different locations to build hybrid spatial-topological maps of places it has experienced that facilitate localisation and path planning. A large dataset was acquired from a dynamic campus environment and used to verify the system's ability to construct representations of the world and simultaneously use these representations to maintain localisation.

Keywords: SLAM, Omnidirectional Vision

1 Introduction

RatSLAM is a methodology for learning or mapping an environment while simultaneously maintaining localisation using visual information. The system was developed from computational models of the hippocampus of rodents - the part of the brain responsible for a creature's sense of place. RatSLAM has been successfully employed on mobile robots in indoor environments [6]. An obvious new application for the system is outdoor mapping and localisation. This paper describes how the RatSLAM system can be made to operate successfully on an outdoor mobile robot platform using a catadioptric camera as its principal external sensor. The most significant change to the system was to the visual processing component which was altered to take full advantage of the omnidirectional nature of the camera. The new visual processing method retains the nature of the original; it operates based purely on the visual appearance of the environment and does not compute any geometrical information. While this work builds in part upon early experiments reported in [8]

many aspects are new or greatly improved. In particular the visual processing has been expanded to include colour information and the test environment is more topologically complex.

2 Background

Most work in the area of Simultaneous Localisation and Mapping for field robots appears focused around the use of range measurement devices, particularly RADAR and laser range finders, which are processed by Kalman or particle filters to produce metric maps, for example [7, 2]. Vision based work with outdoor ground robots appears to be concerned more with localisation than concurrent mapping. Some vision based localisation techniques are of interest for moving the RatSLAM system outdoors as they are functionally similar to the appearance based method RatSLAM used indoors.

2.1 RatSLAM

Figure 1 shows the RatSLAM model. The robot's pose is represented within a competitive attractor network called the Pose Cells. These cells are arranged as a three dimensional array with two dimensions corresponding to x and y position, and the third dimension to orientation. The competitive attractor dynamics cause packets of activity that represent the robot's believed position to form in the Pose Cells. Wheel encoder information is used to perform path integration by shifting the activity packets. The external sensors are used to control the activity in a collection of cells called the Local View. The environment is learnt by associating active cells in the Local View with active Pose Cells. Relocalisation is accomplished by injecting energy into the Pose Cells based on these learnt associations and the current Local View activity. The new pose hypothesis then competes with the existing hypothesis via the Pose Cells' competitive attractor dynamics. The detailed implementation of the RatSLAM algorithm is discussed more thoroughly in [6].

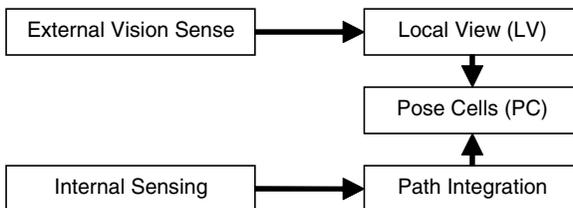


Fig. 1. The Pose Cells store the systems sense of position. This is continually modified by the Path Integration and Local View mechanisms which represent the internal and external sensors respectively.

2.2 Visual Place Recognition

An attractive method of visual information processing for building environment maps is appearance based place recognition. This type of processing is developed from the basic concept that if two camera images are very similar then it is likely that they were acquired at nearby places. If a representative set of images can be acquired at known locations throughout the environment then an estimate of the robot's current position can be determined by finding the best match between these and the current camera image. The problem then becomes one of image matching with a requirement for spatial invariance or generalisation. If the matching process is unable to generalise over a small area then the robot would only be able to localise at precisely the same position as when the reference image was required. Spatial generalisation however limits the physical accuracy to the size of the area in which the image is recognised. Appearance based localisation has been shown to be sufficient for robot navigation without using metric information, provided the environment is unique [3] and has been used in systems that mimic the rodent hippocampus [1]. Visual place recognition has also been used to construct topological maps of indoor environments [9].

Many techniques have been used to solve the image matching problem for this application, usually building an intermediate representation of the image that exhibits properties of spatial invariance and then employing a representation specific matching process. Some methods for image matching or image retrieval that have been used for robot localisation are: histogram matching [10]; image fingerprints [5]; and various types of invariant image features [4, 12]. Omnidirectional cameras are of particular interest since image representations can be constructed that are invariant to the orientation of the robot, removing the need to examine each place in the environment at multiple orientations.

Many visual appearance based localisation schemes use topological representations of space using mechanisms such as connected graphs of learnt views [3]. However RatSLAM operates in a semi-topological, semi-Cartesian manner. Each pose cell has an orderly physical relationship with other nearby cells, which, through path integration, creates a spatial representation which is roughly Cartesian across short distances. Over larger distances Cartesian relevance decreases to the point where the map must be considered topological.

3 Visual Learning

The relationship between visual information and pose is learnt by forming weighted connections between active pose and Local View cells. The connection strength β_{ilmn} between Local View cell i and Pose Cell lmn is increased according to the activities V_i , P_{lmn} of the respective cells and the learning

rate, λ (1). The maximum term limits the strength of a connection to that of the best observed correlation.

$$\beta_{ilmn}^{t+1} = \max(\beta_{ilmn}^t, \lambda V_i P_{lmn}) \quad (1)$$

Relocalisation can then occur by injecting into each Pose Cell the sum of the Local View cells multiplied by the appropriate learnt weights. In order to relocalise the same Local View cells should be active each time the robot is at the same position. The cells must also be sparsely activated to avoid the problem of linear inseparability that can occur within the single layer network between the Local View and Pose Cells. These two requirements can be met by making each Local View cell respond to a different viewpoint in the environment. In the current implementation each cell possesses a histogram representation of a particular view, with which the current camera information is compared. Histograms have several advantages for this application: they are invariant to rotation when used with omnidirectional cameras; and generalise with respect to camera position [10]. They are also compact to store and straightforward to compute. The map that RatSLAM learns is stored in two parts: a learnt set of views which reduce the external sensor data to sparse set of primitives; and the Local View - Pose Cell associations which indicate where these views may be found in the world.

3.1 Histogram Matching

Since the environment is unknown before the robot begins exploring it must simultaneously learn new areas and recognise previously visited viewpoints. This requires rapid online learning and recall where newly learnt information does not invalidate older information. Recognition is accomplished by matching the current hue and saturation histograms against a set of reference histogram pairs using the χ^2 statistic. The best match is reported to RatSLAM by setting its corresponding element in the Local View vector. When the match is weak then the unrecognised histograms are added to the reference set. In this way the system moves through the environment classifying the camera data into a growing set of distinct views. Each learnt histogram pair has a Local View cell to represent it to the rest of the system, so as more of the environment is explored the number of Local View cells increases.

The χ^2 distances are computed separately for the hue and saturation histograms and the results summed to give an overall measure of the distance between two pairs of histograms. Once the smallest χ^2 value is found it is compared to a sensitivity threshold d_m , to determine if a match has been found. The χ^2 statistic has a divide by zero condition when histogram bins are empty so a modified version is used, where a_i and b_i are the i th bins of the histograms a and b (2).

$$\chi^2 = \sum_i \begin{cases} \frac{(a_i - b_i)^2}{a_i + b_i}, & a_i + b_i \neq 0 \\ 0, & a_i + b_i = 0 \end{cases} \quad (2)$$

Each Local View cell then represents one or more physical parts of the environment that can be distinguished from other places by the external sensor. The physical area that a cell represents cannot be calculated or determined a priori, beyond a rough estimate of their average size. Some cells will code for multiple places in the environment that are visually similar. The converse is also true - one physical place may correspond to multiple Local View cells. This can occur when changes to part of the environment make that area better represented by some other Local View cell. If the environment later changes back then the original cell will again activate when the robot re-enters the area. It is responsibility of the rest of the SLAM process to reconcile the Local View information with internal sensors and maintain the robot's sense of position.

3.2 Histogram Formation

The processing of the camera input is limited to transformation from the camera's native YUV colour space to the Hue Saturation Value (HSV) space and the removal of some parts of the image. The individual histograms of hue and saturation are then calculated. It is these histograms that represent the visual scene to the learning process. The centre part of the image does not vary as the robot travels, nor do the regions that do not contain the mirror. These areas are not helpful for localisation and are removed in a masking operation before computing the histograms (Fig. 2). No accounting can be made for the two other constant features in the images: the support post for the mirror and the vehicle's operator, both of which can be seen in the lower part of Fig. 2. While these features create a blind spot that prevents the camera from being truly omnidirectional, there is no benefit in removing them from the image as it is not possible to recover the obscured information.

3.3 Orientation

The histogram matching system will operate independently of the robots orientation, however the mapping and localisation process must consider the orientation component as it is part of the robot's state. When each Local View cell is created it defines its own reference orientation by associating itself with the current Pose Cell activity. During relocalisation a measurement of the relative change in orientation since the recognised Local View cell was created can be used to appropriately shift its associated Pose Cells in the θ direction. Similarly when forming Local View - Pose Cell associations with existing Local View cells the orientation of the Pose Cells are shifted to maintain the Local View cell's reference orientation. The angular difference is added to the θ position of Pose Cells when relocalising and subtracted when learning.

Orientation is derived from the robot's on-board compass. When a Local View cell is first created the robot's compass orientation is recorded with the cell. Later when this cell is reactivated the difference between the current and

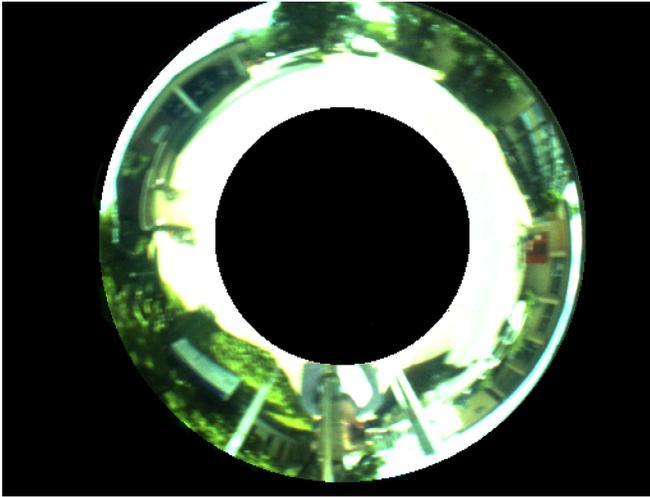


Fig. 2. The region of the image that is included in the histograms. Other parts of the image remain fairly constant as the robot moves, so they do not provide information useful for learning robot location.

stored orientations can be used to calculate the appropriate Pose Cell shift. Using relative measurements means that even if the compass fails to find true North around certain areas the system will still function. The compass is considered to be locally consistent rather than globally accurate.

4 Experimental Setup

Data was recorded from a mobile robot platform while travelling along repeated paths through a busy campus environment. The data was then presented to the RatSLAM system in a manner indistinguishable from on line operation.

4.1 Robot Platform

The robot platform is the CSIRO Autonomous Tractor [11]. While this vehicle is able to operate without human control, in these experiments the robot was driven manually since the environment is so complex and dangerous. The principal sensor used in these experiments is a catadioptric camera mounted on the front half of the vehicle. In these experiments the camera operated at approximately 4 Hz recording 1024×768 YUV422 images. The other sensing capabilities used in these experiments were the internal wheel odometry sensors and an electronic compass. The nature of the vehicle's drive system appears to make measurement of changes in orientation through path integration very inaccurate.

4.2 Environment

The test environment was the University of Queensland's St Lucia campus. The campus is a dynamic environment with sparse yet constant movement of people and vehicles during the experiment. The robot travelled through several different environment types including: footpaths; a board walk; a bridge; under a building; dense stands of trees; an open grass area; pedestrian precincts and roads. The day of data gathering could be described as extremely sunny. In shaded areas the environment appears very dark while in areas of direct sunlight recorded images are over bright, for instance black asphalt roads appear white. The data was recorded over a half hour period during which the robot travelled 2.8 km. Each point on the route shown in Fig. 3 was visited at least twice and every section of the route was traversed in both directions. There were four three-way intersections and three looped paths that began and ended at the same place.

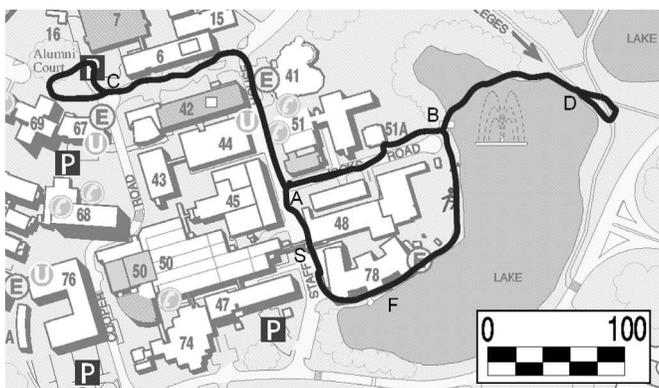


Fig. 3. The approximate path of the robot starting at *S* and finishing at *F*. The robot first travelled the inner loop *SABFS*, followed by the outer loop, *SACABDBFS* in a clockwise direction. Finally the outer loop was revisited in a counter-clockwise direction.

5 Results

The system performance can be examined by looking at the activity profiles over time of both the Local View and the Pose Cells. In both sets of activity, the performance measures are the consistency of the patterns of activity and how clearly this can be related back to the robot's position in the real world. Since the Local View affects the robot's sense of pose, and not the other way around, the Local View activity should be examined first.

5.1 Local View Activity

The Local View activity is driven by the histogram based view matching process. Each Local View cell corresponds to one view and is active when that view is recognised. The results of the histogram based view learning are shown in Fig. 4. The first time the robot moves through an area it will learn a new set of sequentially numbered views (for example the first 200 seconds of Fig. 4). Subsequent traversals of the same path should result in the same views being recognised in the same order, or the reverse order if the robot is travelling in the opposite direction. In Fig. 4 the first 150 or so Local View cells represent the central looped part of Fig. 3, 150–260 represent the loop near junction C and 260–350 the loop near D. The remaining 150 or so views were acquired when previously experienced areas were not recognised. The number of extra views learnt while revisiting already surveyed areas gives an indication of the systems competency and its ability to generalise.

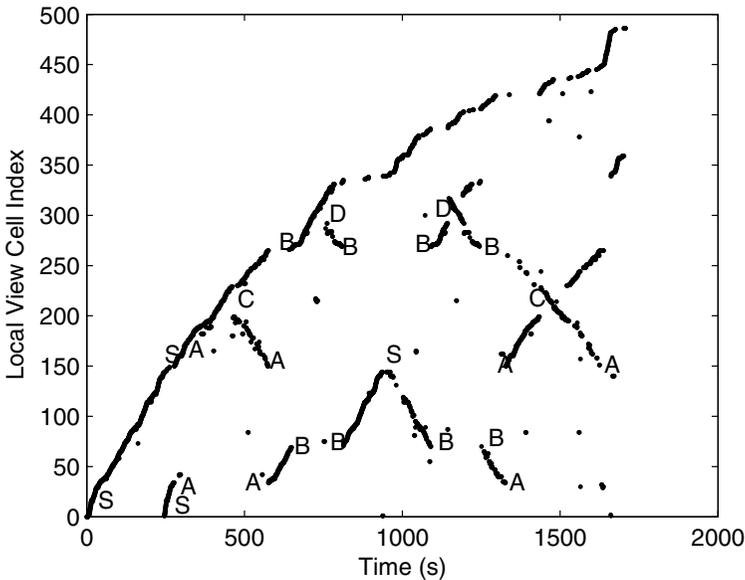


Fig. 4. The index numbers of the active Local View cells as the experiment progresses. The Local View cells that represent the marked places in Fig. 3 are indicated here with the same labels.

There are several reasons why the vision system may fail to produce the desired results. One obvious cause is a physical change to the environment, such as the arrival or departure of another vehicle, which makes it visually distinct from the learnt view of the environment. Another reason is a change not to the physical structure of the environment but in its illumination, which

could make the camera view appear sufficiently different to the original that it can no longer be recognised. In the present system the brightness component of the HSV colour space is deliberately ignored as it is expected to vary with changing illumination. This is not a perfect solution though and extreme changes in solar illumination will still prevent recognition. The final significant cause of failed recognition is insufficient spatial generalisation. If the camera image is not truly omnidirectional then the histograms acquired from one place at different orientations will be different, reducing the chance of recognition. The presence of the driver and the mirror support in the image cause this problem in the current system, although it is not a long term issue as learning one extra view is usually enough to achieve complete visual coverage of an area. In the short term it may make it more difficult to maintain localisation when first experiencing an environment from a novel orientation, since the alternative representation has not yet been learnt.

Some histogram pairs in Fig. 4 are recognised outside of the sequence in which they were originally learnt, appearing as noise in the Local View activity. These are not necessarily errors, rather they indicate some small part of the environment is visually similar in histogram space to some other distant place, a form of perceptual aliasing. RatSLAM associates such Local View cells with two sets of Pose Cells, so that when this view is recognised then it is taken as evidence that the robot could be in either of two locations. A few Local View cells will be activated inconsistently though and do constitute false positive type errors. RatSLAM requires that energy injected into the Pose Cells builds up as a competing packet over several time steps, so that perceptual aliasing and short term noise in the Local View cells do not cause incorrect relocalisation.

5.2 Pose Cell Trajectories

The logged wheel velocity and steering angle data can be integrated to produce the robot path shown in Fig. 5 - a representation that is clearly not consistent or usable. In contrast Fig. 6 shows the x, y location of the most strongly activated Pose Cell throughout the course of the experiment. This is not a Cartesian map but rather a plot over time of position in the system's own representation. Consistency between Fig. 6 and the robots position in the real world is the goal of the system. Repeated travels along a path in the real world translate to overlaid Pose Cell trajectories in Fig. 6.

The RatSLAM Pose Cell array wraps around in both the x and y dimensions so the trajectory moves continuously between the top and the bottom of the figure. The central loop of Fig. 3 becomes the Pose Cell trajectory of S, A, B, F, S', A' in Fig. 6 with the loop closed in both directions by jumps in Pose Cell activity from A' to A and from S to S' . When learning new areas RatSLAM is driven only by path integration so the layout of the Pose Cells used to represent the central loop is similar to the early part of Fig 5. Subsequent travel along this path activates the same Pose Cells as when the path

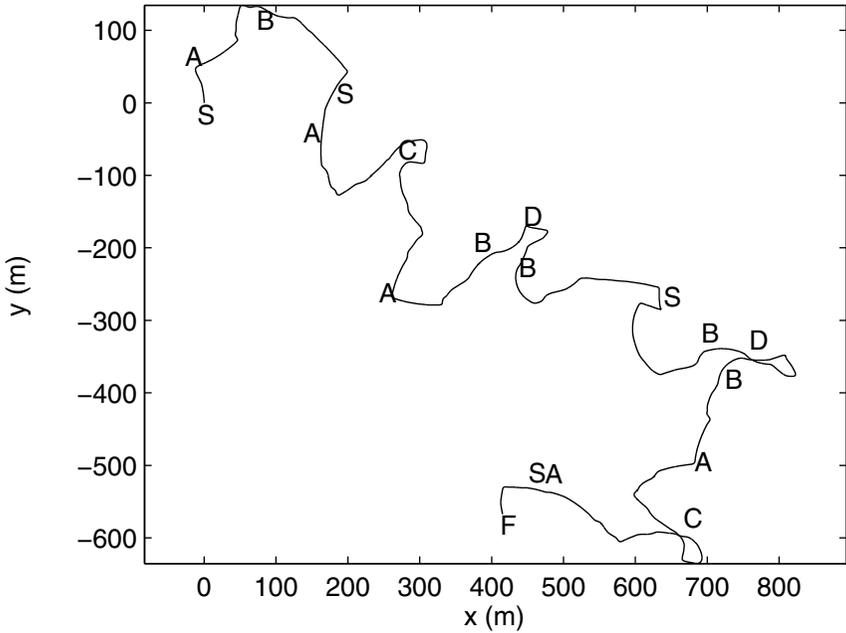


Fig. 5. Robot path computed from wheel velocity and steering information. An error of approximately 0.4° in the measurement of the steering angle is the most significant source of path integration error. Labelled points are the same as those in Fig. 3.

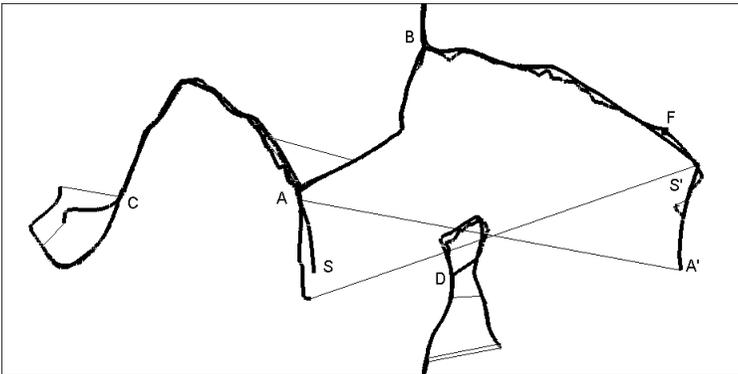


Fig. 6. Trajectory of the most strongly activated Pose Cell in x, y space. The Pose Cells wrap around in the x and y directions. Instantaneous jumps in cell position are shown by thin lines.

was first learnt. Certain areas such as the section of path between S' and B are not as cleanly recognised as other parts, for example $A - B$. This is caused by weaker visual place recognition in these areas. The lack of generalisation in this area is indicated by the formation of new Local View cells during the $S - B$ period in Fig. 4 at around 1000 seconds.

Since Fig. 6 shows only the most active Pose Cell, there appear to be sudden changes in the robots perceived position, for example the jump from A' to A . These jumps occur when a new activity packet becomes dominant after increasing in strength for several frames.

A similar effect can be seen in the small loop around the point marked C in Fig. 3. On the first visit the loop is closed by a jump back to the junction point, C . When the loop is revisited in the opposite direction a small spur is formed before jumping back to original representation of the loop.

The loop around point D results in a more complex situation. On the first pass, path integration fails to close the loop as is the usual case. There is a long delay before the dominant activity packet moves back to the main path. This results in separate forward and reverse paths being learnt for the area. During a second traversal of the loop the dominant activity packet changes between the two representations, jumping to the separate reverse path and then back to the main path. Unlike the other redundant parts of Fig. 6 which occurred during loop closing, this time one path is really being represented by two groups of Pose Cells which correspond to opposite directions of travel.

A pair of jumps also occurred around point A when visual ambiguity caused an incorrect activity packet to briefly become dominant.

The stability of the map with respect to significant global environmental changes, such as altered weather conditions, is untested. Stability could be achieved in a slowly changing environment where the system would be able to adapt to changes as they occur. However, the current method would eventually fail in a such an environment as the number of learnt views would increase indefinitely. Adding a mechanism for removing no longer relevant views from RatSLAM's memory would solve this problem.

6 Conclusion

The chief difficulty in converting the RatSLAM architecture to outdoor operation lay in reworking the vision system to use omnidirectional information. The new system learns to distinguish different camera locations by examining a histogram representation of their visual appearance. Despite the complex and dynamic nature of the environment this recognition process is reliable enough to function as the principal means of relocalisation. The only other structural change was the addition of the orientation learning subsystem described in Sect. 3.3. Some internal parameters also required adjustment so that the system would have an appropriate level of confidence in the new

Local View information. After these changes RatSLAM is able to build a representation of an outdoor environment under uncertainty of perception and motion. This representation is learnt while exploring and simultaneously used to maintain the systems believed position.

Acknowledgement. Thanks go to the CSIRO ICT Centre for the use of their autonomous tractor. In particular the work of Kane Usher, Ashley Tews, and Jon Roberts in conducting the data gathering stage is gratefully acknowledged.

References

1. A. Arleo, F. Smeraldi, S. Hug, and W. Gerstner. Place cells and spatial navigation based on vision, path integration, and reinforcement learning. *Advances in Neural Information Processing Systems*, 2001.
2. G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *Robotics and Automation, IEEE Transactions on*, 17(3):229–241, 2001.
3. M. O. Franz, B. Schölkopf, H. A. Mallot, and H. H. Bülthoff. Learning view graphs for robot navigation. *Autonomous Robots*, 5(1):111–125, 1998.
4. J. Koščeká and F. Li. Vision based topological markov localization. In *Proceedings International Conference on Robotics and Automation*, volume 2, pages 1481–1486, 2004.
5. P. Lamon, A. Tapus, E. Glauser, N. Tomatis, and R. Siegwart. Environmental modeling with fingerprint sequences for topological global localization. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 4, pages 3781–3786, 2003.
6. M. Milford, G. Wyeth, and D. Prasser. RatSLAM: a hippocampal model for simultaneous localization and mapping. In *Proceedings of the International Conference on Robotics and Automation*, volume 1, pages 403–408, 2004.
7. J. Nieto, J. Guivant, E. Nebot, and S. Thrun. Real time data association for FastSLAM. In *Proceedings International Conference on Robotics and Automation*, 2003.
8. D. Prasser, G. Wyeth, M. Milford, J. Roberts, and K. Usher. Experiments in outdoor operation of RatSLAM. In *Proceedings of the 2004 Australasian Conference on Robotics and Automation*. Canberra, 2004.
9. P.E. Rybski, F. Zacharias, and J.-F. Lett. Using visual features to build topological maps of indoor environments. In *Proceedings International Conference on Robotics and Automation*, volume 1, pages 850–855, 2003.
10. I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1023–1029, 2000.
11. K. Usher, P. Ridley, and P. Corke. Visual servoing of a car-like vehicle - an application of omnidirectional vision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 4288–4293, 2003.
12. J. Wolf, W. Burgard, and H. Burkhardt. Using an image retrieval system for vision-based mobile robot localization. In *Proc. of the International Conference on Image and Video Retrieval (CIVR)*. 2002.

Implementation Issues and Experimental Evaluation of D-SLAM

Zhan Wang, Shoudong Huang, and Gamini Dissanayake

ARC Centre of Excellence for Autonomous Systems (CAS), Faculty of Engineering, University of Technology, Sydney, Australia
{zwang,sdhuang,gdissa}@eng.uts.edu.au

Summary. D-SLAM algorithm first described in [1] allows SLAM to be decoupled into solving a non-linear static estimation problem for mapping and a three-dimensional estimation problem for localization. This paper presents a new version of the D-SLAM algorithm that uses an absolute map instead of a relative map as presented in [1]. One of the significant advantages of D-SLAM algorithm is its $O(N)$ computational cost where N is the total number of features (landmarks). The theoretical foundations of D-SLAM together with implementation issues including data association, state recovery, and computational complexity are addressed in detail. Evaluation of the D-SLAM algorithm is provided using both real experimental data and simulations.

Keywords: Decoupled SLAM, Extended Information Filter, Sparse Matrix, Computational Complexity

1 Introduction

Simultaneous localization and mapping (SLAM) is the process of building a feature based map of an environment while concurrently generating an estimate for the location of the robot. The SLAM problem has been the subject of extensive research in the past few years, most of which make use of estimation-theoretic techniques (see for example [2], [3], [4], [5], [6], [7] and the references therein).

In traditional SLAM, the state vector contains the location of the robot and all the feature locations. Some convergence properties of the traditional SLAM algorithm using Extended Kalman Filter are proved in [2]. However, traditional SLAM algorithms lead to a heavy computation burden for large scale problems. Many researchers have exploited the special structure of the SLAM algorithm in order to reduce the computational effort required in the SLAM process thereby make large scale SLAM more tractable. For example,

in [3], a compressed algorithm is presented to store and maintain all the information gathered in a local area, and then the information is transferred to the rest of the global map. In a recent publication [7], Thrun et al. used the Extended Information Filter to exploit the relative sparseness of the information matrix to reduce the computational effort required in SLAM.

Another way to reduce the computational complexity is to decouple the mapping and localization processes in SLAM. Different groups of researchers have been discussing the possibility of the decoupling. Most of them have made use of the idea of constructing a relative map using the observation information. For example, Newman [4] introduced a relative map in which the map state contains the relative locations among the features. Csorba et al. [8], Deans and Herbert [9], and Martinelli [10] have made use of relative map where the map state only contains distances among the features, which are invariants under shift and rotation. However, all the above approaches have redundant elements in the state vector of the relative map. If no further constraint is applied, it may result in inconsistent map. If constraints are applied, the computation complexity will be increased dramatically. Moreover, how to extract the information about the relative map from the observations and the possible information loss in the decoupling of localization and mapping have not been fully addressed.

In our recent research work [1], a novel decoupled SLAM algorithm, D-SLAM using compact relative maps, is proposed. The state vector for the mapping in D-SLAM is a $2n - 3$ dimensional vector containing distances and angles among the features (where n is the total number of features). It is shown that the new formulation retains the significant advantage of being able to improve the location estimates of all the features from one local observation. When Extended Information Filter is applied, D-SLAM results in a sparse information matrix.

This paper provides a D-SLAM algorithm where the state vector for mapping is the absolute locations of the features ($2n$ dimension for n features). The new algorithm is easier to implement than the D-SLAM algorithm using relative map, yet maintains the sparseness of the information matrix and the resulting computational savings. Some discussion on the implementation issues and further evaluation of D-SLAM using experimental data is presented in this paper. The paper is organized as follows. In Section 2, the key idea of D-SLAM and the details of the mapping and localization algorithms are provided. Section 3 addresses some implementation issues in D-SLAM including data association, state recovery and computational complexity. Experimental and simulation results are presented and compared with the results using traditional SLAM in Section 4. Section 5 concludes the paper and addresses future research directions.

2 D-SLAM Algorithm

In traditional SLAM, the state vector contains both the robot location (consisting of the position and orientation of the robot) and the feature locations. In the D-SLAM algorithm proposed below, the state vector for the mapping only contains the absolute locations of the features. The state vector for the localization only contains the robot location. The key step is to recast the measurement vector such that the information about the map contained in the measurements is relatively separated from the information about the robot location. In this section, we first briefly review the recasting, then discuss in detail the procedure of the mapping and localization process in D-SLAM using absolute map.

2.1 New Measurements Used in D-SLAM

We assume that the robot observes more than one feature at a time. Suppose robot observes m features f_1, \dots, f_m at a particular time. The original measurements (used in traditional SLAM) are the measured range and bearing of each observed feature:

$$z_{old} = [r_1, \theta_1, \dots, r_m, \theta_m]^T. \quad (1)$$

It contains Gaussian noise with zero mean and covariance matrix

$$R_{old} = \text{diag}[\sigma_{r_1}^2, \sigma_{\theta_1}^2, \dots, \sigma_{r_m}^2, \sigma_{\theta_m}^2]. \quad (2)$$

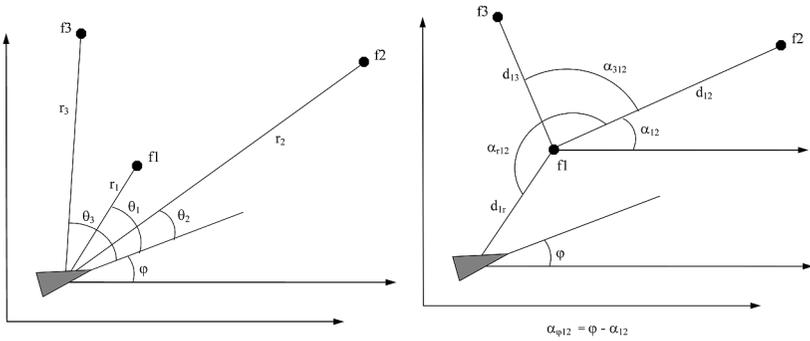
New measurement vector used in D-SLAM is

$$z_{new} = \begin{bmatrix} z_{rob} \\ z_{map} \end{bmatrix} = \begin{bmatrix} \alpha_{r12} \\ d_{1r} \\ \alpha_{\phi12} \\ \text{---} \\ d_{12} \\ \alpha_{312} \\ d_{13} \\ \vdots \\ \alpha_{m12} \\ d_{1m} \end{bmatrix} = \begin{bmatrix} \text{atan2}\left(\frac{-\tilde{y}_1}{-\tilde{x}_1}\right) - \text{atan2}\left(\frac{\tilde{y}_2 - \tilde{y}_1}{\tilde{x}_2 - \tilde{x}_1}\right) \\ \sqrt{(-\tilde{x}_1)^2 + (-\tilde{y}_1)^2} \\ -\text{atan2}\left(\frac{\tilde{y}_2 - \tilde{y}_1}{\tilde{x}_2 - \tilde{x}_1}\right) \\ \text{---} \\ \sqrt{(\tilde{x}_2 - \tilde{x}_1)^2 + (\tilde{y}_2 - \tilde{y}_1)^2} \\ \text{atan2}\left(\frac{\tilde{y}_3 - \tilde{y}_1}{\tilde{x}_3 - \tilde{x}_1}\right) - \text{atan2}\left(\frac{\tilde{y}_2 - \tilde{y}_1}{\tilde{x}_2 - \tilde{x}_1}\right) \\ \sqrt{(\tilde{x}_3 - \tilde{x}_1)^2 + (\tilde{y}_3 - \tilde{y}_1)^2} \\ \vdots \\ \text{atan2}\left(\frac{\tilde{y}_m - \tilde{y}_1}{\tilde{x}_m - \tilde{x}_1}\right) - \text{atan2}\left(\frac{\tilde{y}_2 - \tilde{y}_1}{\tilde{x}_2 - \tilde{x}_1}\right) \\ \sqrt{(\tilde{x}_m - \tilde{x}_1)^2 + (\tilde{y}_m - \tilde{y}_1)^2} \end{bmatrix} \quad (3)$$

where

$$\begin{pmatrix} \tilde{x}_i \\ \tilde{y}_i \end{pmatrix} = \begin{pmatrix} r_i \cos \theta_i \\ r_i \sin \theta_i \end{pmatrix}, \quad i = 1, \dots, m. \quad (4)$$

The physical meaning of the new measurement vector is shown in Figure 1(b) with that of the original measurements shown in Figure 1(a).



(a) Original measurements used in tra- (b) New measurements used in D-
ditional SLAM SLAM

Fig. 1. Measurements used in traditional SLAM and D-SLAM

The noise on z_{rob} and z_{map} are assumed to be Gaussian with zero mean; the covariance matrices R_{rob} and R_{map} can be obtained by (2), (3) and (4) using Jacobian of the functions evaluated at the measurement value r_i, θ_i . This kind of assumption and approximation using linearization have been used in all the Extended Kalman Filter (or Extended Information Filter) related literature.

In the new measurement vector z_{new} , z_{rob} depends on the robot pose and features f_1, f_2 while z_{map} contains information about distances and angles among features which are independent of the coordinate system, namely invariant under shift and rotation. The part z_{map} carries the maximal amount of information of the map that can be extracted from the observations. **In D-SLAM, the key idea is to use only z_{map} in the mapping.**

However, z_{rob} and z_{map} are not independent. Therefore, the estimation process need to be formulated carefully in order that statistically consistent estimates are obtained. In the next two subsections, details of the mapping and localization algorithms in D-SLAM with absolute map are provided.

2.2 Mapping in D-SLAM

State vector: The state vector in mapping contains the locations of the features:

$$X = (X_1, \dots, X_n)^T = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)^T. \quad (5)$$

For convenience, we choose the initial robot coordinate system as the coordinate system, where the origin is the initial robot position and the x -axis is along the initial robot heading.

Since all the features are assumed to be stationary, there is no prediction step and the mapping problem is a non-linear static estimation problem. Extended Information Filter (e.g. [11] [7]) is used to derive the formulas. The

relation between estimated state vector $\hat{X}(k)$ and information vector $i(k)$ is

$$i(k) = I(k)\hat{X}(k) \quad (6)$$

where $I(k)$ is the information matrix which is the inverse of the covariance matrix.

Phase I: The robot is stationary at its initial position

In this phase, the robot location is perfectly known. The original measurements (the range r_i and bearing θ_i) will be used to initialize and/or update feature f_i . The details are omitted.

Phase II: The robot is away from its initial position

Measurement model: Suppose the robot observes m features f_1, \dots, f_m and f_1, f_2 are old features. The model of the new measurement for mapping is

$$z_{map} = [d_{12}, \alpha_{312}, d_{13}, \dots, \alpha_{m12}, d_{1m}]^T = H_{map}(X) + w_{map} \quad (7)$$

where

$$H_{map}(X) = \begin{pmatrix} \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \\ \text{atan2}\left(\frac{y_3 - y_1}{x_3 - x_1}\right) - \text{atan2}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \\ \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2} \\ \dots \\ \text{atan2}\left(\frac{y_m - y_1}{x_m - x_1}\right) - \text{atan2}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \\ \sqrt{(x_m - x_1)^2 + (y_m - y_1)^2} \end{pmatrix} \quad (8)$$

and w_{map} is the new measurement noise whose covariance matrix R_{map} can be computed by (2), (3) and (4).

Initialize new features: Suppose the current estimation of the location of features f_1, f_2 are $\hat{X}_1 = (\hat{x}_1, \hat{y}_1)$ and $\hat{X}_2 = (\hat{x}_2, \hat{y}_2)$. They can be used together with d_{1i}, α_{i12} in z_{map} to initialize the location of new feature f_i as follows:

$$\begin{aligned} \alpha_{12} &= \text{atan2}\left(\frac{\hat{y}_2 - \hat{y}_1}{\hat{x}_2 - \hat{x}_1}\right) \\ \hat{x}_i &= \hat{x}_1 + d_{1i} \cos(\alpha_{12} + \alpha_{i12}) \\ \hat{y}_i &= \hat{y}_1 + d_{1i} \sin(\alpha_{12} + \alpha_{i12}). \end{aligned} \quad (9)$$

Update (old and new) features: When new features are observed, the dimension of the information vector and information matrix will be increased by adding zeros for the new features. We still denote the new information vector as $i(k)$, the new information matrix as $I(k)$, and the new state estimation as $\hat{X}(k)$.

The formulas for the update of the information vector and the information matrix using the measurement z_{map} are as follows:

$$\begin{aligned}
I(k+1) &= I(k) + \nabla H_{map}^T R_{map}^{-1} \nabla H_{map} \\
i(k+1) &= i(k) + \nabla H_{map}^T R_{map}^{-1} [z_{map}(k+1) - H_{map}(\hat{X}(k)) + \nabla H_{map} \hat{X}(k)]
\end{aligned}
\tag{10}$$

where ∇H_{map} is the Jacobian of the function H_{map} evaluated on the current state estimation $\hat{X}(k)$.

2.3 Localization in D-SLAM

State vector: The state vector used in localization is the three dimensional robot location:

$$X_r = (x_r, y_r, \phi_r)^T. \tag{11}$$

Localization is only needed when the robot is away from its initial position. We can obtain two estimates of the robot location. The first estimate is from the process model plus the priori knowledge of the robot location. The details are the same as those in the traditional SLAM and are omitted here. The second estimate is from the measurements.

Measurement model: Suppose the robot observes m features f_1, f_2, \dots, f_m , among which $f_1, \dots, f_{m_1}, m_1 \leq m$ are features that have been previously seen. Part of the original measurement vector z_{old} that involves these old features is used for localization

$$z_{loc} = H_{loc}(X_1, \dots, X_{m_1}, X_r) + w_{loc}. \tag{12}$$

Estimate from measurement: An estimate of X_r can be obtained by z_{loc} and the current estimates of f_1, \dots, f_{m_1} and their corresponding covariance matrix (a submatrix of the whole covariance matrix).

Combine two estimates using Covariance Intersection: Close examination of the estimation process reveals that the two estimates generated above are not independent. In some cases, for example in an indoor robot equipped with a laser sensor, the estimate from measurement itself may provide a sufficiently accurate robot location. In our simulation, we combine the two estimates using Covariance Intersection (CI) [6], which facilitates combining two correlated pieces of information when the extent of correlation itself is unknown.

As in the case of D-SLAM using compact relative map [1], although z_{map} in (7) and z_{loc} in (12) are not independent, the observation information is not reused. This is because the information about the robot location obtained from the localization process will never be used in the mapping process.

3 Implementation Issues

3.1 Data Association

Data association refers to the process of associating the observations to the corresponding features. As in the traditional SLAM, many data association

algorithms can be applied in the proposed D-SLAM algorithm. Generally speaking, batch data association algorithms (e.g [12]) are more robust than the standard maximum likelihood approach but the computational cost is higher.

In our simulation and experiment, we follow the standard maximum likelihood approach described in [2]. Due to erroneous feature detections caused by moving objects or measurement noise, two feature lists are maintained. One list stores features that are confirmed to be valid, and the other stores potential features yet to be validated. Mahalanobis distance between the newly observed features and the features in the two lists are computed in order to decide about the association.

Note that the recovery of feature location estimation and part of the associated covariance matrix is needed for the data association.

3.2 Recovery of the Feature Locations in D-SLAM

Recovery of the feature location estimation is not only needed in data association, but also needed in the map update and robot localization. When the number of features is small, the recovery can be simply obtained by (6) using the inverse of the information matrix. However, when the number of features is large, the computational cost of the inversion of the information matrix will be high. So it is crucial to find more efficient ways of the recovery.

We first consider which part of the map states is needed in the D-SLAM algorithm. (a) For mapping: as we can see in (10), by using the information vector, it is not necessary to compute the inverse of the information matrix $I(k+1)$ in the update step. However, the current state estimation of the features involved in the current observation is still needed to compute ∇H_{map} and $\nabla H_{map} \hat{X}(k)$. (b) For localization: in order to obtain the *Estimate from measurement*, the estimation of the old features f_1, f_2, \dots, f_{m_1} and their corresponding covariance matrix are needed. (c) For data association: only the estimates and the covariance matrices of features in the vicinity of robot (the vicinity here is defined in terms of the range of the sensor used for making observations) are needed.

In other words, we only need the estimation of the features within the sensor range of the current robot location and its corresponding covariance matrix. Since the Jacobian ∇H_{map} in (10) is sparse and there is no prediction step in the mapping process, the information matrix $I(k+1)$ is an exactly sparse matrix with the number of non-zero elements related to the sensor range. In fact, links (by link, we mean the non-zero off-diagonal element in the information matrix) between two features are established only if they are both involved in the same measurements at a particular time. The result is that links exist only between the features that are in the vicinity of each other. This exact sparseness makes it possible to reduce the computational cost of the map recovery significantly.

3.3 Computational Complexity

Let N be the number of features in the map. Two dimensional D-SLAM requires the storage of the information vector with dimension $2N$, the recovered state vector with dimension $2N$, the sparse information matrix with non-zero elements $O(N)$, and the submatrix of the covariance matrix corresponding to the currently observed features $O(1)$. The storage requirements are therefore of $O(N)$.

Updating the information matrix and the information vector requires the Jacobian ∇H_{map} as well as $\nabla H_{map} \hat{X}(k)$. Thus it is necessary to recover the current estimate of map state vector $\hat{X}(k)$. This can be done by solving a set of sparse linear equations, using few iterations requiring $O(N)$ operations as a good initial guess of $\hat{X}(k)$ is always available.

Once the Jacobian is computed, updating the information matrix and the information vector requires constant time as the Jacobian is always sparse and as a prediction step is not necessary.

For data association, locations as well as the uncertainty of the features in the vicinity of the robot are required. The vicinity here is defined in terms of the range of the sensor. This requires $O(N)$ operations to evaluate. The desired columns of the covariance matrix associated with these features can be obtained by solving a constant number of sparse linear equations with the aid of a good initial guess, which also requires $O(N)$ operations. Once the locations of the observed features and the corresponding covariance matrix are available, localization can be performed in constant time. Overall cost of D-SLAM is, therefore, $O(N)$.

4 Evaluation of D-SLAM

4.1 Experimental Evaluation with a Pioneer Robot in an Office Environment

The Pioneer 2 DX robot in our lab is used for the implementation. It is equipped with a laser range finder with a field of view of 180 degrees and an angular resolution of 0.5 degree to produce the relative range and bearing measurements between the robot and the features. We run the pioneer in our laboratory where we put twelve laser reflector strips in a $8 \times 8m^2$ area. The standard software, Player, is used to collect the control and sensor data from the robot. Then we run the D-SLAM algorithm in Matlab with the collected data.

In order to evaluate the robot and feature location estimation, we need the true value of the states. Here we use the traditional SLAM estimation as the truth. Figure 2(a) is the map obtained from D-SLAM. Figure 2(b) is the robot location estimation from D-SLAM with respect to traditional SLAM estimation. Figures 2(c) and 2(d) show the 2σ bound obtained from D-SLAM and traditional SLAM for the estimation of robot location and feature 9.

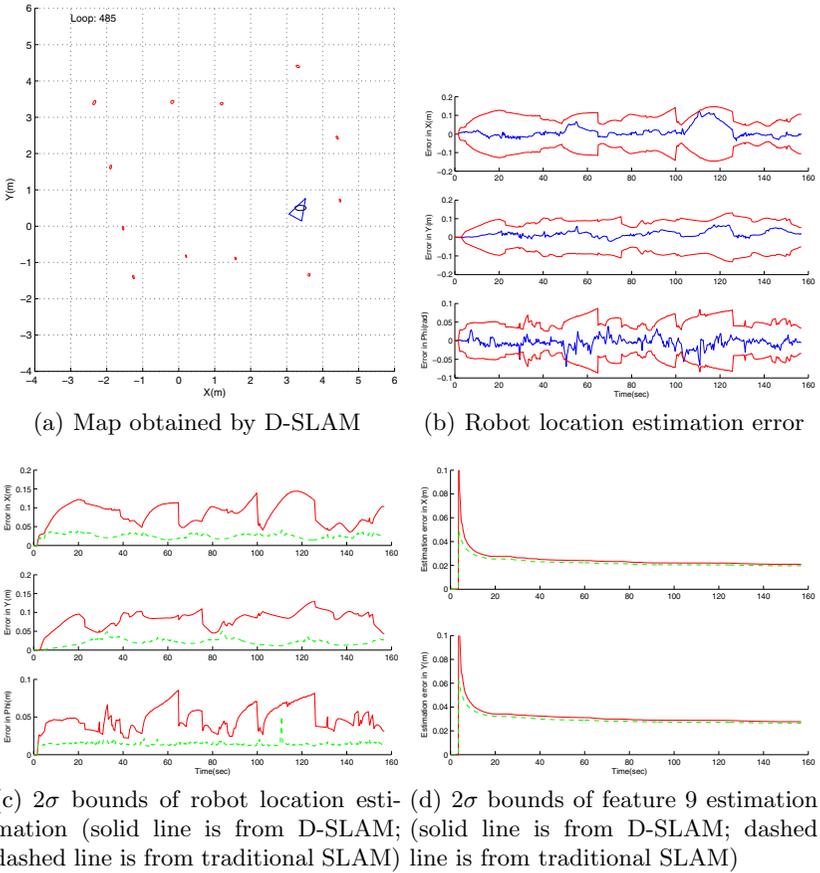


Fig. 2. D-SLAM implementation: map and estimation error

Figure 2(b) shows that the D-SLAM estimation is consistent. The map (Figure 2(a)) is almost as good as that of the traditional SLAM in this small area, as can be seen more clearly in feature 9 estimation in figure 2(d). In this figure, the 2σ bound from D-SLAM is very close to that from traditional SLAM. The slight difference comes from the fact that no information about robot location is fused into the map.

It can be seen from figures 2(b) and 2(c) that the localization result using CI is conservative. The reason is that CI applies conservative combination of the two estimates under the situation of not knowing their correlation [6]. The risk in it lies in the data association. The maximum likelihood method used in data association may fail when the robot estimation uncertainty is large. In D-SLAM, this failure may occur more frequently compared with traditional SLAM algorithms.

4.2 Evaluation of D-SLAM in Simulation with a Large Number of Features

In simulation, we ran D-SLAM algorithm in a much larger area, so as to further verify its convergence and illustrate its properties.

The environment used is a 40 meter square region. We put 196 features arranged in uniformly spaced rows and columns. The interval between two features is 3 meters. The robot starts from the left bottom corner and follows a random trajectory. Robot speed is 20cm/s and turn rate is 0.1rad/s. A sensor with a field of view of 180 degrees and a range of 5 meters is simulated to generate relative range and bearing measurements between the robot and the features.

Figure 3(a) and 3(b) show the maps obtained from D-SLAM and traditional SLAM. It can be seen that the uncertainty of the feature location estimates are more conservative in D-SLAM, compared with the traditional SLAM estimator. This information loss is expected.

Figure 3(c) shows the links among the features in the information matrix. This figure demonstrates more clearly that links only exist among features within sensor range. Figure 3(d) shows the non-zero elements in the information matrix obtained by the D-SLAM algorithm. Non-zero off-diagonal elements are caused by closing loops.

5 Conclusions

In this paper, we proposed a new SLAM algorithm: D-SLAM using absolute map. We addressed some key implementation issues and provided experimental verification for D-SLAM. The convergence of D-SLAM is verified by both real experimental data and simulations. Although the robot location is not incorporated in the state vector used in mapping, correlations among the features are still preserved in the information matrix. Therefore, the estimation uncertainty of the feature locations that are far away from the initial location of the robot is significantly reduced as the “loop is closed”. A significant advantage of D-SLAM is that the information matrix associated with the mapping is exactly sparse resulting in a significant reduction in computational complexity.

Besides the $O(N)$ computational cost, D-SLAM also has the following potential advantages: (1) since the mapping problem is treated as a static estimation problem, the multi-robot SLAM problem can be a simple extension, provided data association issues can be resolved; (2) some recent results have also shown that the large error in the robot orientation introduces inconsistency in traditional SLAM [14] [15]. D-SLAM does not have the robot location in the state vector used for mapping thus may be more robust than traditional SLAM.

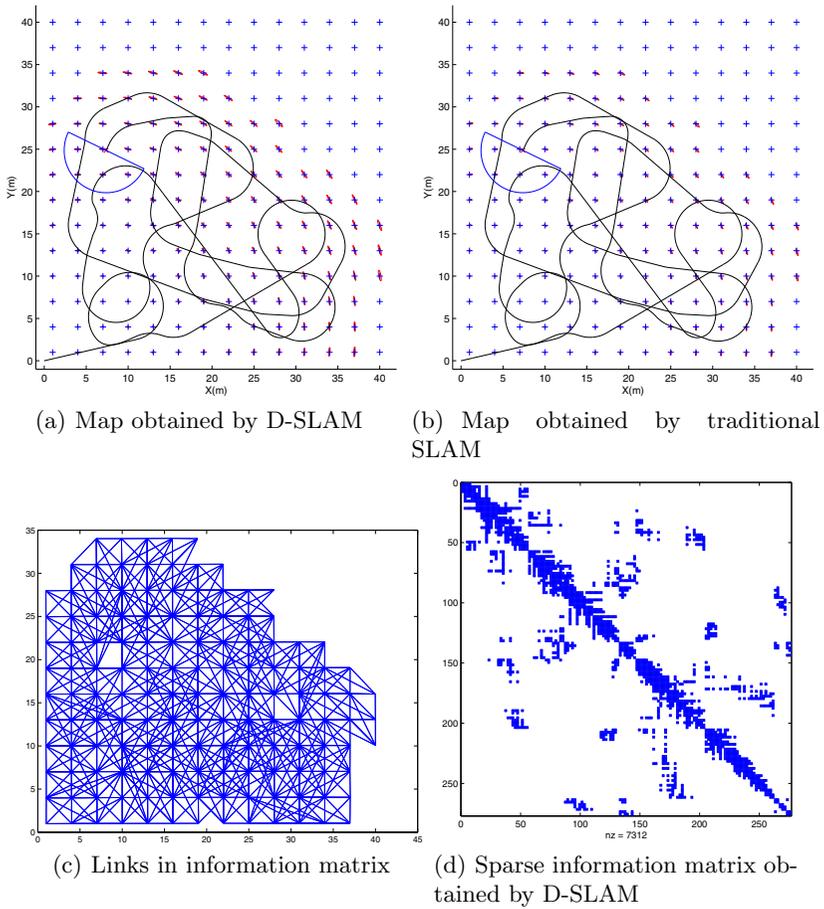


Fig. 3. D-SLAM simulations: Maps and Sparse Information Matrices

D-SLAM, however, results in some information loss because not all the information from the process model and observations is used for the mapping and localization. Preliminary analysis suggests that the extent of the information loss is related to the ratio between process noise and observation noise. It is seen from the experimental results that in many practical scenarios, with the availability of high rate scanners such as the SICK laser, the information loss is not a significant drawback.

Our ongoing research includes the detailed analysis of the information loss in D-SLAM, the verification using data from large outdoor environments, and multi-robot D-SLAM. Active D-SLAM problem where the robot trajectory is actively chosen on-line is our future research topic.

References

1. Wang Z, Huang S, Dissanayake G (2005) “Decoupling Localization and Mapping in SLAM Using Compact Relative Maps”, in *Proceedings of IROS 2005*, Edmonton, Canada
2. Dissanayake G, Newman P, Clark S, Durrant-Whyte H, and Csorba M (2001) “A solution to the simultaneous localization and map building (SLAM) problem”, *IEEE Trans. on Robotics and Automation* 17:229-241
3. Guivant JE, Nebot EM (2001) “Optimization of the simultaneous localization and map building (SLAM) algorithm for real time implementation”, *IEEE Trans. on Robotics and Automation* 17:242-257
4. Newman P (2000) “On the Structure and Solution of the Simultaneous Localization and Map Building Problem”, PhD thesis, Australian Centre of Field Robotics, University of Sydney, Sydney
5. Castellanos JA, Neira J, Tardos JD (2001) “Multisensor fusion for simultaneous localization and map building”, *IEEE Trans. on Robotics and Automation* 17:908-914
6. Julier SJ, Uhlmann JK (2001) “Simultaneous localization and map building using split covariance intersection”, in *Proceedings of IROS 2001*
7. Thrun S, Liu Y, Koller D, Ng AY, Ghahramani Z, Durrant-Whyte H (2004) “Simultaneous Localization and Mapping with Sparse Extended Information Filters”, *International J. of Robotics Research* 23:693-716
8. Csorba M, Uhlmann JK, Durrant-Whyte H (1997) “A suboptimal algorithm for automatic map building”, in *Proceedings of 1997 American Control Conference*. pp 537-541, USA
9. Deans MC, Hebert M (2000) “Invariant filtering for simultaneous localization and map building”, in *Proceedings IEEE International Conference on Robotics and Automation*. pp 1042-1047
10. Martinelli A, Tomatis N, Siegwart R (2004) “Open challenges in SLAM: An optimal solution based on shift and rotation invariants”, in *Proceedings IEEE International Conference on Robotics and Automation*. pp 1327-1332
11. Maybeck P (1979) “Stochastic Models, Estimation, and Control”, Academic, New York
12. Bailey T (2002) “Mobile Robot Localization and Mapping in Extensive Outdoor Environment”, PhD thesis, Australian Centre of Field Robotics, University of Sydney, Sydney
13. Pissanetzky S (1984) “Sparse Matrix Technology”. Academic, London
14. Frese U (2005), “A Discussion of Simultaneous Localization and Mapping”, *Autonomous Robots* (to appear). Available online <http://www.informatik.uni-bremen.de/~ufrese>
15. Castellanos JA, Neira J, Tardos JD (2004) “Limits to the consistency of EKF-based SLAM”, *5th IFAC Symp. on Intelligent Autonomous Vehicles, IAV'04*, Lisbon, Portugal

Scan-SLAM: Combining EKF-SLAM and Scan Correlation

Juan Nieto, Tim Bailey, and Eduardo Nebot

ARC Centre of Excellence for Autonomous Systems (CAS)
The University of Sydney, NSW, Australia
{j.nieto,tbailey,nebot}@acfr.usyd.edu.au

Summary. This paper presents a new generalisation of simultaneous localisation and mapping (SLAM). SLAM implementations based on *extended Kalman filter* (EKF) data fusion have traditionally relied on simple geometric models for defining landmarks. This limits EKF-SLAM to environments suited to such models and tends to discard much potentially useful data. The approach presented in this paper is a marriage of EKF-SLAM with scan correlation. Instead of geometric models, landmarks are defined by templates composed of raw sensed data, and scan correlation is shown to produce landmark observations compatible with the standard EKF-SLAM framework. The resulting *Scan-SLAM* combines the general applicability of scan correlation with the established advantages of an EKF implementation: recursive data fusion that produces a convergent map of landmarks and maintains an estimate of uncertainties and correlations. Experimental results are presented which validate the algorithm.

Keywords: Simultaneous localisation and mapping (SLAM), EKF-SLAM, scan correlation, Sum of Gaussians (SoG), observation model

1 Introduction

A mobile robot must know where it is within an environment in order to navigate autonomously and intelligently. Self-location and knowing the location of other objects requires the existence of a map, and this basic requirement has led to the development of the *simultaneous localisation and mapping* (SLAM) algorithm over the past two decades, where the robot builds a map piece-wise as it explores the environment. The predominant form of SLAM to date is *stochastic SLAM* as introduced by Smith, Self and Cheeseman [11]. Stochastic SLAM explicitly accounts for the errors that occur in sensed measurements: measurement errors introduce uncertainties in the location estimates of map landmarks which, in turn, incur uncertainty in the robot location estimate, and so the landmark and robot pose estimates are dependent. Practical implementations of stochastic SLAM represent these uncertainties and correlations

with a Gaussian probability density function (PDF), and propagate the uncertainties using an *extended Kalman filter* (EKF). This form of SLAM is known as EKF-SLAM [6]. One problem with EKF-SLAM is that requires the sensed data to be modelled as geometric shapes, which limits the approach to environments suited to such models.

This paper presents a new approach to SLAM which is based on the integration of scan correlation methods with the EKF-SLAM framework. The map is constructed as an on-line data fusion problem and maintains an estimate of uncertainties in the robot pose and landmark locations. There is no requirement to accumulate a scan history. Unlike previous EKF-SLAM implementations, landmarks are not represented by simplistic geometric models, but rather are defined by a template of raw sensor data. This way the feature models are not environment specific and good data is not thrown away. The result is *Scan-SLAM* that uses raw data to represent landmarks and scan matching to produce landmark observations. In essence, this approach presents a new way to define generic observation models, and in all other respects Scan-SLAM behaves in the manner of conventional EKF-SLAM.

The format of this paper is as follows. The next section presents a review of related work. Section 3 describes a *sum of Gaussians* (SoG) representation for scans of range-bearing data, based on the work presented in [1, Chapter 4]. Section 4 presents a method for obtaining a Gaussian likelihood function from the scan correlation procedure, which produces an observation in a form compatible with EKF-SLAM. Section 5 describes the generic observation model that is applied to all feature observations obtained by scan correlation. Section 6 presents *Scan-SLAM*, which uses the developments of Sections 4 and 5 to implement a scan correlation based observation update step within the EKF-SLAM framework. The method is validated with experimental results. Finally, conclusions are presented in Section 8.

2 Related Work

A significant issue with EKF-SLAM [4] is the design of the observation model. Current implementations require landmark observations to be modelled as geometric shapes, such as lines or circles. Measurements must fit into one of the available geometric categories in order to be classified as a feature, and non-conforming data is ignored. The chief problem with geometric observation models is that they tend to be environment specific, so that a model suited to one type of environment might not work well in another and, in any case, a lot of useful data is thrown away.

An alternative to geometric feature models is a procedure called *scan correlation*, which computes a maximum likelihood alignment between two sets of raw sensor data. Thus, given a set of observation data and a reference map composed similarly of unprocessed data points, a robot can locate itself without converting the measurements to any sort of geometric primitive. The

observations are simply aligned with the map data so as to maximise a correlation measure. Scan correlation has primarily been used as a localisation mechanism from an *a priori* map [13, 8, 3, 9], with the *iterated closest point* (ICP) algorithm [2, 10] and occupancy grid correlation [5] being the most popular correlation methods.

Two significant methods have been presented that perform scan correlation based SLAM. The first [12] uses *expectation maximisation* (EM) to maximise the correlation between scans, which results in a set of robot pose estimates that give an “optimal” alignment between all scans. The second method [7] accumulates a selected history of scans, and aligns them as a network. This approach is based on the algorithm presented in [10].

The main concern with these methods is that they do not perform data fusion, instead requiring a (selected) history of raw scans to be stored, and they are not compatible with the traditional EKF-SLAM formulation. This paper presents a new algorithm that combines EKF-SLAM with scan correlation methods.

3 Scan Matching Using Gaussian Sum Representation

A set of point measurements may be represented as a sum of Gaussians. This representation permits efficient correlation of two scans of data, and has a Bayesian justification which ensures that, under certain conditions, the scan alignment estimate is consistent (see [1]). SoG correlation also avoids limitations inherent to occupancy grid and ICP correlation methods; these being fixed-scale granularity and point-to-point data associations, respectively.

For a set of range-bearing measurements, such as a range-laser scan, the measurements and their uncertainties are first converted to sensor-centric Cartesian space. That is, a range-bearing measurement $\mathbf{z}_i = (r_i, \theta_i)$ with Gaussian uncertainty \mathbf{R}_i , is converted to Cartesian coordinates

$$\begin{aligned}\mathbf{x}_i &= \mathbf{f}(\mathbf{z}_i) = \begin{bmatrix} r_i \cos \theta_i \\ r_i \sin \theta_i \end{bmatrix} \\ \mathbf{P}_i &= \nabla \mathbf{f}_{\mathbf{z}_i} \mathbf{R}_i \nabla \mathbf{f}_{\mathbf{z}_i}^T\end{aligned}$$

where the Jacobian $\nabla \mathbf{f}_{\mathbf{z}_i} = \frac{\partial \mathbf{f}}{\partial \mathbf{z}_i}$.

We define an n -dimensional Gaussian as

$$g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P}) \triangleq \frac{1}{\sqrt{(2\pi)^n |\mathbf{P}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{p}})^T \mathbf{P}^{-1}(\mathbf{x} - \bar{\mathbf{p}})\right)$$

where $\bar{\mathbf{p}}$ and \mathbf{P} are the mean and covariance, respectively. An n -dimensional sum of Gaussians (SoG) is defined as the sum of k scaled Gaussians.

$$G(x) \triangleq \sum_{i=1}^k \alpha_i g(\mathbf{x}; \bar{\mathbf{p}}_i, \mathbf{P}_i)$$

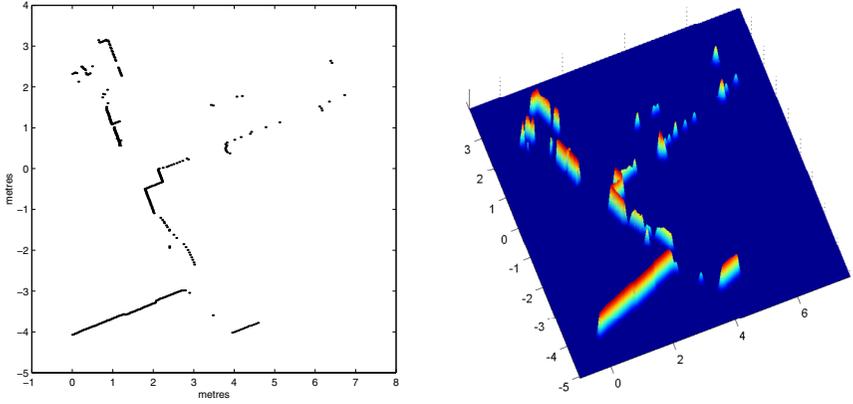


Fig. 1. The left-hand figure shows the set of raw range-laser data points transformed to a sensor-centric coordinate frame. The right-hand figure shows the SoG representation of this scan.

where, for a *normalised* SoG, the sum of the scaling factors α_i is one. However, normalisation is not necessary for correlation purposes—only relative scale is important—and it is more convenient to work with non-normalised SoGs. An example of a SoG produced from a range-laser scan is shown in Fig. 1.

Given two scans of Cartesian data points, where each point has a mean and variance, the respective scans may be represented by two SoGs.

$$G_1(\mathbf{x}) = \sum_{i=1}^{k_1} \alpha_i g(\mathbf{x}; \bar{\mathbf{p}}_i, \mathbf{P}_i)$$

$$G_2(\mathbf{x}) = \sum_{i=1}^{k_2} \beta_i g(\mathbf{x}; \bar{\mathbf{q}}_i, \mathbf{Q}_i)$$

A likelihood function for the correlation of these two SoGs is given by their cross-correlation.

$$\begin{aligned} \Lambda(\mathbf{x}) &= G_1(\mathbf{x}) \star G_2(\mathbf{x}) \\ &= \int \sum_{i=1}^{k_1} \alpha_i g(\mathbf{u} - \mathbf{x}; \bar{\mathbf{p}}_i, \mathbf{P}_i) \sum_{j=1}^{k_2} \beta_j g(\mathbf{u}; \bar{\mathbf{q}}_j, \mathbf{Q}_j) d\mathbf{u} \\ &= \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \alpha_i \beta_j \gamma_{ij}(\mathbf{x}) \end{aligned} \quad (1)$$

where $\gamma_{ij}(\mathbf{x})$ is the cross-correlation of two Gaussians $g(\mathbf{x}; \bar{\mathbf{p}}_i, \mathbf{P}_i)$ and $g(\mathbf{x}; \bar{\mathbf{q}}_j, \mathbf{Q}_j)$.

$$\gamma_{ij}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mu})^T \Sigma^{-1}(\mathbf{x} - \bar{\mu})\right)$$

$$\begin{aligned}\bar{\boldsymbol{\mu}} &= \bar{\mathbf{p}}_i - \bar{\mathbf{q}}_j \\ \boldsymbol{\Sigma} &= \mathbf{P}_i + \mathbf{Q}_j\end{aligned}$$

The result is a likelihood function that provides a measure of scan alignment, and a maximum-likelihood alignment can be obtained as

$$\mathbf{x}_M = \arg \max_{\mathbf{x}} \Lambda(\mathbf{x}) \quad (2)$$

where pose \mathbf{x}_M is the maximum-likelihood location of scan 1 with respect to scan 2. More precisely, \mathbf{x}_M is the location of the coordinate frame of scan 1 with respect to the coordinate frame of scan 2.

Full details of Gaussian sum correlation can be found in [1, Section 4.4]. In particular, it describes SoG scaling factors, SoG correlation in a plane (with alignment over position *and* orientation), and various alternatives for efficient implementation.

4 Scan Correlation Variance

For scan correlation to be compatible with EKF-SLAM, it is necessary to approximate the correlation likelihood function in (1) by a Gaussian. This section presents a method to compute a mean and variance for scan correlation based on the shape of the likelihood function in the vicinity of the point of maximum-likelihood. The resulting approximation is reasonable because the likelihood function tends to be Gaussian in shape in the region close to the maximum-likelihood.

The first step in deriving this approximation is to compute the variance of a Gaussian function given only a set of point evaluations of the function. Given a Gaussian PDF

$$g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{P}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{p}})^T \mathbf{P}^{-1}(\mathbf{x} - \bar{\mathbf{p}})\right)$$

the maximum-likelihood value is found at its mean

$$g(\bar{\mathbf{p}}; \bar{\mathbf{p}}, \mathbf{P}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{P}|}} = C_M$$

Any other sample \mathbf{x}_i from this distribution will have the value

$$\begin{aligned}g(\mathbf{x}_i; \bar{\mathbf{p}}, \mathbf{P}) &= C_M \exp\left(-\frac{1}{2}(\mathbf{x}_i - \bar{\mathbf{p}})^T \mathbf{P}^{-1}(\mathbf{x}_i - \bar{\mathbf{p}})\right) \\ &= C_i\end{aligned}$$

By taking logs and rearranging terms, we get

$$(\mathbf{x}_i - \bar{\mathbf{p}})^T \mathbf{P}^{-1}(\mathbf{x}_i - \bar{\mathbf{p}}) = -2(\ln C_i - \ln C_M) \quad (3)$$

Thus, given a set of samples $\{\mathbf{x}_i\}$ and their associated function evaluations $\{C_i\}$, along with the maximum-likelihood parameters $\bar{\mathbf{p}}$ and C_M , then the inverse covariance matrix \mathbf{P}^{-1} (and hence \mathbf{P}) can be evaluated. The only requirement is that the number of samples equals the number of unknown elements in \mathbf{P}^{-1} .

In this paper, we are concerned with 3-dimensional Gaussians, (i.e., to represent the distribution of a landmark pose $[x_L, y_L, \phi_L]^T$), and so we present the full derivation of variance evaluation for this case. We define the following variables

$$\begin{aligned} C'_i &= -2(\ln C_i - \ln C_M) \\ \mathbf{x}_i - \bar{\mathbf{p}} &= [x_i, y_i, z_i]^T \\ \mathbf{P}^{-1} &= \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix} \end{aligned}$$

Substituting these into (3) and expanding terms gives

$$x_i^2 a + 2x_i y_i b + 2x_i z_i c + y_i^2 d + 2y_i z_i e + z_i^2 f = C'_i \quad (4)$$

The result is an equation with six unknowns (a, \dots, f), and so a solution can be found given six samples from the Gaussian. This is posed as a matrix equation of the form

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

where the i -th row of \mathbf{A} is $[x_i^2, 2x_i y_i, 2x_i z_i, y_i^2, 2y_i z_i, z_i^2]$, \mathbf{x} is the unknowns $[a, b, c, d, e, f]^T$, and \mathbf{b} is the set of solutions $\{C'_i\}$. For a Gaussian function, the solution of this system of equations gives the exact covariance matrix of the function.

Since the scan correlation likelihood function is not exactly Gaussian (although we presume it has approximately Gaussian shape near the maximum likelihood location), different sets of samples will produce different values for \mathbf{P} . To reduce this variation, we evaluate more than the minimum number of samples and compute a least-squares solution using *singular value decomposition* (SVD), which results in a much more stable covariance estimate.

In summation, two SoGs are aligned according to a maximum likelihood correlation, to give the pose \mathbf{x}_M between scan coordinate frames. A number of samples, $N > 6$, from the region near \mathbf{x}_M are evaluated and the alignment variance \mathbf{P}_M is computed by SVD. At a higher level of abstraction, the result of this algorithm can be described by the following pseudocode function interface

$$[\mathbf{x}_M, \mathbf{P}_M] = \text{scan_align}(G_1(\mathbf{x}), G_2(\mathbf{x}), \mathbf{x}_0)$$

where \mathbf{x}_0 is an initial guess of the pose of $G_1(\mathbf{x})$ relative to $G_2(\mathbf{x})$. A good initial pose is required to promote reliable convergence.

5 Generic Observation Model

We define a landmark by a SoG in a local landmark coordinate frame, and the Scan-SLAM map stores a global pose estimate of this coordinate frame in its state vector (see Section 6 for details). Thus, all observations of landmarks obtained by scan matching can be modelled as the measurement of a global landmark frame \mathbf{x}_L as seen from the global vehicle pose \mathbf{x}_v (see Fig. 2).

The generic observation model for the pose of a landmark coordinate frame with respect to the vehicle is as follows.

$$\begin{aligned} \mathbf{z} &= [x_\delta, y_\delta, \phi_\delta]^T = \mathbf{h}(\mathbf{x}_L, \mathbf{x}_v) \\ &= \begin{bmatrix} (x_L - x_v) \cos \phi_v + (y_L - y_v) \sin \phi_v \\ -(x_L - x_v) \sin \phi_v + (y_L - y_v) \cos \phi_v \\ \phi_L - \phi_v \end{bmatrix} \end{aligned} \quad (5)$$

6 Scan-SLAM Update Step

When an object is observed for first time, a new landmark is created. A landmark definition template is created by extracting from the current scan the set of measurements that observe the object. These measurements form a SoG, which is transformed to a coordinate frame local to the landmark. While there is no inherent restriction as to *where* this local axis is defined, it is more intuitive to locate it somewhere close to the landmark data-points and, in this paper, we define the local coordinate frame as the centroid of the template SoG. A new landmark is added to the SLAM map by adding the global pose of its coordinate frame to the SLAM state vector. Note that the landmark description template is not added to the SLAM state and is stored in a separate data structure.

As new scans become available, the SLAM estimate of existing map landmarks can be updated by the following process. First, the location of a map

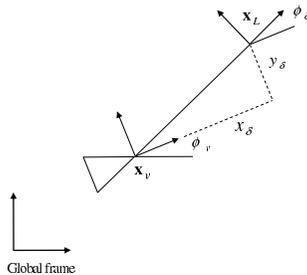


Fig. 2. All SoG features are represented in the SLAM map as a global pose identifying the location of the landmark coordinate frame. The generic observation model for these features is a measurement of the global landmark pose \mathbf{x}_L with respect to the global vehicle pose \mathbf{x}_v . The vehicle-relative observation is $\mathbf{z} = [x_\delta, y_\delta, \phi_\delta]^T$.

landmark relative to the vehicle is predicted to determine whether the landmark template SoG $G_L(\mathbf{x})$ is in the vicinity of the current scan SoG $G_o(\mathbf{x})$. This vehicle-relative landmark pose is the predicted observation $\hat{\mathbf{z}}$ according to (5). If the predicted location is sufficiently close to the current scan, the landmark template is aligned with the scan using the SVD correlation algorithm, using $\hat{\mathbf{z}}$ as an initial guess (see Fig. 3).

$$[\mathbf{z}, \mathbf{R}] = \text{scan_align}(G_L(\mathbf{x}), G_o(\mathbf{x}), \hat{\mathbf{z}})$$

The result of scan alignment gives the pose of the landmark template frame with respect to the current scan coordinate frame, which is defined by the current vehicle pose. This is the new landmark observation \mathbf{z} with uncertainty \mathbf{R} , in accordance with the generic observation model in (5). Having obtained the observation \mathbf{z} and \mathbf{R} , the SLAM state is updated in the usual manner of EKF-SLAM.

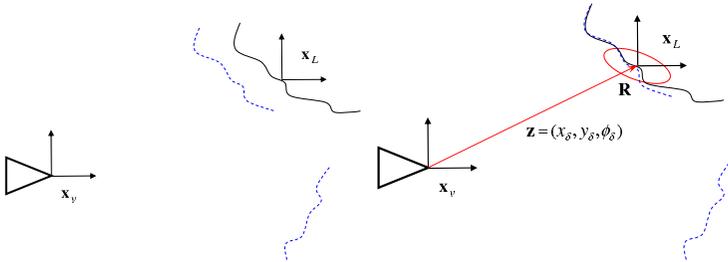


Fig. 3. The left-hand figure shows a stored scan landmark template (solid line) and a new observed scan (dashed line). The right-hand figure shows the scan alignment evaluated with the scan correlation algorithm from which the observation vector \mathbf{z} is obtained.

7 Results

This section presents simulation and experimental results of the algorithm presented. The importance of the simulation results is in the possibility to compare the actual objects position with the estimated by Scan-SLAM.

Fig. 4 shows the simulation environment. The experiment was done in a large area of 180 by 160 metres with a sensor field of view of 30 metres. The vehicle travels at a constant speed of $3m/s$. The sensor observations are corrupted with Gaussian noise with standard deviations 0.1 metres in range and 1.5 degrees in bearing. The simulation map consists of objects with different geometry and size. In order to select the segments to be added into

the navigation map, a basic segmentation algorithm was implemented that selects sensor segments that contain a minimum number of neighbour points.

The results for the Scan-SLAM algorithm are shown in Fig. 4. Here the solid line depicts the ground truth for the robot pose and the dashed line the estimated vehicle path. The actual object's position is represented by the light solid line and the segment's position by the dark points. The local axis pose for each scan landmark is also shown and the ellipses indicate the 3σ uncertainty bound of each scan landmark. The local axis position was defined equal to the average position of the raw points included in the segment and the orientation equal to the vehicle orientation. Fig. 5 shows the result after the vehicle closes the loop and the EKF-SLAM updates the map. The alignment between the actual object's position and that estimated by the algorithm after closing the loop can be observed.

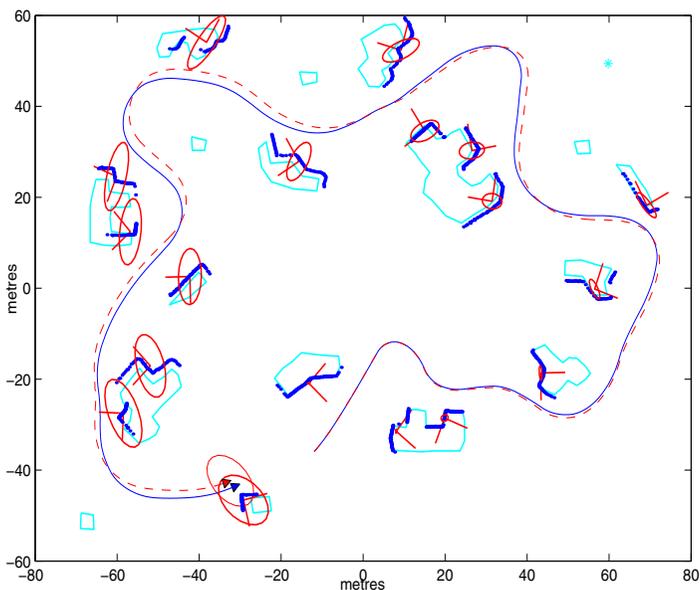


Fig. 4. The figure shows the simulation environment. The solid line depicts the ground truth for the robot pose and the dashed line the estimated vehicle path. The actual object positions are represented by the light solid line and the segment positions by the dark points. The ellipses indicate the 3σ uncertainty bound of each scan landmark.

The algorithm was also tested using experimental data. In the experiment a standard utility vehicle was fitted with dead reckoning and laser range sensors. The testing environment was the car park near the ACFR building. The environment is mainly dominated by buildings and trees. Fig. (6) illustrates the result obtained with the algorithm. The solid line denotes the trajec-

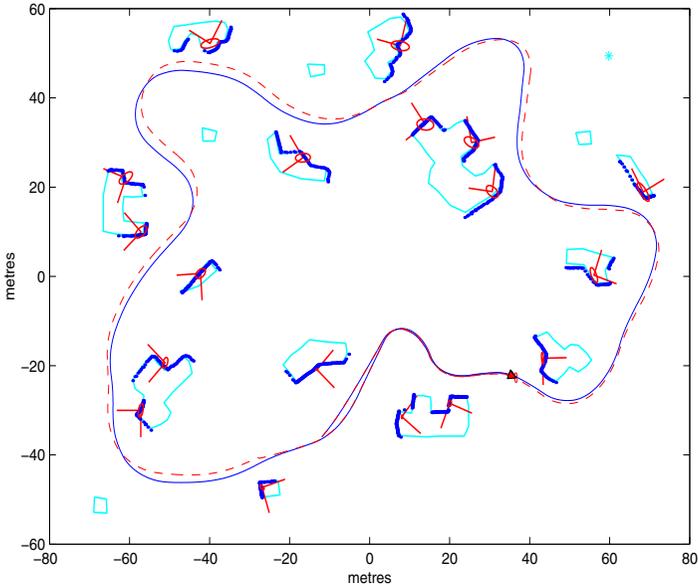


Fig. 5. Simulation result after closing the loop.

tory estimated. The light points is a laser-image obtained using feature-based SLAM and GPS, which can be used as a reference. The dark points represent the template scans and the ellipses the 1σ covariance bound. The local axis for the scan landmarks were also drawn in the figure. The segmentation criterion was also based on distance. Seven scan landmarks were incorporated and used for the SLAM.

8 Conclusions and Future Work

EKF-SLAM is currently the most popular filter used to solve stochastic SLAM. An important issue with EKF-SLAM is that it requires sensory information to be modelled as geometric shapes and the information that does not fit in any of the geometric models is usually rejected. On the other hand, scan correlation methods use raw data and are not restricted to geometric models. Scan correlation methods have mainly be used for localisation given an *a priori* map. Some algorithms that perform scan correlation based SLAM have appeared, but they do not perform data fusion and they require storage of a history of raw scans.

The Scan-SLAM algorithm presented in this paper combines scan correlation with EKF-SLAM. The hybrid approach uses the best of both paradigms; it incorporates raw data into the map representation and so does not require geometric models, and estimates the map in a recursive manner without the

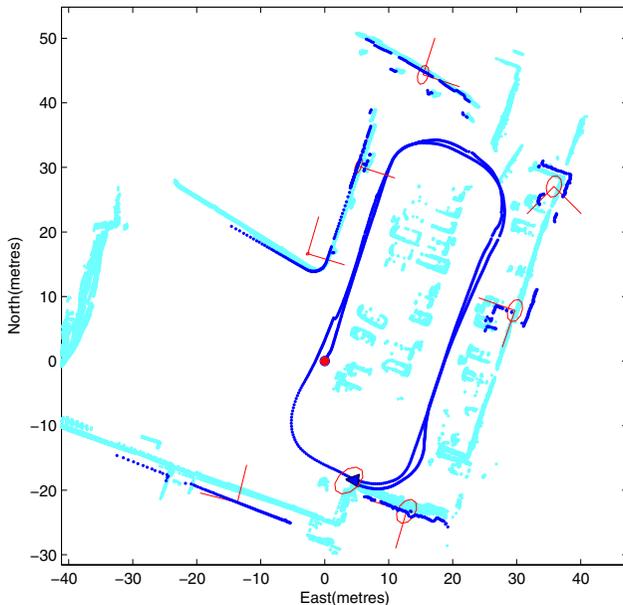


Fig. 6. Scan-SLAM result obtained in the car park area. The solid line denotes the trajectory estimated. The light points is a laser-image obtained using feature-based SLAM and GPS. The dark points represent the template scans and the ellipses the 1σ covariance bound.

need to store the scan history. It works as an EKF-SLAM that uses raw data as landmarks and utilises scan correlation algorithms to produce landmark observations. Finally experimental results were presented that showed the efficacy of the new algorithm.

In terms of future research, there is a lot of scope for developing the data association capabilities that arise from combining scan correlation with EKF-SLAM metric constraints. The ability of batch data association within an EKF framework to reject spurious data is well developed (e.g., [1, Chapter 3]). Scan correlation has the potential to strengthen the rejection of outliers by matching consecutive sequences of scans and removing points that are not reobserved. Also, scan correlation provides a measure of how well the *shape* of one scan fits another and can reject associations that have compatible metric constraints but misfitting shape.

A second area for future work is the development of continuously improving landmark templates. As a landmark is reobserved, perhaps from different view-points, the template model that describes it can be refined to better represent the object.

Acknowledgments

This work is supported by the ARC Centre of Excellence programme, funded by the Australia Research Council (ARC) and the New South Wales State Government.

References

1. T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, University of Sydney, Australian Centre for Field Robotics, 2002.
2. P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
3. W. Burgard, A. Derr, D. Fox, and A.B. Cremers. Integrating global position estimation and position tracking for mobile robots: The dynamic markov localization approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.
4. M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.
5. A. Elfes. Occupancy grids: A stochastic spatial representation for active robot perception. In *Sixth Conference on Uncertainty in AI*, 1990.
6. J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001.
7. J.S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 318–325, 1999.
8. J.S. Gutmann and C. Schlegel. Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *1st Euromicro Workshop on Advanced Mobile Robots (Eurobot'96)*, pages 61–67, 1996.
9. K. Konolige and K. Chou. Markov localization using correlation. In *International Joint Conference on Artificial Intelligence*, pages 1154–1159, 1999.
10. F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275, 1997.
11. R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Fourth International Symposium of Robotics Research*, pages 467–474, 1987.
12. S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *International Conference on Robotics and Automation*, pages 321–328, 2000.
13. G. Weiß, C. Wetzler, and E. Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *International Conference on Intelligent Robots and Systems*, pages 595–601, 1994.

A Non-rigid Approach to Scan Alignment and Change Detection Using Range Sensor Data

Ralf Kaestner¹, Sebastian Thrun¹, Michael Montemerlo¹, Matt Whalley²

¹ Robotics Laboratory
Computer Science Department
Stanford University
Stanford, CA

`ralfk@cs.stanford.edu`, `thrun@stanford.edu`, `mmde@stanford.edu`

² Army/NASA Rotorcraft Division
Aeroflightdynamics Directorate (AMRDEC)
US Army Research, Development and Engineering Command,
Ames Research Center, CA
`mwhalley@mail.arc.nasa.gov`

Summary. We present a probabilistic technique for alignment and subsequent change detection using range sensor data. The alignment method is derived from a novel, non-rigid approach to register point clouds induced by pose-related range observations that are particularly erroneous. It allows for high scan estimation errors to be compensated distinctly, whilst considering temporally successive measurements to be correlated. Based on the alignment, changes between data sets are detected using a probabilistic approach that is capable of differentiating between likely and unlikely changes. When applied to observations containing even small differences, it reliably identifies intentionally introduced modifications.

1 Introduction

We provide a unified probabilistic technique for alignment and subsequent change detection using range sensor data. Our work has been motivated by the goal to identify even small changes of the size of a little box by (airborne) vehicle observations. To allow for an exhaustive application of our approach, a terrain shall be sensed as infrequently as possible.

An autonomous Yamaha RMAX helicopter (see Fig. 1) that has been developed within the scope of the NASA Autonomous Rotorcraft Project (ARP), serves as model and experimental platform for our alignment and change detection approach. Tests were performed at the Ames Disaster Assistance and Rescue Team (DART) Collapsed Structure Rescue Training Site (see Fig. 2) located at NASA Ames Research Center in Mountain View, California.



Fig. 1. The NASA/ARP helicopter equipped with a range measurement device from SICK.



Fig. 2. The DART Collapsed Structure Rescue Training Site located in Mountain View, CA.

The vehicle’s pose is estimated by a Kalman filter integrating GPS position and velocity. The RMAX has been modified to include an avionics payload which, for our purpose, contains a SICK PLS laser range scanning unit that is mounted under the helicopter’s nose, pointing straight down at an angle of 90 degrees. The device provides centimeter-accuracy range measurement every one degree over a field-of-view of 180 degrees and at a frequency of 75 Hz. Details on ARP as well as the helicopter’s operational issues are to be found in [14].

2 Related Work

In the past, automatic change detection based on probabilistic techniques has widely been studied in the field of computer vision and image processing. In [1] the history of change detection from remotely sensed digital images is summarized comprehensively. A variety of approaches addressing interesting purposes, like environmental monitoring [3], urban [4], [5], [6] and forest surveillance [7], etc. have further been developed. These studies exclusively base on optical and range imagery acquired by camera or aerial mapping radar and lidar systems.

Recent major breakthroughs in the field of high-precision range sensor technologies led to an increasing availability of inexpensive scanning devices. As a consequence thereof, applying change detection methods to spatially interpreted range sensor data as presented in [8] and [9] has become more and more attractive. However, compared to the major efforts and breakthroughs in the imagery-related theories, these approaches have only been studied to some unsatisfactory extent. Nevertheless, they promise to open up a wide area of potentially superior applications.

In this paper, we are therefore deriving a novel unsupervised technique to evaluate changes in spatial point clouds. Our method deeply relates to those based on the so-called “difference image” as proposed in [1]. In principle, these methods analyze spatial or spatio-temporal distributions of a distance

metric between previously aligned reference and sample imagery. But in spite of their relative simplicity and widespread use, they usually exhibit a major weakness: As shown in [2], change detection accuracy strongly correlates with precise registration results that again suffer from the prevailing neglect of building sufficient models.

In contrast to related work that merely makes insufficient effort to precisely register data sets by only few rigid transformations, the theory presented in this paper focuses on enhancing alignment results significantly. This goal is achieved by applying a novel algorithm based on the non-rigid registration approach as presented in [13]. We therefore explicitly derive and consider models of the measuring processes involved in data acquisition.

3 Models

3.1 Helicopter Model

Whenever a range scan is acquired, the helicopter is at a specific pose relative to a global GPS coordinate system. Let us denote the pose by two sets of variables, pertaining to the x - y - z coordinates of the vehicle and its three Euler angles. Specifically, we assume that at time t , for $t = 1, \dots, N$, the helicopter's location is given by the variables x_t , y_t , and z_t . We denote the angles at time t by ϕ_t , θ_t and ψ_t . Thus, the state vector of the helicopter at time t is given by $\xi_{\mathbf{t}} = (x_t \ y_t \ z_t \ \phi_t \ \theta_t \ \psi_t)^T$.

An estimate of the helicopter's pose is provided by the onboard Kalman filter. Specifically, at time t we receive from the EKF a Gaussian pose estimate. The mean of an estimate will be denoted $\mu_{\mathbf{t}}$ and the covariance Σ_t . The sequence of all poses acquired during a single flight of the helicopter is represented by a high-dimensional Gaussian trajectory estimate ξ with mean μ and covariance Σ . Stated differently, we have $\xi \sim \mathcal{N}(\mu, \Sigma)$. This implies that the corresponding negative logarithm of the pose trajectory distribution is given by

$$-\log pr(\xi) = \text{const.} + \frac{1}{2} (\xi - \mu)^T \Sigma^{-1} (\xi - \mu) \quad (1)$$

The constant term in Eq. 1 is the logarithm of the normalizer $|2\pi\Sigma|^{-\frac{1}{2}}$. However, this normalizer does not depend on the target variable ξ . Henceforth, its actual value will play no important role, and it can safely be omitted.

3.2 Measurement Model

The helicopter acquires ground data using a range scanner. Each scan consists of $M = 180$ range measurements. The i -th measurement is oriented at angle

α_i . Let the actual range measurement be denoted $r_{t,i}$, where t is once again the time index, and i is the index of the measurement beam acquiring the range scan at time t .

Basic geometry suggests that the projection of this measurement into spatial coordinates is now obtained by

$$\mathbf{p}_{\mathbf{t},i}(\boldsymbol{\xi}_{\mathbf{t}}, r_{t,i}) = R_{\phi_t \theta_t \psi_t} R_i (0 \ 0 \ r_{t,i})^T + (x_t \ y_t \ z_t)^T \quad (2)$$

Here $R_{\phi_t \theta_t \psi_t}$ is the joint rotation matrix that maps the orientation of the helicopter's local coordinate system back into global world coordinates. The vector (x_t, y_t, z_t) represents the helicopter's location in 3-D world coordinates. Finally, the matrix R_i is a rotation matrix that captures the angle α_i of the i -th measurement beam.

In practice, even in static environments each measurement will be corrupted by noise. To model the noise, we assume the existence of a "true" surface point, denoted $\hat{\mathbf{p}}_{\mathbf{t},i}$. Of course, $\hat{\mathbf{p}}_{\mathbf{t},i}$ is unknown. However, this surface point induces a true range, denoted $\hat{r}_{t,i}$, which is unknown as well, but it will play an important role in determining whether two scans refer to the same static surface patch in the differencing process. Again, we assume the noise probability distribution elongating along the measurement beam to be Gaussian with mean $\hat{r}_{t,i}$ and variance s . Thus, we define $r_{t,i} \sim \mathcal{N}(\hat{r}_{t,i}, s)$. The corresponding negative logarithm shall then be given by

$$-\log pr(r_{t,i}) = \text{const.} + \frac{1}{2} (r_{t,i} - \hat{r}_{t,i}) s^{-1} (r_{t,i} - \hat{r}_{t,i}) \quad (3)$$

Once again, the constant term in Eq. 3 is the logarithm of the normalizer $|2\pi s|^{-\frac{1}{2}}$ and can be omitted. We remark that the assumed measurement error s plays a critical role in change detection. It characterizes the normal variation we are expecting when measuring the ground surface.

From Eq. 2 we learn that the distribution over probable surface points $\mathbf{p}_{\mathbf{t},i}$ depends on the time-related pose estimation $\boldsymbol{\xi}_{\mathbf{t}}$ and the measurement model for $r_{t,i}$. Both models are represented by Gaussians and we therefore infer that the $\mathbf{p}_{\mathbf{t},i}$ are also distributed normally. Hence, we define the joint probability $\mathbf{p}_{\mathbf{t},i} \sim \mathcal{N}(\hat{\mathbf{p}}_{\mathbf{t},i}, Q_{t,i})$ and the negative logarithm

$$-\log pr(\mathbf{p}_{\mathbf{t},i}) = \text{const.} + \frac{1}{2} (\mathbf{p}_{\mathbf{t},i} - \hat{\mathbf{p}}_{\mathbf{t},i})^T Q_{t,i}^{-1} (\mathbf{p}_{\mathbf{t},i} - \hat{\mathbf{p}}_{\mathbf{t},i}) \quad (4)$$

Here, the mean $\hat{\mathbf{p}}_{\mathbf{t},i}$ is once again the "true" surface point we have sensed, and the covariance $Q_{t,i}$ incorporates the helicopter model as stated in Sec. 3.1 along with the projection of our measurement model from Sec. 3.2 into spatial coordinates. The constant part is omitted as usual.

3.3 Practical Considerations

In change detection, the same ground is purposely sensed more than once. To distinguish variable values arising from two independent scanning runs, we introduce a special notation remarking that estimations occurred at different times. Therefore, poses, measurements and induced surface points originating from an earlier run will further be referred to as $\xi_{t'}$, $r_{t',i'}$ and $\mathbf{p}_{t',i'}$ respectively. Observations acquired during a subsequent pass will remain denoted ξ_t , $r_{t,i}$ and $\mathbf{p}_{t,i}$. The noise models will be modified analogously.

4 Scan Alignment

In this section, we are deriving a probabilistic model for the alignment process. Based on that, we will show why classical approaches addressing the problem of registering range scan data that originates from measurements carried out by a helicopter must perform poorly. Hence, we will introduce a novel and superior method that is much more capable of aligning maps of the discussed type.

4.1 Alignment Model

Based on the assumptions made in Sec. 3, we can now formally consider alignment as the problem of maximizing the posterior over locations of points $\mathbf{p} = \{\mathbf{p}_{t,i}\}$ in the world, although they might have been induced by different sensor readings acquired in different runs. Put differently, we seek to find

$$\operatorname{argmax}_{\xi,r} pr(\mathbf{p}) \tag{5}$$

where ξ denotes the helicopter’s trajectory estimate, whilst $r = \{r_{t,i}\}$ refers to all the measurements acquired during the entire duration of the observation flight.

As common, our approach minimizes the negative likelihood. From Eq. 5, we can therefore derive under independent sensor noise:

$$\operatorname{argmax}_{\xi,r} pr(\mathbf{p}) = \operatorname{argmin}_{\xi,r} -\log pr(\mathbf{p}) = \operatorname{argmin}_{\xi,r} -\log \prod_{t,i} pr(\mathbf{p}_{t,i}) \tag{6}$$

This expression may then be converted into a simple non-linear least squares problem by applying the negative logarithm stated in Eq. 4. The optimization then resolves to

$$\operatorname{argmin}_{\xi, r} \left\{ \sum_{t, i} \frac{1}{2} (\mathbf{p}_{t, i} - \hat{\mathbf{p}}_{t, i})^T Q_{t, i}^{-1} (\mathbf{p}_{t, i} - \hat{\mathbf{p}}_{t, i}) + \text{const.} \right\} \quad (7)$$

A basic assumption our approach does not make is the existence of an explicit model of the environment. This is an important factor, because we are aiming at being able to detect changes by sensing parts of the world not more than twice. Hence, it seems nearly impossible to incorporate enough knowledge to actually learn an adequate probabilistic representation for it. We therefore have to refer to measurements originating from a previous observation run as the reference. Put differently, we define

$$\hat{\mathbf{p}}_{t, i} = m(\mathbf{p}_{t, i}) \quad (8)$$

Here, the function $m(\mathbf{p}_{t, i}) \rightarrow \mathbf{p}_{t', i'}$ denotes a matching of points acquired at time t and angle α_i with corresponding surface points of the reference, that is points induced by sensor readings of an earlier pass. Hence, the posterior's optimization from Eq. 7 can be restated

$$\operatorname{argmin}_{\xi, r} \sum_{t, i} [\mathbf{p}_{t, i} - m(\mathbf{p}_{t, i})]^T Q_{t, i}^{-1} [\mathbf{p}_{t, i} - m(\mathbf{p}_{t, i})] \quad (9)$$

where constant parts and factors that apply to the whole term have been omitted. This is safe, because they do not depend on the target variables and thus, they will play no role in the overall optimization.

One might have noticed that we have not yet derived the joint covariance matrix $Q_{t, i}$ that appears in our distribution over probable surface points. From Eq. 2 we learn that inferring the elements of $Q_{t, i}$ involves several rotational and translational transformations to be applied to the distributions introduced by our helicopter and our measurement model. However, for the purpose of simplicity, the correct form of $Q_{t, i}$ will play no role in the overall alignment. Its determination is therefore dispensable.

4.2 Aligning Rigid Models

Iterative Closest Point Algorithm

Besl and McKay suggest in [10] to solve the alignment problem by using a special iteration scheme. They prove that, with respect to a given metric, an optimal matching between a reference and a sample point cloud can be achieved by repeatedly optimizing the registration parameters $R_{\Delta\phi\Delta\theta\Delta\psi}$ and $\Delta\mathbf{t}$ such that they minimize the distance metric between the sample and the best matching parts of the reference at a time. Here, $R_{\Delta\phi\Delta\theta\Delta\psi}$, or written briefly ΔR , denotes a joint transformation matrix that rotates all sample

points about the three spatial axes, whilst $\Delta \mathbf{t} = (\Delta x \ \Delta y \ \Delta z)^T$ performs a translation. Put formally, ICP therefore seeks to find

$$\operatorname{argmin}_{\Delta R, \Delta \mathbf{t}} \sum_{t,i} [\Delta R \mathbf{p}_{t,i} + \Delta \mathbf{t} - m(\mathbf{p}_{t,i})]^T Q_{t,i}^{-1} [\Delta R \mathbf{p}_{t,i} + \Delta \mathbf{t} - m(\mathbf{p}_{t,i})] \quad (10)$$

Here, the desired metric that matches correspondent points of the clouds is the Mahalanobis distance. However, the classical ICP approach only allows for metrics that assume the constant isotropic and time-invariant case of our posterior. Put differently $Q_{t,i} = q I_3$, where I_3 is the 3×3 identity matrix. With respect to the overall minimization, $Q_{t,i}$ is now constant and can be omitted. The optimization therefore reduces to

$$\operatorname{argmin}_{\Delta R, \Delta \mathbf{t}} \sum_{t,i} [\Delta R \mathbf{p}_{t,i} + \Delta \mathbf{t} - m(\mathbf{p}_{t,i})]^2 \quad (11)$$

Still, the problem is a non-linear least squares problem, because the rotation is a non-linear function of the angles involved.

The ICP algorithm itself is stated in [10] and its convergence is proven. We therefore waive a complete notation of the iteration scheme. The reader should note that ΔR and $\Delta \mathbf{t}$ both incorporate global transformation parameters that align reference and sample as a whole. Thus, both point clouds are considered to be rigid.

Linear Optimization Approach

Commonly, it is convenient to linearize the optimization criterion. For that purpose, we simply expect aligning rotations at each step of ICP to be small and approximate $\cos \alpha$ by 1 and $\sin \alpha$ by α . In our case, this is a reasonable assumption, because we seek to register point clouds that are pre-aligned to some extent. Accordingly, $R_{\Delta\phi\Delta\theta\Delta\psi}$ denotes

$$R_{\Delta\phi\Delta\theta\Delta\psi} = \Delta R \approx \begin{pmatrix} 1 & -\Delta\psi & \Delta\theta \\ \Delta\psi & 1 & -\Delta\phi \\ -\Delta\theta & \Delta\phi & 1 \end{pmatrix} \quad (12)$$

Now that the optimization problem is linear, it can easily be solved exactly (relatively to the approximations). For this purpose, the partial derivatives of Eq. 11 with respect to the dependents Δx , Δy , Δz , $\Delta\phi$, $\Delta\theta$ and $\Delta\psi$ are set to zero, yielding 6 equations containing the 6 sought variables. This gives us a matrix equation of the form $A \mathbf{x} = -\mathbf{b}$, where A is a 6×6 matrix collecting the gradients of the optimization criterion, \mathbf{b} is a 6-dimensional vector incorporating the constant components of the derivatives and $\mathbf{x} = (\Delta x \ \Delta y \ \Delta z \ \Delta\phi \ \Delta\theta \ \Delta\psi)^T$.



Fig. 3. Registration results obtained by considering reference (light blue) and sample (black) to be rigid. The side view reveals, that there remain huge divergences (red boxes) between numerous parts of the two point clouds.

Here, the exact determination of A and \mathbf{b} be omitted, because this solely involves basic maths. However, we shall remark the important fact that A is symmetric and positive definite. This is an important property a variety of particularly efficient numerical algorithms take advantage of. For example, $A\mathbf{x} = -\mathbf{b}$ may be solved using the Cholesky Decomposition.

Preliminary Results

Fig. 3 shows two point clouds, one being the reference $\mathbf{p}_{t',i'}$, the other being the sample $\mathbf{p}_{t,i}$. Both have been registered using rigid body transformations as suggested by Besl and McKay. The scans inducing the point clouds were acquired independently during two flights of our laser-equipped helicopter. The classical approach obviously performs poorly in that there remain huge divergences between numerous parts of the two data sets.

To gain more insight into the apparent lacks that occur when applying standard ICP to helicopter range observations, we need to investigate the reasons for its failing. By looking at Fig. 3, we find that the errors in our observations are nearly uncorrelated. This implies that the variances and covariances of the pose estimation and the measurement distribution are significant. The attempt to apply rigid body transformations to our sample is obviously not capable of compensating for this kind of errors.

5 Aligning Non-rigid Models

In Sec. 4.2 we have shown that assuming an alignment model consisting of several individual scans to be rigid leads to results that implicitly induce a vast amount of false differences. Therefore, we now present a novel approach that is capable of dealing with high, randomly distributed estimation errors ICP does not consider.

The key idea is to extent the classical alignment approach by treating the sample we wish to align to a reference model as a non-rigid point cloud. More specifically, each scan is considered as rigid body, whilst it remains subject to local transformations as a whole. Based on that, we are then able to involve dependency assumptions that account for correlations between observation errors.

Extended Optimization Criterion

In a first step we simply allow for scans to be transformed independently instead of assuming a global ΔR and a global $\Delta \mathbf{t}$. For that purpose, we introduce a rotation matrix $\Delta R_t = R_{\Delta\phi_t\Delta\theta_t\Delta\psi_t}$ and a translation vector $\Delta \mathbf{t}_t = (\Delta x_t \Delta y_t \Delta z_t)^T$. Both are applying to all $\mathbf{p}_{t,i}$ that were induced by readings acquired at time t . We now modify the simplified posterior from Eq. 11 to incorporate the scan-related transformation parameters and henceforth seek to find

$$\operatorname{argmin}_{\Delta R_t, \Delta \mathbf{t}_t} \sum_{t,i} [\Delta R_t \mathbf{p}_{t,i} + \Delta \mathbf{t}_t - m(\mathbf{p}_{t,i})]^2 \quad (13)$$

At first glance, this only seems to be a slight modification, but Eq. 13 essentially differs from Eq. 11 in that we now use a vast sequence of scan-related, local rotation matrices ΔR_t and translation vectors $\Delta \mathbf{t}_t$ instead of a single scan-common, global ΔR and a global $\Delta \mathbf{t}$ to align reference and sample.

Nevertheless, this gives us a linear optimization problem we are able to solve exactly. Setting the partial derivatives with respect to the dependents Δx_t , Δy_t , Δz_t , $\Delta \phi_t$, $\Delta \theta_t$ and $\Delta \psi_t$ for all t to zero, we yield $6N$ equations with $6N$ unknown variables. That is, the dimensionality of our optimization problem grew by the factor N . However, we collect the elements to a linear matrix equation of the form $A^* \mathbf{x}^* = -\mathbf{b}^*$, where A^* is once again the matrix containing all the gradients of our optimization criterion and now incorporates $6N \times 6N$ elements. As usual, the $6N$ -dimensional vector \mathbf{b}^* accumulates the constant components of the derivatives and \mathbf{x}^* represents all $6N$ sought transformation parameters.

A closer look to the matrix equation reveals that A^* remains symmetric and positive definite. Although its dimensionality increased tremendously, non-zero elements can solely be found along the main diagonal. The reason for that is covered in our model assumption. In particular, we considered all pose estimations and measurements to be independent in terms of noise. The missing gradients to the left and the right of the main diagonal therefore account for missing dependencies in the posterior. Hence, A^* is said to be sparse and the stated problem is usually referred to as a *sparse energy minimization problem*, for which a rich family of efficient solvers exist. Amongst them, the conjugate gradient (CG) algorithm is the most prominent iterative method.

Descriptions of CG can be found in contemporary textbooks on optimization [15]. The details of the algorithm are omitted for brevity.

Plausible Dependency Assumptions

The alignment approach stated so far is based on a non-rigid model that assembles rigid scan lines in a non-specific way. Estimation noise is assumed to be independent and consequently, scans are considered as incoherent. However, this is a very weak assumption that may lead to unexpected registration results. The Kalman filter estimating the helicopter's pose at each step t , for example, emits sequences of state vectors ξ_t that are known to be statistically dependent over time.

To account for correlated observation errors of this type, we therefore assume non-zero covariances between temporally successive observations. Put differently, the idea may be compared to attaching optimization constraints in the form of little springs between scans arising from time t and time $t - 1$. We modify the posterior from Eq. 13 accordingly to take advantage of a translational covariance parameter Δq_t and a rotational covariance Δq_r . Henceforth, we seek to find

$$\begin{aligned} & \operatorname{argmin}_{\Delta R_t, \Delta \mathbf{t}_t} \sum_{t,i} [\Delta R_t \mathbf{p}_{t,i} + \Delta \mathbf{t}_t - m(\mathbf{p}_{t,i})]^2 \\ & + \frac{1}{\Delta q_t} \sum_t [\Delta \mathbf{t}_t - \mathbf{t}_{\Delta t-1}]^2 + \frac{1}{\Delta q_r} \sum_t [\Delta \mathbf{r}_t - \Delta \mathbf{r}_{t-1}]^2 \end{aligned} \quad (14)$$

Here, we introduced another vector $\Delta \mathbf{r}_t = (\Delta \phi_t \ \Delta \theta_t \ \Delta \psi_t)^T$ that simply accumulates the corresponding Euler angles from our rotation matrix $\Delta R_t = \Delta R_{\Delta \phi_t \Delta \theta_t \Delta \psi_t}$.

To exactly solve the linear optimization problem stated in Eq. 14, we are once again setting the partial derivatives with respect to the dependents Δx_t , Δy_t , Δz_t , $\Delta \phi_t$, $\Delta \theta_t$ and $\Delta \psi_t$ for all t to zero. Furthermore, we collect the gradients and constants to yield a linear matrix equation of the familiar form and structure. Again, the gradient matrix remains sparse as well as symmetric and positive definite. Hence, applying the CG algorithm to solve the system of linear equations remains a feasible approach.

Online Approach

The approach presented so far constructs a global optimization problem that incorporates knowledge on the complete data sets. However, a lot of applications exist where an online algorithm is a much more favorable solution. We are therefore briefly presenting a slightly modified, iterative approach that incorporates range scans as they occur.

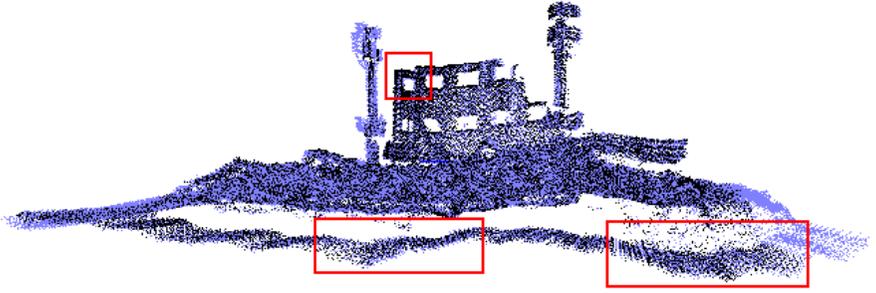


Fig. 4. Registration results obtained by applying the non-rigid approach presented in this paper. Reference (light blue) and sample (black) are aligned near-perfectly.

The key idea of our alternative, real-time computable alignment method, which can directly be derived from Eq. 14, additionally considers a translational variance parameter s_t as well as a rotational variance s_r . Thus, the online posterior optimization shall be denoted

$$\begin{aligned} \operatorname{argmin}_{\Delta R_t, \Delta \mathbf{t}_t} \sum_i [\Delta R_t \mathbf{p}_{t,i} + \Delta \mathbf{t}_t - m(\mathbf{p}_{t,i})]^2 \\ + \frac{1}{\Delta q_t} [\Delta \mathbf{t}_t - \mathbf{t}_{\Delta t-1}]^2 + \frac{1}{\Delta q_r} [\Delta \mathbf{r}_t - \Delta \mathbf{r}_{t-1}]^2 + \frac{1}{s_t} \Delta \mathbf{t}_t^2 + \frac{1}{s_r} \Delta \mathbf{r}_t^2 \end{aligned} \quad (15)$$

and has to be determined for every arising observation at time t .

5.1 Alignment Results

Alignment results are depicted in Fig. 4. Compared to the preliminary results shown in Fig. 3 that were obtained by applying classical ICP, our approach provides a near-perfect registration of reference and sample, even though the helicopter's pose estimation is exceptionally imprecise and noisy.

In order to yield a quantitative evaluation of our alignment results, we assume a normal distribution of all difference vectors between registered sample points and corresponding, that is nearest, reference points. This approach allows for the mean and the standard deviation of the Gaussian alignment error for each of the spatial dimensions to be determined empirically.

Table 1 presents the alignment errors achieved by applying the classical, rigid method as well as the non-rigid offline and the non-rigid online approach proposed in this paper to the helicopter's observations. Obviously, all methods perform fairly well in that they are capable of shifting the distributions' means into the origin. However, the rigid registration does not compensate for the error's variant components, whilst both non-rigid approaches significantly narrow the Gaussians, thus clearly indicating better alignment results.

Table 1. Alignment errors (in meters) obtained by applying the classical registration as well as the approaches presented. Significantly smaller standard deviations of the error distributions achieved by applying the non-rigid methods clearly indicate better alignment results.

| Alignment/Error Mean | Std. Deviation |
|----------------------|--|
| None | (0.017 0.032 -0.063) (0.117 0.129 0.212) |
| Rigid | (0.001 0.000 0.002) (0.094 0.090 0.153) |
| Non-Rigid, Offline | (0.002 0.002 0.004) (0.058 0.054 0.061) |
| Non-Rigid, Online | (0.002 0.003 0.000) (0.053 0.052 0.058) |

6 Change Detection

This section of the paper deals with the question of how to actually identify relevant changes in the previously aligned models. It will become apparent that in order to reliably detect changes in naturally erroneous and noisy sensor data, several assumptions are to be made. Therefore a probabilistic approach with the capability of considerably improving the detection results is presented.

6.1 Change Model

The critical question in change detection is whether a surface in the real world has changed. In our context, a change is manifested by the fact that two “true” measurements differ, that is the environment has been altered between two consecutive scans. The probability for this to happen shall be given by the expression

$$pr(\hat{\mathbf{p}}_{\mathbf{t},i} \neq \hat{\mathbf{p}}_{\mathbf{t}',i'}) \quad (16)$$

We introduce a joint probability distribution giving an estimate of changes in the environment. Therefore, we are considering two distinct cases.

Case 1: The world has not changed. Consider the range scan $r_{t,i}$ and its induced point $\mathbf{p}_{\mathbf{t},i}$ along with the acquisition pose $\xi_{\mathbf{t}}$. Furthermore, let $\bar{\mathbf{p}}_{\mathbf{t}',i'}$ be the nearest neighbor to $\mathbf{p}_{\mathbf{t},i}$, and $\bar{r}_{t',i'}$ the associated range sensed relative to the same pose. Then our approach gives us $pr(r_{t,i} | \bar{\mathbf{p}}_{\mathbf{t}',i'}, \xi_{\mathbf{t}}, \text{false}) \sim \mathcal{N}(\bar{r}_{t',i'}, 2s)$. The variance of $2s$ accounts for the two measurement noise variables involved in this process, one from each of the scans. We assume independence in measurement noise, hence the variances of both noise variables are simply added.

Case 2: The world has changed. Since we make no assumptions as to how the world changes, the best we can assume is a uniform distribution over

scan lengths. Thus, we define $pr(r_{t,i} | \bar{\mathbf{p}}_{t',i'}, \boldsymbol{\xi}_t, \text{true}) \sim \mathcal{U}(0, r_{\max})$, with \mathcal{U} denoting the desired uniform distribution.

These two probabilities enable us to arrive at a probabilistic expression to estimate when the world changed. In particular, Bayes rule suggests:

$$\begin{aligned} pr(\text{true} | r_{t,i}, \bar{\mathbf{p}}_{t',i'}, \boldsymbol{\xi}_t) & \quad (17) \\ = \frac{pr(r_{t,i} | \bar{\mathbf{p}}_{t',i'}, \boldsymbol{\xi}_t, \text{true}) \cdot p(\text{true})}{pr(r_{t,i} | \bar{\mathbf{p}}_{t',i'}, \boldsymbol{\xi}_t, \text{true}) \cdot p(\text{true}) + pr(r_{t,i} | \bar{\mathbf{p}}_{t',i'}, \boldsymbol{\xi}_t, \text{false}) \cdot p(\text{false})} \end{aligned}$$

Here $p(\text{true})$ is the prior probability of a change, and $p(\text{false}) = 1 - p(\text{true})$ the probability that the world remains static at any given location.

The logarithm of this expression can be approximated using Jensen's inequality:

$$\begin{aligned} \log pr(\text{true} | r_{t,i}, \bar{\mathbf{p}}_{t',i'}, \boldsymbol{\xi}_t) & \\ = \log[pr(r_{t,i} | \bar{\mathbf{p}}_{t',i'}, \boldsymbol{\xi}_t, \text{true}) \cdot p(\text{true})] - \log[pr(r_{t,i} | \bar{\mathbf{p}}_{t',i'}, \boldsymbol{\xi}_t, \text{true}) \cdot p(\text{true})] & \\ - \log[pr(r_{t,i} | \bar{\mathbf{p}}_{t',i'}, \boldsymbol{\xi}_t, \text{false}) \cdot p(\text{false})] & \\ = -\log[pr(r_{t,i} | \bar{\mathbf{p}}_{t',i'}, \boldsymbol{\xi}_t, \text{false}) \cdot p(\text{false})] & \\ = -\log p(\text{false}) + \log \sqrt{4\pi s} + \frac{1}{2} (r_{t,i} - \bar{r}_{t',i'})^T (2s)^{-1} (r_{t,i} - \bar{r}_{t',i'}) & \\ = \text{const.} + \frac{1}{2} (r_{t,i} - \bar{r}_{t',i'})^T (2s)^{-1} (r_{t,i} - \bar{r}_{t',i'}) & \quad (18) \end{aligned}$$

Therefore, we are now able to determine how probable a likely change is. We compute a simple quadratic distance of the type

$$d_{t,i} = (r_{t,i} - \bar{r}_{t',i'})^T (2s)^{-1} (r_{t,i} - \bar{r}_{t',i'}) \quad (19)$$

Assuming that a change with a probability $pr(\text{true} | r_{t,i}, \bar{\mathbf{p}}_{t',i'}, \boldsymbol{\xi}_t) > 0.5$ may be significant, the distance $d_{t,i}$ has to be compared to the following threshold

$$\begin{aligned} d_{t,i} & \stackrel{?}{>} 2 \log pr(\text{true} | r_{t,i}, \bar{\mathbf{p}}_{t',i'}, \boldsymbol{\xi}_t) - 2 \text{const.} \\ & = 2 \log 0.5 + 2 \log p(\text{false}) - 2 \log \sqrt{4\pi s} \quad (20) \end{aligned}$$

If the comparison evaluates to "true", a probable change can be marked as detected.

6.2 Change Detection Results

The change detection results shown in Figs. 5, 7 and 6 were all obtained by preliminarily registering reference and sample according to our alignment

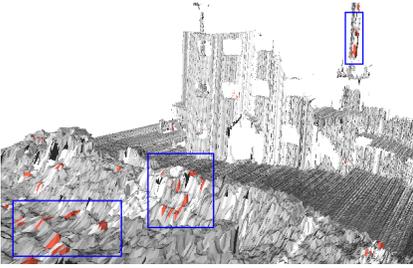


Fig. 5. Change detection results obtained by evaluating the Euclidean distance between reference and sample. Changes (red) are commonly occurring in scattered areas.

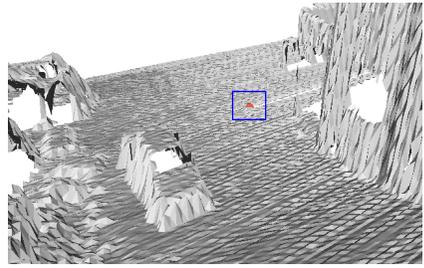


Fig. 6. The intentionally introduced change (a small box with an edge length of about 20 cm) has been detected correctly by our approach.

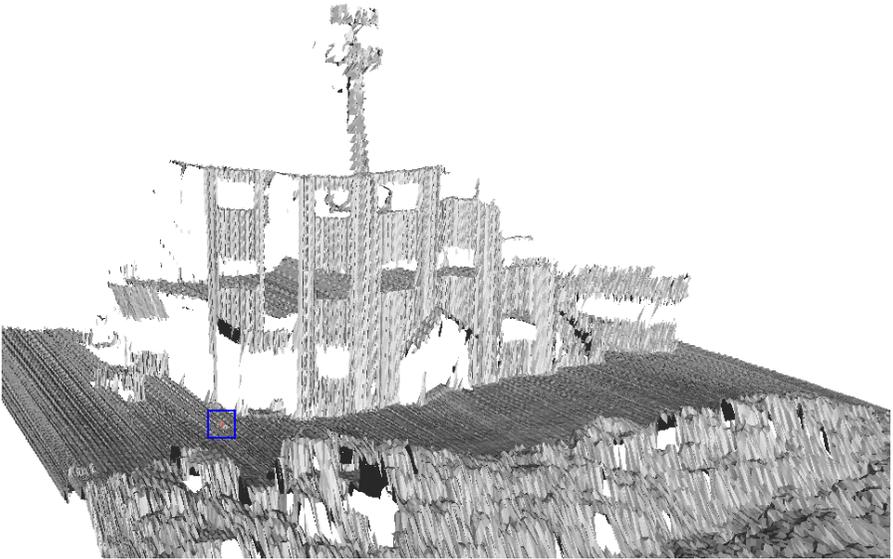


Fig. 7. By applying the probabilistic threshold analysis presented in our approach, all unlikely changes could successfully be extracted from the results. The “true” change is marked.

approach. A small cubical box with a maximum edge length of about 20 cm has been placed within the area examined by the helicopter.

Fig. 5 illustrates that solely evaluating the Euclidean distances between both point clouds explicitly fails in scattered areas of the environment. The superiority of our probabilistic change detection approach is visualized in Fig. 7, where no false changes were marked within these parts. In Fig. 6 a closer look to our intentionally introduced change reveals that it has been detected correctly after all.

7 Discussion and Future Work

We presented a probabilistic approach for alignment and change detection using range sensor data. The alignment method provides a non-rigid point cloud registration that near-perfectly deals with high estimation errors. Changes between data sets are detected using a probabilistic threshold that is capable of differentiating between likely and unlikely changes. Even small intentional modifications are detected reliably, whilst errors are excluded.

There is ample opportunity for future research. On the basis of a reliable change detection, changes could be classified. That would allow for a variety of applications in robotics to superiorly deal with dynamic environments. Another opportunity pertains to the learning component of our work: In addition to learning alignment models, it should be possible to learn the actual noise models of the sensors and the pose estimation system. We also suspect that further improvements can be achieved by better involving the exact physical noise characteristics of a sensor.

References

1. Singh, A. 1989. Digital change detection techniques using remotely-sensed data. *International Journal of Remote Sensing*, 10(6):989-1004.
2. Dai, X., Khorram, S. 1998. The effects of image misregistration on the accuracy of remotely sensed change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 36(5):1566-1577.
3. Chavez, P. S., MacKinnon, D. J. 1994. Automatic detection of vegetation changes in the southwestern United States using remotely sensed images. *Photogramm Eng. Remote Sensing*, 60(5):571-582.
4. Carlotto, M. J. 1997. Detection and analysis of change in remotely sensed imagery with application to wide area surveillance. *IEEE Transactions on Image Processing*, 6(1):189-202.
5. Murakami, H., Nakagawa, K., Hasegawa, H., Shibata, T., Iwanami, E. 1999. Change detection of buildings using an airborne laser scanner. *Journal of Photogrammetry and Remote Sensing*, 54(2-3):148-152.
6. Vu, T. T., Matsuoka, M., Yamazaki, F. 2004. Lidar-based change detection of buildings in dense urban areas. *IEEE 200 International Geoscience and Remote Sensing Symposium*, CD-ROM, 4p, 2004.9.
7. Coppin, P. R., Bauer, M. E. 1996. Digital change detection in forest ecosystems with remote sensing imagery. *Remote Sensing Reviews*, 13(3-4):207-234.
8. Hsiao, K. H., Liu, J. K., Yu, M. F., Tseng, Y. H. 1999. Change detection of landslide terrains using ground-based lidar data. *Proceedings of the XXth ISPRS Congress*, Istanbul, Turkey.
9. Habib, A. F., Lee, Y.-R., Morgan, M. 2001. Surface matching and change detection using a modified Hough transformation for robust parameter estimation. *Photogrammetric Record*, 17(98):303-315.
10. Besl, P. and McKay, N. 1992. A method for registration of 3d shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239-256.

11. Chen, Y. and Medioni, G. 1992. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10:145-155.
12. Chui, H., Rangarajan, A. 2000. A new point matching algorithm for non-rigid registration. *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.
13. Haehnel, D., Thrun, S., Burgard, W. 2003. An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico.
14. Whalley, M., Freed, M., Harris, R., Takahashi, M., Schulein, G., and Howlett, J. 2005. Design, integration, and flight test results for an autonomous surveillance helicopter. *AHS International Specialists Meeting on Unmanned Rotorcraft*, Mesa, AZ.
15. Press, W. H. 1988. *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, Cambridge, New York.

An Efficient Extension of Elevation Maps for Outdoor Terrain Mapping

Patrick Pfaff and Wolfram Burgard

Department of Computer Science, University of Freiburg, Germany,
{pfaff, burgard}@informatik.uni-freiburg.de

Summary. Elevation maps are a popular data structure for representing the environment of a mobile robot operating outdoors or on not-flat surfaces. Elevation maps store in each cell of a discrete grid the height of the surface the corresponding place in the environment. The use of this $2\frac{1}{2}$ -dimensional representation, however, is disadvantageous when it is used for mapping with mobile robots operating on the ground, since vertical or overhanging objects cannot be represented appropriately. Such objects furthermore can lead to registration errors when two elevation maps have to be matched. In this paper we propose an approach that allows a mobile robot to deal with vertical and overhanging objects in elevation maps. We classify the points in the environment according to whether they correspond to such objects or not. We also describe a variant of the ICP algorithm that utilizes the classification of cells during the data association. Experiments carried out with a real robot in an outdoor environment demonstrate that the scan matching process becomes significantly more reliable and accurate when our classification is used.

1 Introduction

The problem of learning maps with mobile robots has been intensively studied in the past. In the literature, different techniques for representing the environment of a mobile robot prevail. Topological maps aim at representing environments by graph-like structures, where edges correspond to places, and arcs to paths between them. Geometric models, in contrast, use geometric primitives for representing the environment. Whereas topological maps have the advantage to better scale to large environments, they lack the ability to represent the geometric structure of the environment. The latter, however, is essential in situations, in which robots are deployed in potentially unstructured outdoor environments where the ability to traverse specific areas of interest needs to be known accurately. However, full three-dimensional models typically have too high computational demands for a direct application on a mobile robot.

Elevation maps have been introduced as a more compact $2\frac{1}{2}$ -dimensional representation. An elevation map consists of a two-dimensional grid in which each cell stores the height of the territory. This approach, however, can be problematic when a

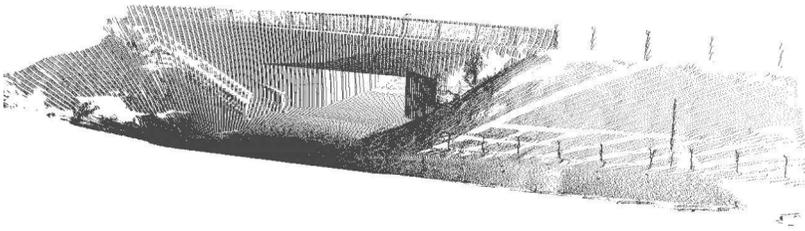


Fig. 1. Scan (point set) of a bridge recorded with a mobile robot carrying a SICK LMS laser range finder mounted on a pan/tilt unit.

robot has to utilize these maps for navigation or when it has to register two different maps in order to integrate them. For example, consider the three-dimensional data points shown in Figure 1. They have been acquired with a mobile robot standing in front of a bridge. The resulting elevation map, which is computed from averaging over all scan points that fall into a cell of a horizontal grid (given a vertical projection), is depicted in Figure 2. As can be seen from the figure, the underpass has completely disappeared and the elevation map shows a non-traversable object. Additionally, when the environment contains vertical structures, we typically obtain varying average height values depending on how much of this vertical structure is contained in a scan. Accordingly, if one registers two such elevation maps, one obtains incorrect alignments.

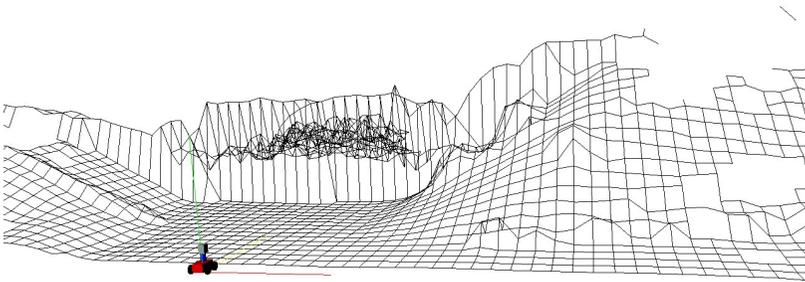


Fig. 2. Standard elevation map computed for the outdoor environment depicted in Figure 1. The passage under the bridge has been converted into a large un-traversable object.

In this paper we present a system for mapping outdoor environments with elevation maps. Our algorithm transforms range scans into local elevation maps and combines these local elevation maps using a variant of the ICP algorithm [3]. In our elevation maps, we classify locations in the environment into four classes, namely locations sensed from above, vertical structures, vertical gaps, and traversable cells. The advantage of this classification is twofold. First, the robot can represent obstacles corresponding to vertical structures like walls of buildings. It also can deal with

overhanging structures like branches of trees or bridges. Furthermore, the classification can be utilized in the ICP algorithm to more robustly match local elevation maps. We present experimental results illustrating the advantages of our approach regarding the representation aspect as well as regarding the robust matching.

This paper is organized as follows. After discussing related work in the following section, we will describe our extension to the elevation maps in Section 3. In Section 4 we then describe how to incorporate our classification into the ICP algorithm used for matching elevation maps. Finally, we present experimental results in Section 5.

2 Related Work

The problem of learning three-dimensional representations has been studied intensively in the past. One of the most popular representations are raw data points or triangle meshes [1, 7, 12, 15]. Whereas these models are highly accurate and can easily be textured, their disadvantage lies in the huge memory requirement, which grows linearly in the number of scans taken. An alternative is to use three-dimensional grids [9] or tree-based representations [13], which only grow linearly in the size of the environment. Still, the memory requirements for such maps in outdoor environments are high.

In order to avoid the complexity of full three-dimensional maps, several researchers have considered elevation maps as an attractive alternative. The key idea underlying elevation maps is to store the $2\frac{1}{2}$ -dimensional height information of the terrain in a two-dimensional grid. Bares et al. [2] as well as Hebert et al. [4] use elevation maps to represent the environment of a legged robot. They extract points with high surface curvatures and match these features to align maps constructed from consecutive range scans. Parra et al. [11] represent the ground floor by elevation maps and use stereo vision to detect and track objects on the floor. Singh and Kelly [14] extract elevation maps from laser range data and use these maps for navigating an all-terrain vehicle. Ye and Borenstein [16] propose an algorithm to acquire elevation maps with a moving vehicle equipped with a tilted laser range scanner. They propose special filtering algorithms to eliminate measurement errors or noise resulting from the scanner and the motions of the vehicle. Lacroix et al. [6] extract elevation maps from stereo images. They use a two-dimensional grid and store in each cell of this grid the average height. Hygounenc et al. [5] construct elevation maps in an autonomous blimp using 3d stereo vision. They propose an algorithm to track landmarks and to match local elevation maps using these landmarks. Olson [10] describes a probabilistic localization algorithm for a planetary rover that uses elevation maps for terrain modeling.

Compared to these techniques the contribution of this paper lies in two aspects. First, we classify the points in the elevation map into horizontal points seen from above, vertical points, and gaps. This classification is important especially when a rover is deployed in an urban environments. In such environments, typical structures like the walls of buildings cannot be represented in standard elevation maps. Second,

we describe how this classification can be used to enhance the matching of different elevation maps.

3 Extended Elevation Maps

As already mentioned above, elevation maps are $2\frac{1}{2}$ -dimensional representation of the environment. They maintain a two-dimensional grid and maintain in every cell of this grid an estimate about the height of the terrain at the corresponding point of the environment. To correctly reflect the actual steepness of the terrain, a common assumption is that the initial tilt and the roll of the vehicle is known.

When updating a cell based on sensory input, we have to take into account, that the uncertainty in a measurement increases with the distance measured due to errors in the tilting angle. In our current system, we apply a Kalman filter to estimate the parameters $\mu_{1:t}$ and $\sigma_{1:t}$ about the elevation in a cell and its standard deviation. We apply the following equations to incorporate a new measurement z_t with standard deviation σ_t at time t [8]:

$$\mu_{1:t} = \frac{\sigma_t^2 \mu_{1:t-1} + \sigma_{1:t-1}^2 z_t}{\sigma_{1:t-1}^2 + \sigma_t^2} \quad (1)$$

$$\sigma_{1:t}^2 = \frac{\sigma_{1:t-1}^2 \sigma_t^2}{\sigma_{1:t-1}^2 + \sigma_t^2} \quad (2)$$

Note that the application of the Kalman filter allows us to take into account the uncertainty of the measurement. In our current system, we apply a sensor model, in which the variance of the height of a measurement increases linearly with the distance of the corresponding beam. This process is indicated in Figure 3.

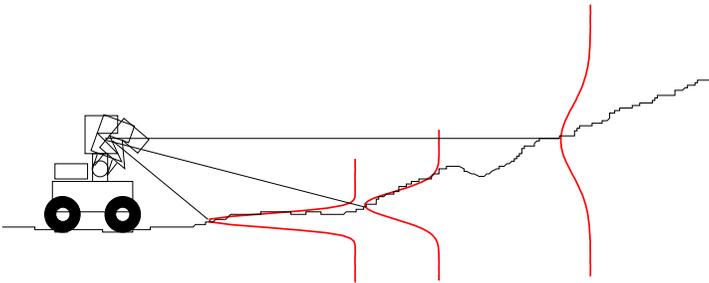


Fig. 3. Variance of a height measurements depending on the distance of the beam.

In addition we need to identify which of the cells of the elevation map correspond to vertical structures and which ones contain gaps. In order to determine the class of a cell, we first consider the variance of the height of all measurements falling into this cell. If this value exceeds a certain threshold, we identify it as a point that has not

been observed from above. We then check, whether the point set corresponding to a cell contains gaps exceeding the height of the robot. When a gap has been identified, we determine the minimum traversable elevation in this point set.

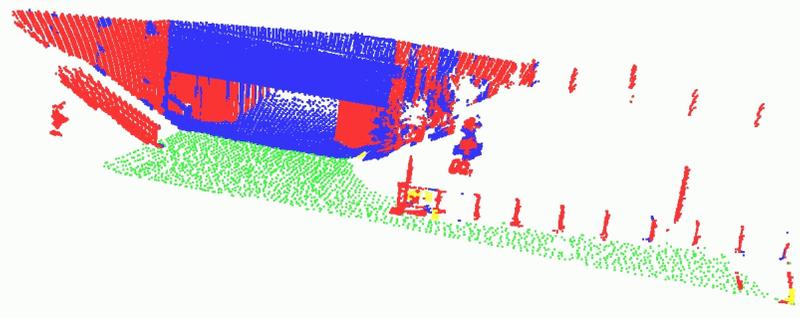


Fig. 4. Labeling of the data points depicted in Figure 2 according to their classification. The different colors/grey levels indicate the individual classes.

Figure 4 shows the same data points already depicted in Figure 2. The classes of the individual cells in the elevation map are indicated by the different colors/grey levels. The blue/dark points indicate the data points above a gap. The red/medium grey values indicate cells that are classified as vertical. The green/light grey values, however, indicate traversable terrain. Note that the not traversable cells are not shown in this figure.

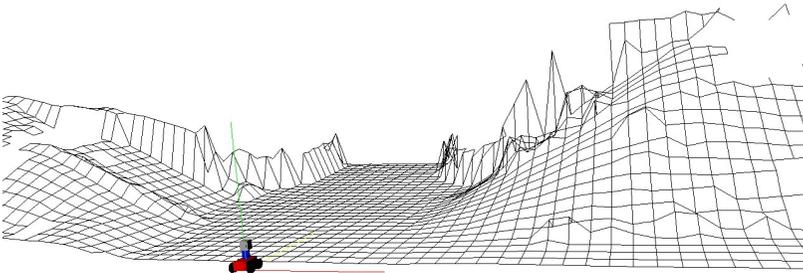


Fig. 5. Extended elevation map for the scene depicted in Figure 1.

A major part of the resulting elevation map extracted from this data set is shown in Figure 5. As can be seen from the figure, the area under the bridge can now be represented appropriately by ignoring data points above the lowest surface. This in turn enables the robot to plan a path through the passage under the bridge.

4 Efficient Matching of Elevation Maps in 6 Dimensions

To integrate several local elevation maps into a single global elevation map we need to be able register two maps relative to each other. In our current system, we apply the ICP algorithm for this purpose. The goal of the matching process is to minimize an error function defined over two point sets $X = \{x_1, \dots, x_L\}$ and $Y = \{y_1, \dots, y_L\}$, where each pair x_i and y_i is assumed to be the points that corresponding to each other. We are interested in the rotation R and the translation t that minimizes the following cost function:

$$E(R, t) = \frac{1}{n} \sum_{l=1}^L \|x_l - Ry_l - t\|^2, \quad (3)$$

where $\|\cdot\|$ is a distance function that takes into account the variance of the Gaussians corresponding to each pair x_i and y_i .

In principle, one could define this function to directly operate on the height values and their variance when aligning two different elevation maps. The disadvantage of this approach, however, is that in the case of vertical objects, the resulting height seriously depends on the view point. The same vertical structure may lead to varying heights in the elevation map when sensed from different points. In practical experiments we observed that this introduces serious errors and often prevents the ICP algorithm from convergence. To overcome this problem, we separate Equation (3) into four components each minimizing the error over the individual classes of points. The first two classes consist of the cells corresponding to vertical objects and gaps. The latter two classes contain only cells whose points have been sensed from above. To increase the efficiency of the matching process, we only consider a subset of these cells. In practical experiments we found out that traversable cells and edge cells yield the best registration results. The traversable cells are those cells for which the elevation of the surface normal obtained from a plane fitted to the local neighborhood exceeds 83 degrees. Additionally, we consider edge cells, i.e., cells which lie more than 20cm above their neighboring points.

Let us assume that $\alpha_1, \dots, \alpha_{N_\alpha}$ and $\alpha'_1, \dots, \alpha'_{N_\alpha}$ are the corresponding vertical points, $\beta_1, \dots, \beta_{N_\beta}$ and $\beta'_1, \dots, \beta'_{N_\beta}$ are the vertical gaps, $\gamma_1, \dots, \gamma_{N_\gamma}$ and $\gamma'_1, \dots, \gamma'_{N_\gamma}$ are the edge points, and $\delta_1, \dots, \delta_{N_\delta}$ and $\delta'_1, \dots, \delta'_{N_\delta}$ are the traversable cells. The resulting error function then is

$$E(R, t) = \underbrace{\sum_{n=1}^{N_\alpha} d(\alpha_n, \alpha'_n)}_{\text{vertical objects}} + \underbrace{\sum_{n=1}^{N_\beta} d(\beta_n, \beta'_n)}_{\text{vertical gaps}} + \underbrace{\sum_{n=1}^{N_\gamma} d(\gamma_n, \gamma'_n)}_{\text{edge cells}} + \underbrace{\sum_{n=1}^{N_\delta} d(\delta_n, \delta'_n)}_{\text{traversable cells}}, \quad (4)$$

where $d(x, y) = \|x - Ry - t\|^2$.

Figure 6 illustrates how two elevation maps are aligned over several iterations of the minimization process. Whereas the left column shows the point clouds the right column shows the cells in the elevation map used for minimizing Equation (4). In our

current implementation, each iteration of the ICP algorithm usually takes between one and five seconds on a 2.8GHz Pentium 4. The time necessary to acquire a scan by tilting the laser is 5 seconds.

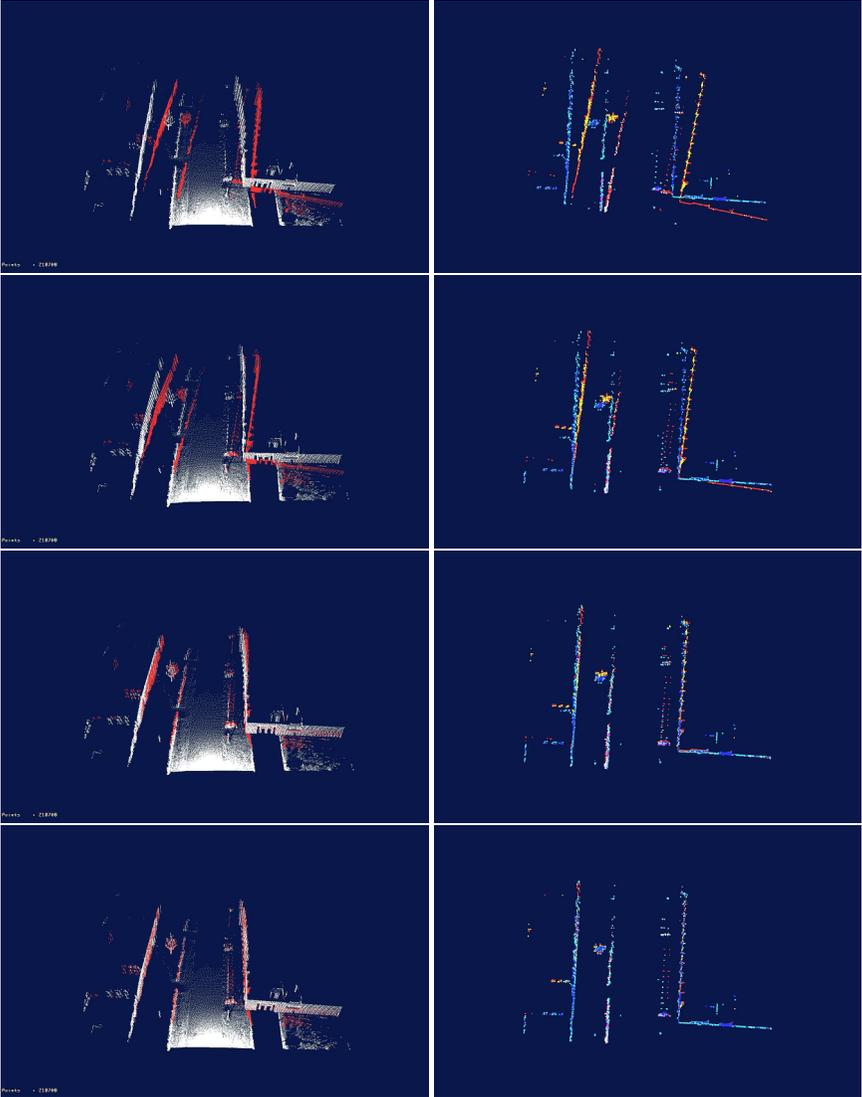


Fig. 6. Incremental registration of two elevation maps. The left column depicts the original point clouds. The right column shows the vertical and edge cells of the elevation maps used by the ICP algorithm. The individual rows correspond to the initial relative pose (top row), alignment after 5 iterations (second row), after 10 iterations (third row) and the final alignment after 30 iterations (fourth row).

In addition to the position and orientation of the vehicle we also have to estimate the tilt and roll of the vehicle when integrating two elevation maps. In practical experiments we found that an iterative scheme, in which we repeatedly estimate the tilt and roll of the robot and then determine the relative position and orientation of the involved elevation maps, improves the registration accuracy. In most cases, two iterations are sufficient to achieve precise matchings and to obtain highly accurate maps from multiple local maps generated from different viewpoints.

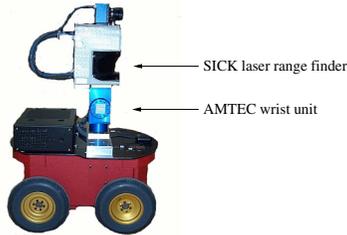


Fig. 7. Robot Herbert used for the experiments.

5 Experimental Results

The approach described above has been implemented and tested on a real robot system and in simulation runs with real data. The robot used to acquire the data is our outdoor robot Herbert, which is depicted in Figure 7. The robot is a Pioneer II AT system equipped with a SICK LMS range scanner and an AMTEC wrist unit, which is used as a pan/tilt device for the laser.

5.1 Learning Accurate Elevation Maps from Multiple Scans

To evaluate our approach we steered our robot Herbert through different areas of our university campus and visually inspected the maps obtained with our technique. In all cases, we obtained highly accurate maps. Figure 8 shows a typical example, in which the robot traveled under the bridge depicted in Figure 1 and then continued driving up a ramp. Along its path the robot generated local elevation maps from 36 scans. The overall number of data points recorded was 9,500,000. The size of each cell in the elevation map is 20 by 20cm. The whole map spans approximately 70 by 30 meters. As can be seen from the figure, the map clearly reflects the details of the environment. Additionally, the matching of the elevation maps is quite accurate.

Figure 9 shows a typical example in which our algorithm yields more accurate maps than the standard approach. In this situation the robot traveled along a paved way and scanned a tree located in front of the scene. Whereas the left image shows the map obtained with the standard elevation map approach, the right image shows

the map obtained with our method. The individual positions of the robot where the scans were taken are also shown in the images. As can be seen from the figures, our method results in more free space around the stem of the tree.

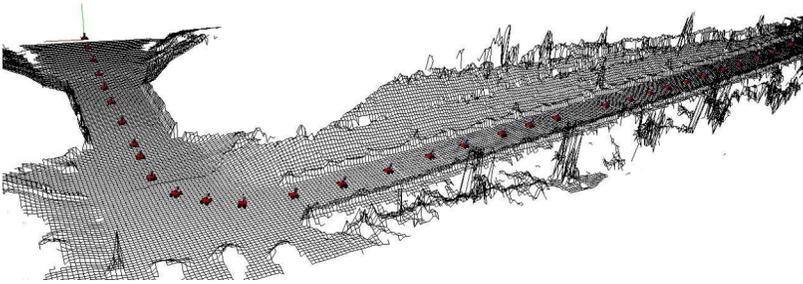


Fig. 8. Elevation map generated from 36 local elevation maps. The size of the map is approximately 70 by 30 meters.

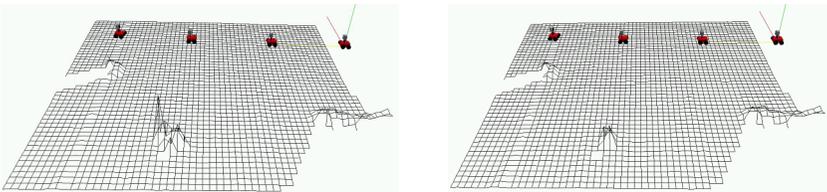


Fig. 9. Maps generated from four local elevation maps acquired with Herbert. The left image shows a standard elevation map. The right image depicts the map obtained with our approach. The peak in front of the scene corresponds to a tree, which is modeled more accurately with our approach.

5.2 Statistical Evaluation of the Accuracy

Additionally, we performed a series of experiments to get a statistical assessment as to whether the classification of the data points into normal, vertical and gap points combined with the sub-sampling of the normal points leads to better registration results. To perform these experiments we considered two different elevation maps for which we computed the optimal relative pose using several runs of the ICP algorithm. We then randomly added noise to the pose of the second map and applied the ICP algorithm to register both maps. We performed two sets of experiments to compare the registration results for the unclassified and the classified point sets. Table 1 shows the individual classes of noise that we added to the true relative pose of the two maps before we started the ICP algorithm. In this experiment described here, we only varied the pose error of the maps and kept the error in the rotations constant. In

particular, we randomly chose rotational displacements from ± 5 degrees around the real relative angle and also varying random displacements in the x and y direction.

Table 1. Displacement classes used to evaluate the performance of the ICP algorithm on the classified and unclassified points extracted from the elevation maps.

| displacement class | max. rot. displ. | max. displ. in x and y |
|--------------------|------------------|----------------------------|
| 1 | ± 5 degrees | $\pm 0.5m$ |
| 2 | ± 5 degrees | $\pm 1.0m$ |
| 3 | ± 5 degrees | $\pm 1.5m$ |
| 4 | ± 5 degrees | $\pm 2.0m$ |
| 5 | ± 5 degrees | $\pm 2.5m$ |

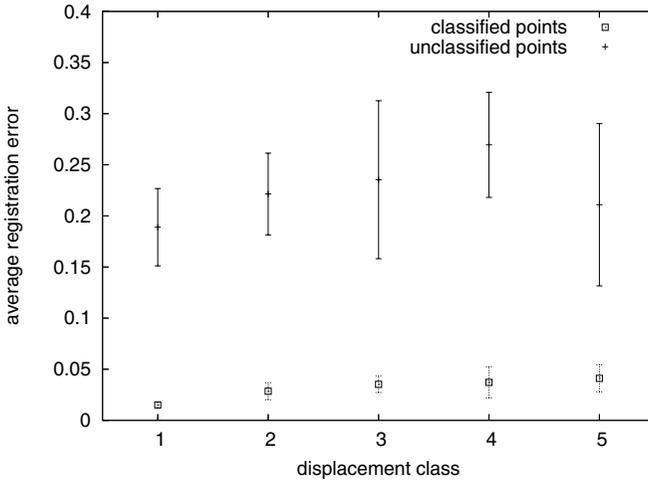


Fig. 10. Average registration error for the individual types of initial displacement.

The resulting average displacement errors after convergence of the ICP algorithm are depicted in Figure 10. As can be seen from the figure, the ICP algorithm performed significantly better on the classified point sets. In this figure, the error bars indicate the $\alpha = 0.05$ confidence level.

Additionally, we evaluated how often the ICP algorithm failed to accurately register the two maps. Figure 11 depicts the normalized divergence frequencies in percent for the individual displacement classes. As this plot illustrates, the utilization of the individual classes in the ICP algorithm leads to a seriously better convergence rate. In additional experiments not reported here we obtained similar results for the different orientational errors.

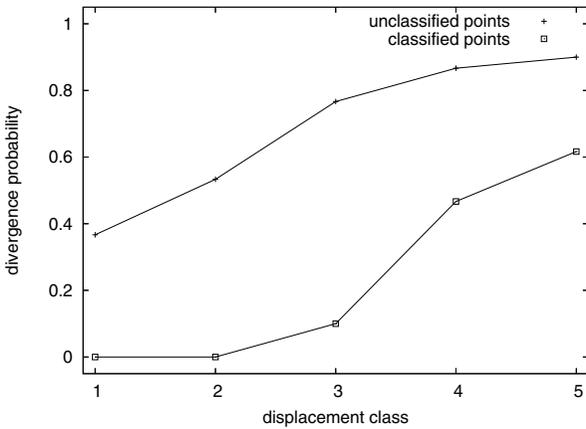


Fig. 11. Number of times the ICP algorithm diverges for the individual initial displacements.

6 Conclusions

In this paper we presented an approach to generate elevation maps from three-dimensional range data acquired with a mobile robot. Our approach especially addresses the problem of acquiring such maps with a ground-based vehicle. On such a system one often encounters situations, in which certain objects, such as walls or trees, are not seen from above. Accordingly, the resulting elevation maps contain incorrect information. The approach in this paper classifies the individual cells of elevation maps into four classes representing parts of the terrain seen from above, vertical objects, overhanging objects such as branches of trees or bridges, and traversable areas. We also presented an extension of the ICP algorithm that takes this classification into account when computing the registration.

Our algorithm has been implemented and tested on a real robot and using outdoor terrain data. Experimental results show that our classification yields more accurate elevation maps, especially in the cases of vertical objects and overhanging objects. Additionally, our extension of the ICP algorithm, which utilizes our classification, produces more accurate alignments and additionally converges more often.

Acknowledgment

This work has partly been supported by the German Research Foundation (DFG) within the Research Training Group 1103 and under contract number SFB/TR-8.

References

1. P. Allen, I. Stamos, A. Gueorguiev, E. Gold, and P. Blaer. Avenue: Automated site modeling in urban environments. In *Proc. of the 3rd Conference on Digital Imaging and Modeling*, pages 357–364, 2001.

2. J. Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, and W. R. L. Whitaker. Ambler: An autonomous rover for planetary exploration. *IEEE Computer Society Press*, 22(6):18–22, 1989.
3. P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.
4. M. Hebert, C. Caillas, E. Krotkov, I.S. Kweon, and T. Kanade. Terrain mapping for a roving planetary explorer. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 997–1002, 1989.
5. E. Hygounenc, I.-K. Jung, P. Souères, and S. Lacroix. The autonomous blimp project of laas-cnrs: Achievements in flight control and terrain mapping. *International Journal of Robotics Research*, 23(4-5):473–511, 2004.
6. S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury and; M. Herrb, and R. Chatila. Autonomous rover navigation on unknown terrains: Functions and integration. *International Journal of Robotics Research*, 21(10-11):917–942, 2002.
7. M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3D scanning of large statues. In *Proc. SIGGRAPH*, pages 131–144, 2000.
8. P.S. Maybeck. The Kalman filter: An introduction to concepts. In *Autonomous Robot Vehicles*. Springer Verlag, 1990.
9. H.P. Moravec. Robot spatial perception by stereoscopic vision and 3d evidence grids. Technical Report CMU-RI-TR-96-34, Carnegie Mellon University, Robotics Institute, 1996.
10. C.F. Olson. Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1):55–66, 2000.
11. C. Parra, R. Murrieta-Cid, M. Devy, and M. Briot. 3-d modelling and robot localization from visual and range data in natural scenes. In *1st International Conference on Computer Vision Systems (ICVS)*, number 1542 in LNCS, pages 450–468, 1999.
12. K. Pervözl, A. Nüchter, H. Surmann, and J. Hertzberg. Automatic reconstruction of colored 3d models. In *Proc. Robotik 2004*, 2004.
13. Hanan Samet. *Applications of Spatial Data Structures*. Addison-Wesley Publishing Inc., 1989.
14. S. Singh and A. Kelly. Robot planning in the space of feasible actions: Two examples. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1996.
15. S. Thrun, C. Martin, Y. Liu, D. Hähnel, R. Emery Montemerlo, C. Deepayan, and W. Burgard. A real-time expectation maximization algorithm for acquiring multi-planar maps of indoor environments with mobile robots. *IEEE Transactions on Robotics and Automation*, 20(3):433–442, 2003.
16. C. Ye and J. Borenstein. A new terrain mapping method for mobile robot obstacle negotiation. In *Proc. of the UGV Technology Conference at the 2002 SPIE AeroSense Symposium*, 1994.

Online Reconstruction of Vehicles in a Car Park

Christopher Tay Meng Keat¹, Cédric Pradalier², and Christian Laugier³

¹ INRIA Rhône Alpes GRAVIR Laboratory tay@inrialpes.fr

² CSIRO ICT Center, Canberra-Australia cedric.pradalier@csiro.au

³ INRIA Rhône Alpes GRAVIR Laboratory christian.laugier@inrialpes.fr

Summary. In this paper, a method of obtaining vehicle hypothesis based on laser scan data only is proposed. This is implemented on the robotic vehicle, CyCab, for navigation and mapping of the static car park environment. Laser scanner data is used to obtain hypothesis on position and orientation of vehicles with Bayesian Programming. Using the hypothesized vehicle poses as landmarks, CyCab performs Simultaneous Localization And Mapping (SLAM). A final map consisting of the vehicle positions in the car park is obtained.

Keywords: Vehicle Detection, Bayesian Programming

1 Introduction

In the framework of automatic vehicles in car parks, a 2D map of a car park is constructed using the autonomous vehicle, CyCab, as the experimental platform. The map of the car park will contain the positions and orientations of the different vehicles in the car park. An application of such a map is to serve as a reference to indicate obstacle positions. Furthermore, it can indicate the state of the parking lots in the car park, and possibly be used in higher level applications such as automatic parking.

Several object based representation of the environment were proposed. Chatila et al. [8] represented the map with a set of lines. More advanced methods in mapping consists of approximating the environment using small polygons [3] [4]. Such methods used a variant of the Expectation-Maximization to generate increasingly accurate 3D maps as more observations are made. In this paper, a higher level of representation (vehicles in this case) of the environment is used instead of fundamental geometrical entities.

Currently, the CyCab robotic vehicle localizes itself in a static environment with respect to artificially installed reflective cones. This localization serves to build a grid map of the environment and has the capability to perform motion planning with safe navigation as described in [1]. However, reflective cones as

artificial landmarks is not a very practical approach. An improvement from an application point of view is to use naturally occurring objects often found in the car park as landmarks. In this paper, vehicles found in the car park are used.

The general idea is to use only the laser data without artificial or predefined landmarks, CyCab will navigate the car park autonomously while generating a map of its environment. While CyCab is travelling around the car park, scanning the environment, CyCab continuously reads in odometric and laser data. At each stage of the iteration, CyCab estimates its own position and orientation of the form (x, y, θ) and creates a map of the car park in the world frame of reference. The origin of the world frame of reference is taken from the initial position of CyCab. The map is then represented as a set of tuples, each containing the position and orientation of the vehicles detected. CyCab hypothesizes the configuration of the vehicles in the surrounding based on the laser scan inputs from laser scanner only.

The advantage of the approach is its ability to map any car park without installing any external aids. With the set of vehicle poses representing the map, a compact and semantic representation of the map can be obtained.

2 System Overview

The mechanism of the entire system can be broken down into three fundamental components, *vehicle detection*, the *simultaneous localization and mapping (slam)* and the *map construction*. CyCab is provided with two types of raw data, the laser scans and CyCab's odometric data. Figure 1 shows the block diagram of the different stages and its interactions:

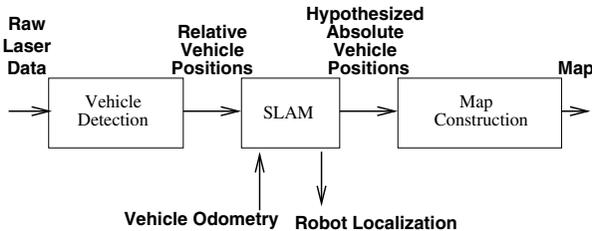


Fig. 1. Overview showing the mapping process

1. **Vehicle Detection:** With only raw laser scan data, vehicle detection constructs hypotheses about the positions and orientation of vehicles in the car park.
2. **SLAM:** Coupled with information about odometry of CyCab, SLAM module makes use of the constructed vehicle hypotheses as landmarks

to localize itself. With its own configuration, SLAM can then calculate the absolute configuration of the vehicles with respect to real world coordinates.

3. **Map Construction:** The hypotheses of vehicles have to be checked for inconsistencies. It is possible for the hypotheses obtained to conflict with a previous corresponding hypothesis. Furthermore, multiple hypothesis SLAM methods such as FastSLAM produces a set of hypotheses, a map construction module is required to merge the information from the different hypotheses to obtain the final map.

3 Vehicle Detection

Vehicle detection is the process of forming possible vehicle hypotheses based on the laser data readings. This problem is addressed in this paper using bayesian programming[2]. Bayesian programs provides us with a framework for encoding our a priori knowledge on the vehicle to infer the possible vehicle poses. In this case, the a priori knowledge consists of the length and width of vehicles which is assumed to be the same across standard vehicles (cars) and that the type of vehicles in the car park is of the same class. The subtle differences in the size of the vehicles can be accomodated for in the bayesian paradigm and this property renders our assumption practical.

The detection of vehicles takes place in two stages. The first stage is basically composed of three portions:

1. **Clustering and segmentation**, to group a set of points indicating objects, using a distance criterion. Next, segments are obtained using classical split and merge techniques.
2. **Vehicle hypotheses construction** using bayesian programming. The construction of hypotheses by bayesian programming results in a mechanism similar to that of hough transform. Peak values in the histogram indicates the most probable vehicle poses.

With real data, the first stage produces too many false positives. A second stage of filtering is applied to each vehicle hypothesis obtained after the first stage as a form of validation gating in order to reduce the number of false positives. It is broken down into two portions:

1. **Edge Filtering** is applied to extract the set of line segments that is only relevant to the vehicle hypothesis in question.
2. **Vehicle Support Filtering** is based on our proposed metric, vehicle support, that measures how much of the two adjacent sides of a vehicle are seen. We try to remove as many false positives as possible using the vehicle support.

3.1 Construction of Vehicle Hypotheses

A bayesian program is used to infer vehicle positions. The formulation of our bayesian program results in a mechanism similar to that of a hough transform. We can infer on positions and orientations of vehicles from segments detected from laser scan data which is analogous to the way line segments are recovered from an ensemble of points. However our histogram cells are updated in terms of probability which takes into account the length and the width of vehicles instead of voting in the case of hough transform.

Briefly, a bayesian program is composed of:

- the list of *relavant variables*;
- a *decomposition* of the joint distribution over these variables;
- the *parametrical form* of each factor of the decomposition;
- the *identification* of the parameters of these parametrical forms;
- a *question* in the form of probability distribution inferred from the joint distribution.

In the construction of the histogram, each line segment is treated independently. In doing so, it will be sufficient to simply go through the list of segments and add necessary information into the histogram for each line segment. This is achieved by performing data fusion with diagnostic [7]. Inference of vehicle poses is represented by the bayesian program in fig. 2 and fig. 3. In this paper, the following variables are adopted:

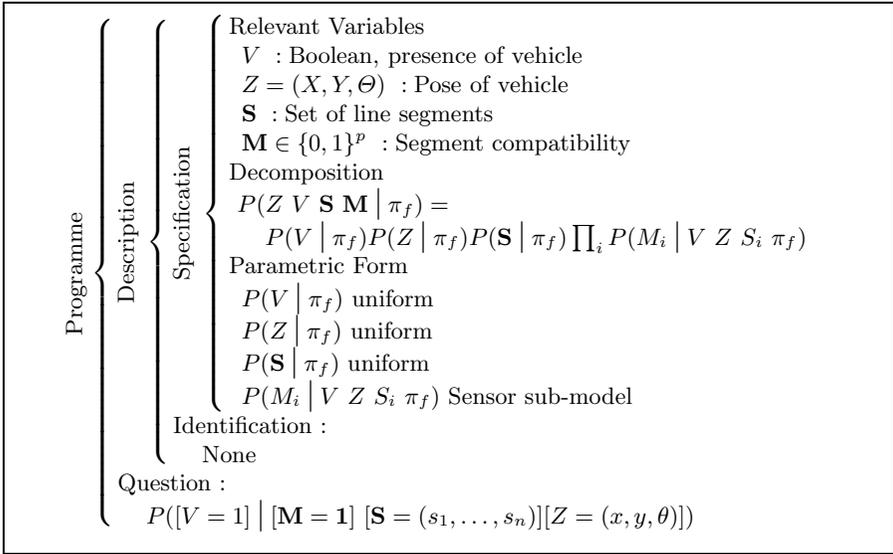
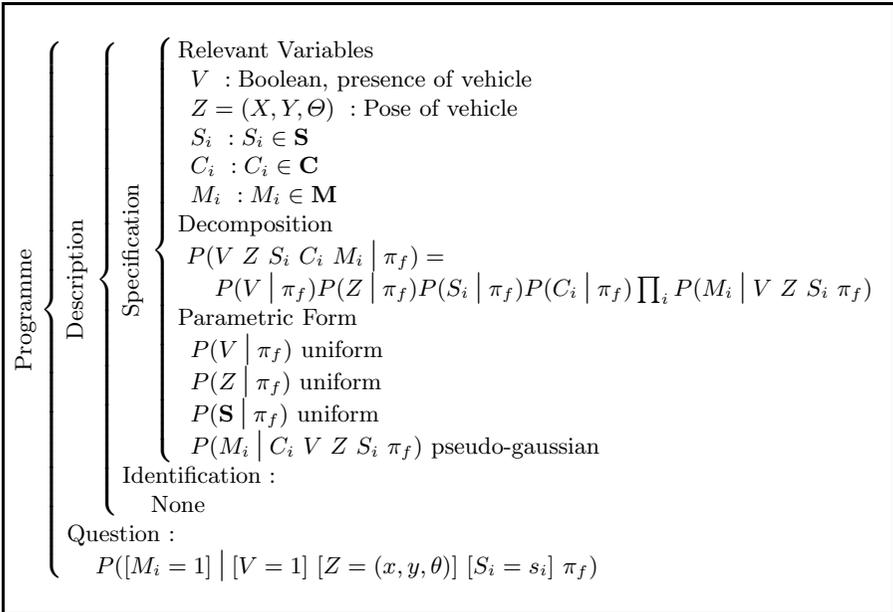
- V : A boolean value indicating the presence of a vehicle
- $Z = (x, y, \theta)$: The pose of a vehicle
- \mathbf{S} : Ensemble of extracted line segments
- $\mathbf{M} \in \{0, 1\}^p$: Compatibility of segments with vehicle pose
- $\mathbf{C} \in \{1, 2\}^p$: A value of 1 or 2 if segment corresponds to the width and length of a vehicle respectively
- π : A priori knowledge

To represent the absence of specific knowledge on the presence of vehicle $P(V | \pi_f)$, the pose of the vehicle $P(Z | \pi_f)$ and the segments $P(\mathbf{S} | \pi_f)$, they are represented as a uniform distribution.

The semantic of the question from the bayesian program (fig. 2) is to find the probability of a vehicle given all segments and the vehicle pose. This question can be simplified using baye's rule:

$$\begin{aligned} & P([V = 1] | [\mathbf{M} = \mathbf{1}] [\mathbf{S}] Z \pi_f) \\ &= K \prod_i P([M_i = 1] | [V = 1] Z S_i \pi_f) \end{aligned}$$

With K constant. The simplification of the question gives the product of the probability of the contributions of each line segment, which is given by each sensor sub-model. The local maximas of the function, $P([V = 1] | [\mathbf{M} = \mathbf{1}] [\mathbf{S} =$


Fig. 2: Detection of vehicles bayesian program

Fig. 3: Sensor Sub-Model bayesian program

$(s_1, s_2, \dots, s_n) [Z = (x, y, \theta)] \pi_f$, gives the possible vehicles hypotheses. For ease of calculation, the logarithm is applied. The sub-model used in the

calculation of $P(M_i | V Z S_i \pi_f)$ is described in fig. 3. The question of the sub-model can be further resolved:

$$\begin{aligned} & P([M_i = 1] | [V = 1] Z S_i \pi_f) \\ &= \sum_{C_i} P(C_i | \pi_f) P([M_i = 1] | C_i [V = 1] Z S_i \pi_f) \end{aligned}$$

The remaining problems lies in expressing $P(C_i | \pi_f)$, the probability that a segment corresponds to the length or width of a vehicle, and $P([M_i = 1] | C_i [V = 1] [Z = (x, y, \theta)] [S_i = s_i] \pi_f)$, the probability that a segment and its given association to the sides of the vehicles, corresponds to a vehicle at pose (x, y, θ) . For $P(C_i | \pi_f)$, a simple model is given by:

- $P([C_i = 1] | \pi_f) = \frac{L}{l+L}$
- $P([C_i = 2] | \pi_f) = \frac{l}{l+L}$

Given the length L and the width l of the assumed vehicle size, we consider the prior probability that the segment belongs to either the length or width is based on the simple ratio of the side in question against the sum of the two sides.

$P([M_i = 1] | C_i [V = 1] [Z = (x, y, \theta)] [S_i = s_i] \pi_f)$ is expressed by a pseudo-gaussian function $G(S_c, s_i)$. Where S_c represents the segment of a side of the hypothesized vehicle position (x, y, θ) . The more s_i is further or have a different orientation with respect to S_c , the smaller the value of $G(S_c, s_i)$. In practice, as most of the values in the histogram are negligible if the histogram is to be filtered by a threshold, it is sufficient to go through the list of segments and filling in the histogram values only for the possible vehicle poses (x, y, θ) that are compatible with the line segments. The vehicle poses can be easily calculated with simple geometry.

3.2 Edge Filtering

To calculate how well the sides conform to a vehicle hypothesis configuration, the set of relevant segments around the contours of the hypothesized vehicles have to be selected. The filtered edges are to provide data for the calculation of the vehicle support (section 3.3).

Two bounding rectangles are calculated from the vehicle hypothesis configuration with one rectangle a ratio smaller than the original vehicle size and the other a ratio bigger as illustrated in figure 4. The two bounding rectangles will then be oriented and positioned in the same manner as the hypothesized vehicle configuration.

The algorithm begins with the segment that contains the end point nearest to the origin (where Sick is). Starting from this segment, the algorithm starts to grow outward by searching for any segments where any of its endpoints lies sufficiently close to one of the endpoints in the original segment. This continues till a sufficiently close segment cannot be found.

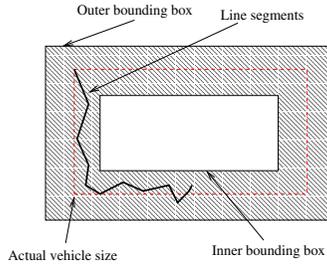


Fig. 4: Edge filtering with 2 bounding boxes. Shaded area indicates valid area

3.3 Vehicle Support Calculation

In adopting a conservative approach, both length and width of the vehicle must be adequately observed. Cases where either its length or width is verified are rejected as it introduces ambiguities and potentially false positives. A metric, the vehicle support, based on the sum of the magnitude of cross products can be used to enforce our conservative approach.

The calculation of vehicle support is given by eqn. 1. Under the ideal case where there are only two segments perfectly aligned to the edges of the vehicle, the result is a multiplication of the length and width of the vehicle. The equation for calculating the support is given by:

$$support = \forall i, j \sum_{i \neq j} | S_i \times S_j | \quad (1)$$

Intuitively, if a large portion of the length and a small portion of the width is detected, the support will give a low value and vice versa. Hence it enforces the verification of both the length and the width of the hypothesized vehicle before classifying it as a positive vehicle hypothesis. The enforcement of such rules are all conveniently embodied in a single equation (eqn 1).

4 Map Construction

The SLAM algorithm is independent of the construction of the vehicle hypotheses. Hence, a variety of SLAM methods can be used. If multiple hypotheses SLAM algorithms are used, the different hypotheses might come up with conflicting hypotheses caused by association of landmarks. Such conflicting associations of landmarks must be resolved in order to obtain a final consistent map. In such cases, a single observation might be associated with different landmarks across the various SLAM hypotheses. The most likely group of hypotheses with the same data association for a given observation can be combined together to obtain the landmark to be represented in the

final map while the rest of the hypotheses are ignored. A detailed description and example based on FastSLAM [5] [6], which is used in this paper, is presented in [9].

The bayesian programs used previously to construct the vehicle pose hypotheses gives a set of possible vehicle poses by taking each line segment extracted from the laser scan independently. It does not consider the previously made vehicle pose hypotheses. The disadvantage of the vehicle pose hypotheses construction and conservative gating approach is its inability to handle occlusion and false positives are obtained as a result. For example in fig. 5, a wrong hypothesis is constructed due to occlusion. A corner of a building is mistook as a vehicle pose hypothesis. By making more observations as CyCab navigates around to obtain a more comprehensive picture of the environment and taking into account previous hypotheses, wrongly made vehicle hypotheses can be removed.

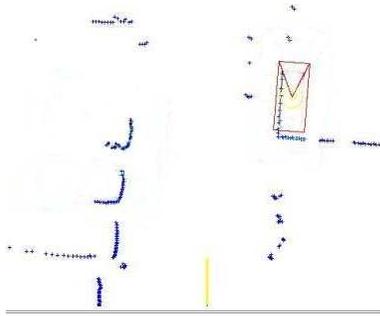


Fig. 5: An example of a wrong hypothesis due to occlusion

All hope is not lost for vehicle hypotheses that failed gating, often due to occlusion. Hypotheses that failed gating can be checked against previously accepted hypotheses. A current hypothesis that is very similar in position and angle with a previously accepted hypotheses can be taken to be a positive hypothesis by virtue of being previously accepted.

4.1 Considering Previous Hypotheses

The two main criteria for measuring similarity of a current hypothesis and a previously accepted hypothesis are their position and angle. It will be convenient to obtain a similarity measurement function that returns a bounded value of between 0 and 1. The measurement of difference in position is given by function f :

$$f = \frac{2 \times \text{Area}(P \cap Q)}{\text{Area}(P) + \text{Area}(Q)} \quad (2)$$

Where P and Q are both the geometry of the two vehicle hypotheses. A perfect fit of P and Q gives a value of 1 and a non overlapping P and Q gives a value of 0. Similarly, a measurement of difference in angle, g :

$$g = 1 - |\sin \theta| \quad (3)$$

The two hypotheses are considered similar as long as $\min(f, g) > \text{threshold}$.

5 Experimental Results

5.1 Vehicle Detection

After stage 1 (without gating), the vehicle hypothesis to the left in figure 6 is accepted even when only either the length of the vehicle is detected. But in fact, that detected side corresponds to a wall. This example demonstrates the weakness of using a laser scanner as ambiguous situations renders the system incapable of inferring correctly if it is vehicle. Additional information from a camera would be more useful. Such examples are the main motivation for a gating approach.

Stage 2 (gating) rejects the wrong hypothesis as represented by its inner and outer bounding boxes in fig. 7. Due to the conservative approach in validating hypotheses in stage 2, some potential hypotheses are inevitably eliminated in the process (eliminated vehicle hypothesis to the right in fig. 7).

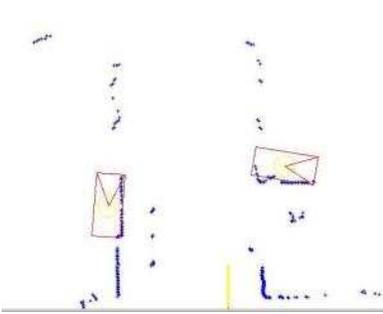


Fig. 6. Accepted vehicle hypothesis after stage 1 of vehicle detection

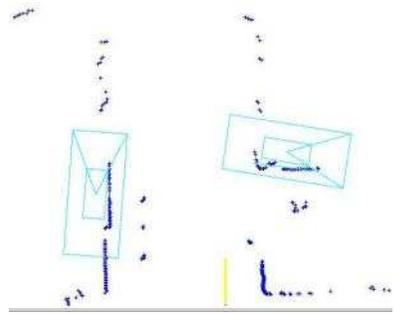


Fig. 7. The same configuration as in figure 6 but with rejected vehicle hypothesis in their inner and outer bounding boxes after stage 2 of vehicle detection

Vehicles parked in the car park are often spaced out sufficiently enough to be able to view its sides from the point of view of the laser scan. However, in the less likely event of being spaced very close together such that the relevant

laser impact points cannot be obtained, the relevant hypotheses cannot be obtained. This is one of the limitations of using only a laser scanner. A fusion of data with a camera will be desirable.

5.2 Map Construction

Tests were conducted within the context of the car park in INRIA Rhône-Alpes. The current implementation is a naive and unoptimized version of FastSLAM that executes with a frequency of between 4-6Hz on a pentium 4.

Figure 8 presents the multiple hypotheses of the position of CyCab and the other vehicles in the environment. The various CyCab hypotheses are followed by a curve indicating its mean path taken and the various landmark hypotheses.

Figures 9 and 10 illustrates final constructed map. It is basically a combined map of all the various hypotheses obtained from FastSLAM. In addition, the figure is overlaid with the laser impact points from the first laser scan and the laser scan at CyCab's current position.

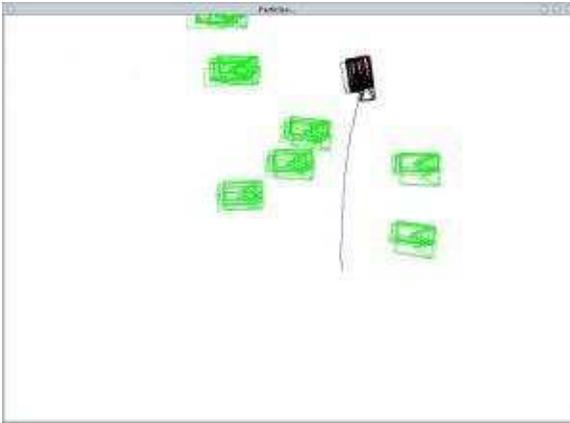


Fig. 8: The various CyCab and landmark hypotheses

6 Conclusion and Further Work

In this paper, the extraction of hypothesized vehicle poses from laser scan data only is presented. Due to limitations in the laser scan data, gating is required to remove the false positives. However the conservative vehicle hypotheses construction approach is at the cost of eliminating potential vehicle hypotheses. Vehicle hypotheses construction can be ameliorated by taking into account previously accepted vehicle hypotheses.

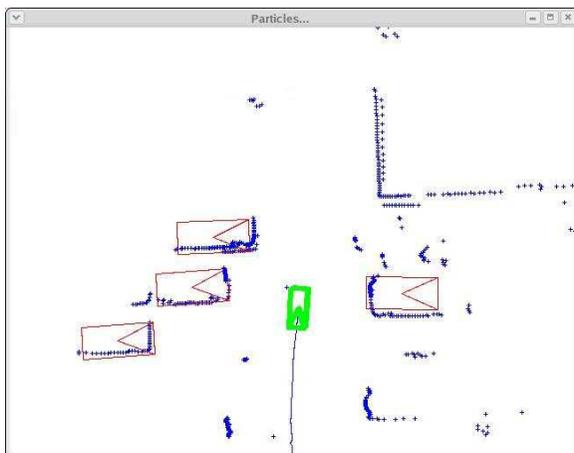


Fig. 9: Final map obtained by fusion of different hypotheses

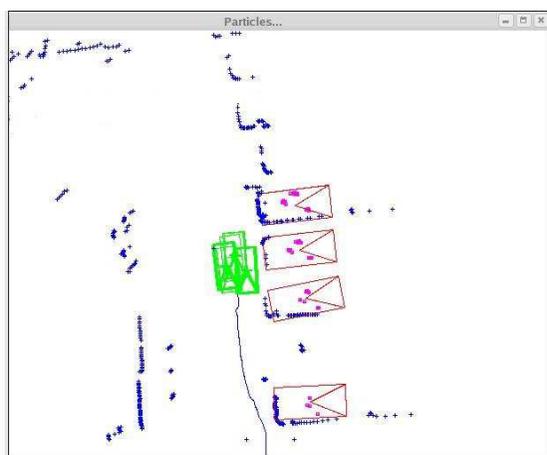


Fig. 10: Another mapping example

A more reliable method of vehicle hypotheses construction will be explored by fusing laser scan information with information from a camera. The current implementation assumes a static environment where landmarks do not move. Extensions will be made to handle moving objects.

7 Acknowledgements

This work is made possible by a study grant from the French Embassy of Singapore.

References

1. C.Pradalier, J.Hermosillo, C.Koike, C.Braillon, P.Bessiere, and C.Laugier. An autonomous car-like robot navigating safely among pedestrians. In *International Conference in Robotics and Automation*, 2004.
2. O. Lebeltel. *Programmation Bayienne des Robots*. PhD thesis, Institut National Polytechnique de Grenoble, France, October 1999.
3. Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun. Using EM to learn 3D models with mobile robots. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2001.
4. C. Martin and S. Thrun. Online acquisition of compact volumetric maps with mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, 2002. ICRA.
5. M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.
6. M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003. IJCAI.
7. C. Pradalier and F. Colas. Expressing bayesian fusion as a product of distributions: Applications in robotics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, October 2003.
8. Chatila R. and Laumond J.-P. Position referencing and consistent world modeling for mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 138–145, 1985.
9. C. Tay, C. Pradalier, and C. Laugier. Vehicle detection and car park mapping using laser scanner. 2005.

Wavelet Occupancy Grids: A Method for Compact Map Building*

Manuel Yguel, Olivier Aycard, and Christian Laugier

¹ ϵ -motion, GRAVIR-UJF-INRIA-INP Grenoble, France
firstname.lastname@inrialpes.fr

² Inria Rhône-Alpes, 655 avenue de l'Europe - Montbonnot 38334 Saint Ismier
Cedex, France

1 Introduction

The capacity to know about the environment is a major requirement for robots and automated systems. Real scenes are complex to perceive, dynamic and large. Therefore to perceive those scenes, robots have access to complementary sensors such as sonar/laser range-finders, cameras or bumpers, but all these are always noisy. In automated navigation, the internal representation of the environment is used for all fundamental tasks: localization[14], path-planning[6], obstacle avoidance[1] or target tracking[2]. But this modelling can require huge memory space, especially when the level of abstraction is fairly low and the representation is close to the sensor data. In these cases the time of data analysis is also prohibitive, making it necessary to change sophisticated algorithms for naive ones. This paper addresses the problem of data representation and data storage for large maps, under the constraints of multi-sensor real-time updates and hierarchical representation. Thus multi-scale algorithms could be applied to very large maps with efficient calculating time.

The first step when mapping the environment is obviously to define how to map. Geometrical properties are powerful guidelines to do it. A focus on the ability to reach a point from an other leads to the class of topological maps and the associated representation is the graph of connected places. Topological maps are semantically rich, can be drawn even without an accurate localisation of the robot and their size is very small. However it is difficult to build and update them automatically and they do not allow direct complex data manipulations. The possibility to index all the spatial information in a common reference frame, via a global coordinate system, leads to the class of metric maps. In an inversely symmetrical manner, metric maps are easy

* Work supported by ProBayes and the French ANRT (National Association for Technical Research).

to build and update and they allow direct complex interactions with the environment such as obstacle avoidance, but they are semantically poor, they require an accurate localisation of the robot and the size of data is critically huge. Among metrical maps, occupancy grids (OGs) are a classical approach [9] with the essential property, that they make it straightforward to integrate noisy measurements of different sensors over time. To deal with the sensor uncertainty is a main requirement for a mapping framework, because chip sensors are very noisy. And an unified theory for sensor integration allowed sensor networks, that achieve robustness for a robotic system. Convincing new world representations have recently used topological and metric maps in intricate manners to combine their advantages [13], [5]. Now, both of these new world representations resort to OGs for precise mapping. Therefore focusing on the main drawback of OGs which is the data size is a major challenge for robotic.

The wavelet framework [3], [8] is very interesting for non linear signal approximation as images. Image and occupancy grid processing are two related subjects as pointed out by Elfes in [4]. Moreover wavelets offer a rigorous mathematical framework for multi-resolution representations and signal compression. Thus unlike mixed representations as metric-topologic ones, we propose a new unified framework for multi-resolution map building based on wavelets, which we call the wavelet occupancy grid (WavOG). Pai and Reissell [10] have shown that wavelets could be used to represent 3D static maps, and also how to devise efficient path planning through rough terrain with this kind of representation. Drawing upon this, Sinopoli *et al.* [12] have used this static wavelets representation for global path-planning while using traditional 3D updatable OG for local navigation. What we propose is a synthesis of wavelet representation and OG representation in order to do OG building in the wavelet space. And also, we describe a probabilistic formulation for every scaled information provided by the wavelet transform.

In the rest of this paper we first present the wavelet mathematical framework in simple terms. Second, we explain how to link the theoretical building of OGs with wavelets in order to on-line construct occupancy grids in wavelet space. We will then pay attention to the multi-resolution occupancy meaning of WavOGs. Fourth, we describe how WavOGs are implemented and a compact representation is obtained. This new building was validated through a mapping tour with the cycab autonomous robot. Finally we present these results in a comparison with a standard occupancy grid, which proves the compactness of the new representation.

2 The Wavelet Mathematical Framework

The function which is discretized by a 2D-occupancy grid is the continuous occupancy function:

$$\begin{aligned} \mathbb{R}^2 &\longrightarrow [0; 1] \\ (x, y) &\longmapsto p(x, y, \textit{occupied}) \end{aligned}$$

which is the probability that an obstacle lies in the point (x, y) (Fig. 1).

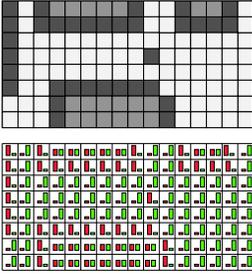


Fig. 1. Occupancy grid (in grey level) and its corresponding probability distribution. The green boxes represents the probability that the cell is empty while the red ones the probability that the cell is occupied.

The aim of this work is to compress the occupancy function. The main idea at work in several compression schemes is to project the function onto a set of elementary functions which is a basis for the *vector space* of approximation functions. For example the Fourier transformation projects the functions onto the infinite set of sines and cosines. Then, in this case, the approximation process consists in selecting a finite set of the lowest frequencies and rejects high ones, which are almost considered as noise. But this leads to poor compression results, especially for non linear functions as OGs certainly are. Indeed, There exists a similarity between OGs and images Fig. 1 and there exist some approximation spaces that are useful for this kind of signals called wavelet spaces [3].

We will now present a 1-dimensional wavelet decomposition using Mallat's algorithm [7]. Then we will present the wavelet notation and the Haar basis we use.

2.1 Example of the Haar Wavelet Transform in 1D

We will now focus on 1-dimensional wavelet decomposition using Haar basis.

A 1D function p , which is regularly discretized over n points $\{x_0, \dots, x_n\}$, is seen as a vector $[p(x_0), \dots, p(x_n)]$ (Fig. 2).

One elementary step of the Haar wavelet transform (HWT) uses two neighboring samples $p(x_i)$ and $p(x_{i+1})$ to generate a scale s_i coefficient and a detail coefficient d_i :

Table 1. elementary step of direct and inverse Haar transform

| Haar wavelet transform: | Haar inverse wavelet transform : |
|--|--|
| $s_i = (p(x_{2i}) + p(x_{2i+1})) / \sqrt{2}$ | $p(x_{2i}) = (s_i + d_i) / \sqrt{2}$ |
| $d_i = (p(x_{2i}) - p(x_{2i+1})) / \sqrt{2}$ | $p(x_{2i+1}) = (s_i - d_i) / \sqrt{2}$ |

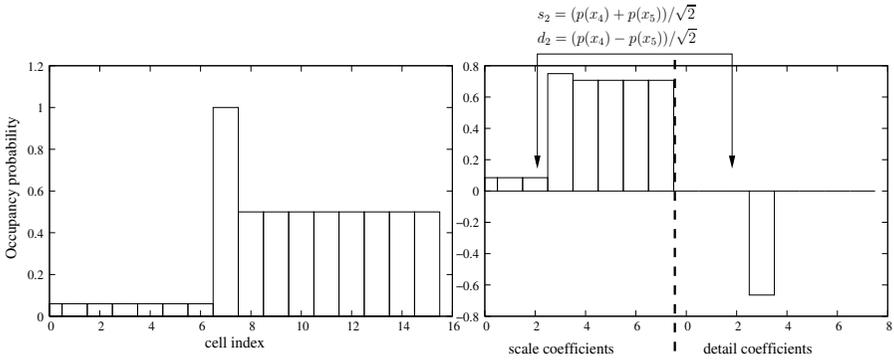


Fig. 2. left : mono-dimensional OG: an observer in the first cell views an obstacle in the 7th cell. Righth : first iteration of a 1D Haar transform; scaled and detail coefficients are grouped together. There is only one nonzero detail coefficient corresponding to the impacted cell.

Except for a constant, the scale coefficient is the average of the neighboring samples and the detail coefficient is the difference between them. It is clear that it exploits correlation for better encoding, because the more similar $p(x_{2i})$ and $p(x_{2i+1})$ are, the closer to zero d_i is.

The principle of the HWT is to apply the HWT elementary step recursively on the scale coefficients (Tab. 1, Fig. 2). The process can be stopped at every iteration but the best encoding is obtained when there is only one scale coefficient.

2.2 The Haar Wavelet Transform in 2D

It is straightforward to deduce HWT in 2 dimensions by alternating one step of the 1d wavelet transform on rows (Fig. 3(b)) and then on columns (Fig. 3(c)).

One step of the 2D HWT divides the space in 4 parts: the scale, raw detail, column detail and diagonal detail spaces (Fig. 3(c)). Then one square of 2×2 pixels in the original image produces 4 coefficients by one step of the 2D HWT (Fig. 4). These coefficients are the weights of special functions, called the Haar basis functions, in the Haar wavelet representation (Fig. 5). Performing HWT is just iterating this step on the previous scale space until the size of the resulting scale space is one.

Wavelet notation

Here we consider the 2D Haar basis. It is built upon 2 types of basis functions: the scale and detail functions.

- The scaling mother function is of unit value over the unit square:

$$\Phi(x, y) = 1 \text{ for } (x, y) \in [0, 1]^2, \text{ zero elsewhere}$$

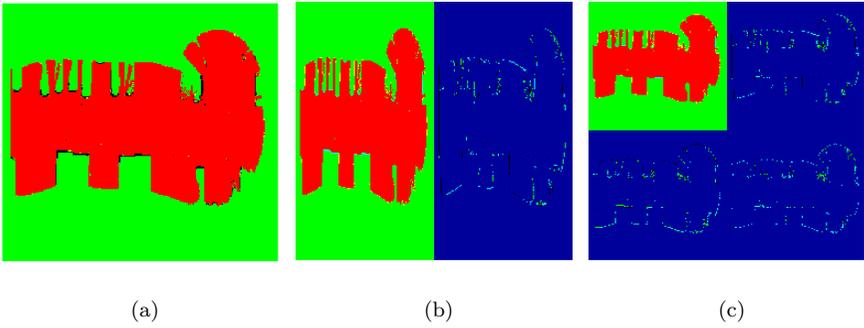


Fig. 3. One step of the Haar wavelet transform in 2d. From the original image (a): first (a): one 1d wavelet transform on each row (b), then one 1d wavelet transform on each column (c). In the result: from up to bottom, left to right: there are the scale, raw detail, column detail and diagonal detail spaces.

$$\begin{array}{c}
 \begin{array}{|c|c|} \hline u & v \\ \hline w & x \\ \hline \end{array}
 \xrightarrow{HWT2D}
 \begin{array}{|c|c|} \hline s = & r = \\ \frac{u+v+w+x}{2} & \frac{u-v+w-x}{2} \\ \hline \end{array}
 \begin{array}{|c|c|} \hline c = & d = \\ \frac{u+v-w-x}{2} & \frac{u-v-w+x}{2} \\ \hline \end{array}
 \xrightarrow{HWT2D^{-1}}
 \begin{array}{|c|c|} \hline u = \frac{s+r+c+d}{2} & v = \frac{s-r+c-d}{2} \\ \hline w = \frac{s+r-c-d}{2} & x = \frac{s-r-c+d}{2} \\ \hline \end{array}
 \end{array}$$

Fig. 4. 2D Haar wavelet transform and inverse transform: elementary step algorithm. The direct and inverse transform are the same algorithm.

- The three wavelet mother functions over the unit square³:

$$\Psi_{01}(x, y) = \begin{array}{|c|c|} \hline + & - \\ \hline \end{array}
 \quad
 \Psi_{10}(x, y) = \begin{array}{|c|c|} \hline + & \\ \hline - & \end{array}
 \quad
 \Psi_{11}(x, y) = \begin{array}{|c|c|} \hline + & - \\ \hline - & + \\ \hline \end{array}$$

Fig. 5. These graphically-defined functions are +1 where white and -1 where black in the unit square shown and implicitly zero outside that domain.

We define the Haar basis at scale s as the union of the set of scaled functions: $\{\Phi^{sij} | (i, j) \in \mathbb{Z}^2\}$ and the set of details functions: $\{\Psi_M^{lij} | l = 0, \dots, s; (i, j) \in \mathbb{Z}^2\}$. Where :

$$\Phi^{lij} = 2^{-l}\Phi(2^{-l}x - i, 2^{-l}y - j) \tag{1}$$

$$\Psi_M^{lij} = 2^{-l}\Phi(2^{-l}x - i, 2^{-l}y - j) \tag{2}$$

³ The type M take three values -01, 10 or 11- corresponding to one of the three mother wavelets for horizontal, vertical and diagonal differencing.

Each triplet (s, i, j) defines a *wavelet square* at scale s and offset (i, j) . The scale s is just the number of steps in the HWT. Squares at scale s are disjoint and have an area of $4^l u$ where u is the area of a square at scale 0. We use the term *coarser* to describe basis functions in square at higher scale and *finer* for those at lower scale.

Thanks to the orthogonality of the Haar basis, the weight of a basis vector function, e.g. Φ^{sij} , is formally given by the scalar product (noted $\langle . | . \rangle$) with the occupancy function:

$$\langle p(x, y) | \Phi^{sij} \rangle = \int_{x, y \in \mathbb{R}^2} p(x, y) \Phi^{sij}(x, y) dx dy \quad (3)$$

3 Occupancy Grids and Wavelets

We first present the bayesian definition of occupancy grids. As the purpose is to build dynamic maps, we show how occupancy grids are updated. Then as the wavelet framework is a vector space framework with special constraints, we show how to combine OGs and wavelets to perform updates in wavelet space.

We will now introduce OGs as a statistical representation of the distribution of obstacles over space (Fig.1). Each cell (x) corresponds to a binary random variable (E_x) whose value is either *occupied* (occ) or *empty* (emp)⁴. Each cell is considered independent from all others, which makes it possible to define the consequences of an observation for each specific cell. This leads to efficient update computation, linear in the number of cells in the sensor range.

3.1 Relation Between Sensor Measurement and Cell Occupancy

To build a robust modelisation, we use a statistical framework to capture measurement uncertainty. Let Z be a random variable which takes values among all possible sensor values. Let $P(Z|E_x)$ be the probability distribution over Z knowing the occupancy state of cell x ; $P(Z|E_x)$ is called the sensor model (Fig. 6(a) for E_x “occupied” and Fig. 6(b) for E_x “empty”).

We define the joint distribution over Z and E_x as $P(Z, E_x) = P(E_x)P(Z|E_x)$. For a new sensor measure z_t , the Bayes rule gives:

$$p(e_x|z_t) = \frac{p(e_x)p(z_t|e_x)}{p(z_t)} \quad (4)$$

⁴ We use capital letters for random variables and normal case letters for their realisation. We use the P capital letter for probability distributions and the p normal case letter for probability values.

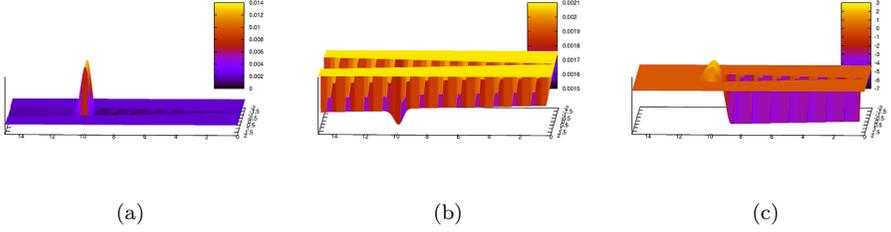


Fig. 6. Sensor models with a laser-range finder located in $(0, 0)$ the sensor measures an impact in $(0, 10)$. (a) Sensor model for occupied cells, (b) Sensor model for empty cells, (c) Log ratio of previous sensor models.

where the measurement probability is equal to a marginalisation term:

$$p(z_t) = p(occ)p(z_t|occ) + p(emp)p(z_t|emp) \quad (5)$$

So from an occupancy state $P(E_x)$ at time $t : p_t(e_x)$, a new observation gives a new occupancy state $P(E_x|[Z = z_t])$. This process defines a bayesian filter which can be indexed by time:

$$p_{t+1}(e_x) = p(e_x|z_t) = \frac{p_t(e_x)p(z_t|e_x)}{\sum_{e_x} p_t(e_x)p(z_t|e_x)} \quad (6)$$

In a mapping process the robot gets a series of observations $\mathcal{Z} = \{z_0, z_1, \dots, z_t\}$. Therefore $p_t(e_x)$ represents $p(e_x|z_0, \dots, z_{t-1})$ in the above equation. The process is initialized with a prior for $P_{t=0}(E_x)$, it could be the map obtained by a previous mapping process or uniform distribution if the aera is unknown. Thus Eq. 6 provides a general framework for updating a map each cell apart.

3.2 Logarithmic-Form for Occupancy Grid Updates

We will now link OGs to wavelets. We have seen that we can project a huge representation of a function in a wavelet *vector* space in order to compress it. However an occupancy update (Eq. 6) is not a linear operation. Thus, we present now a well known logarithmic transformation which makes it possible to perform the entire update operation with only sums. Since $p_t(occ) = 1 - p_t(emp)$, we can summarize $p_t(occ)$ and $p_t(emp)$ with only one number q_t :

$$q_t = p_t(occ)/p_t(emp) \quad (7)$$

As can be seen⁵, $p_t(occ)$ and $p_t(emp)$ are easy to compute from q_t .

⁵ $p_t(occ) = \frac{q_t}{1+q_t}$ and $p_t(emp) = \frac{1}{1+q_t}$

This leads to the elimination of the marginalisation term in (6), so that the time process appears to be:

$$q_t = \frac{p(z_{t-1}|occ)}{p(z_{t-1}|emp)} q_{t-1} = q_0 \prod_{i=0}^{t-1} \frac{p(z_i|occ)}{p(z_i|emp)} \quad (8)$$

where q_{t-1} was recursively evaluated.

Now, the products can be changed into sums by using a logarithm. Let $odds_t = \log(q_t)$ then it leads to *log-form* expressions of the occupancy grid updates:

$$odds_t = \log\left(\frac{p(z_{t-1}|occ)}{p(z_{t-1}|emp)}\right) + odds_{t-1} = odds_0 + \sum_{i=0}^{t-1} \log\left(\frac{p(z_i|occ)}{p(z_i|emp)}\right) \quad (9)$$

In (9) $\log\left(\frac{p(z_{t-1}|occ)}{p(z_{t-1}|emp)}\right)$ is the observation term of the update; we note it $Obs(z_{t-1})$. $odds_{t-1}$ is the a priori term so that the observation term corrects it.

Thus, updating a WavOG comes down to adding the wavelet transform of observation terms to the wavelet representation of the map.

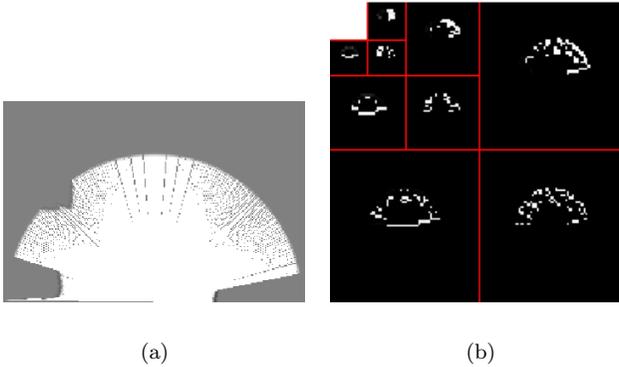


Fig. 7. mapping obtained by a single laser range-finder in OG and WavOG: (a) classical OG, (b) the corresponding 3 first detail spaces of the WavOG.

4 Multi-resolution

The wavelet representation is a hierarchical one, and WavOGs are represented from the coarser scales to the finer. So the coarse resolution information must be interpreted in order to use multi-scale algorithms.

Let us analyse the scaled coefficients of a WavOG. As we have seen in sec. 2, it is given by a scalar product eq. (3). It is the integral of a product between the log-form of the occupancy grid function and the scale basis function $\Phi_s^{(i,j)}$, where s indexes the scale.

$$\begin{aligned} < \sum_x \log\left(\frac{p(occ)}{p(emp)}(x)\right) \delta_x | \Phi_s^{(i,j)}(x) > = \sum_x \log\left(\frac{p(occ)}{p(emp)}(x)\right) \cdot \Phi_s^{(i,j)}(x) \\ = \log\left(\frac{\prod_x p([E_x = occ])^{\Phi_s^{(i,j)}(x)}}{\prod_x p([E_x = emp])^{\Phi_s^{(i,j)}(x)}}\right) = k \log(q^s) \end{aligned} \quad (10)$$

In the case of the Haar wavelet basis (Eq. 2) $\Phi_s^{(i,j)}$ is constant over a square S and zero elsewhere. Let k this constant⁶:

$$k \log(q^s) = k \log\left(\frac{\prod_{x \in S} p([E_x = occ])}{\prod_{x \in S} p([E_x = emp])}\right) \quad (11)$$

$$q^s = \frac{\prod_{x \in S} p([E_x = occ])}{\prod_{x \in S} p([E_x = emp])} \quad (12)$$

let us define an area of n cells: $\mathcal{A} = \{c_0, \dots, c_n\}$, then let us define 2 events: e^{full} when all the cells in \mathcal{A} are occupied and e^{open} when all the cells in \mathcal{A} are empty. So the weight for the scaled function leads us immediately to the probability of e^{full} or e^{open} when the system must make a choice between those two possibilities:

$$P(e^{open} | e^{open} \vee e^{full}) = \frac{\prod_{x \in \mathcal{A}} p([E_x = emp])}{\prod_{x \in \mathcal{A}} p([E_x = occ]) + \prod_{x \in \mathcal{A}} p([E_x = emp])} = \frac{1}{1 + q^s}$$

and in a symmetrical way, $P(e^{full} | e^{open} \vee e^{full})$ is obtained. So each resolution of the map provides a continuous information about the occupancy of the map.

The pixel (i, j, s) in the scale space of a WavOG is proportional to the mean over the square (i, j, s) ⁷ of the log-ratio occupancy values (Fig. 8).

5 Implementation

We have implemented a classical mapping process using a WavOG with the Cycab. The Cycab is an autonomous robot devoted to urban transport equipped with a laser range-finder : SICK LMS-291. In the mapping loop, there is first a localisation step using a SLAM algorithm and a second step where the map is updated from the current position using sensor values. We use the classical SLAM algorithm [11] to get the absolute position.

⁶ with value: 2^{-s} .

⁷ The square has its top left corner at coordinate: $(2^s i, 2^s j)$ and has a side of 2^s pixels.

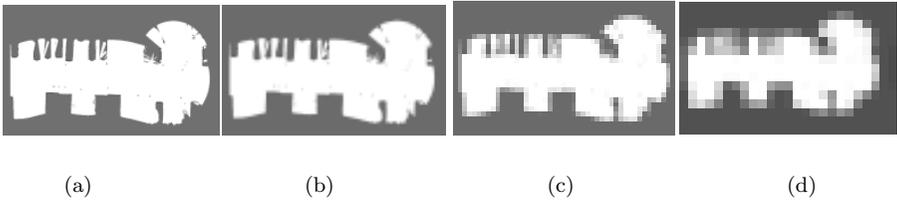


Fig. 8. Scaled view of OG provided in WavOG representation (a) scale $\frac{1}{16}$: 192×128 pixels, $s = 2$. (b) scale $\frac{1}{64}$: 96×64 pixels, $s = 3$. (c) scale $\frac{1}{256}$: 48×32 pixels, $s = 4$. (d) scale $\frac{1}{1024}$: 24×16 pixels, $s = 5$.

The laser range-finder has an 8-meter range. The side of the finer scale is 6.25cm then sizes double over 5 scales. Thus the side of the coarsest cells is 1 meter and so there still exist many coarser cells which appear totally open at this scale.

We consider a square window which encloses the sensor view. We maintain an area of this size as a buffer. As long as the laser impacts belong to this window, the updates occur in this buffer. Then we apply a 2D Haar wavelet transform to the buffer and add it to the WavOG. The buffer is flushed and the process is repeated. Waiting for an entire sick scan at least before doing the wavelet transform gives a large enough area to see regular fields appear. So that the result of the wavelet transform is sparse and it is only necessary to perform wavelet transforms at distant time intervals.

After experimental studies we set a compression threshold which is a compromise between data fitting and sparsity. *i.e.* after updating a wavelet detail coefficient, if it is lower than this threshold, it is removed from the WavOG.

6 Results

As a validation of our method we compare the number of cells in a classical occupancy grid with a WavOG (Fig. 9(a), Fig. 9(b)).

Fig. 9 shows the results obtained on the car park of INRIA. These experimental results clearly shows that we have obtained a significant reduction of the size of the model (about 80% relatively to the *OG* model), and that the interesting details are still represented (such as the beacons represented by dark dots in Fig. 9(c)). It should be noticed that the coarser model give a quite good representation of the empty space (see Fig. 9(d)); this model could be used for path planning, and refined when necessary using the wavelet model. In the previous experiments, the map building has been done in real-time. This kind of compression is however weaker than we expected, the reason seems to be due to the logarithm use. Indeed the very weak probabilities which must appear to be the same in human eye, are very different in logarithm space *e.g.* consider

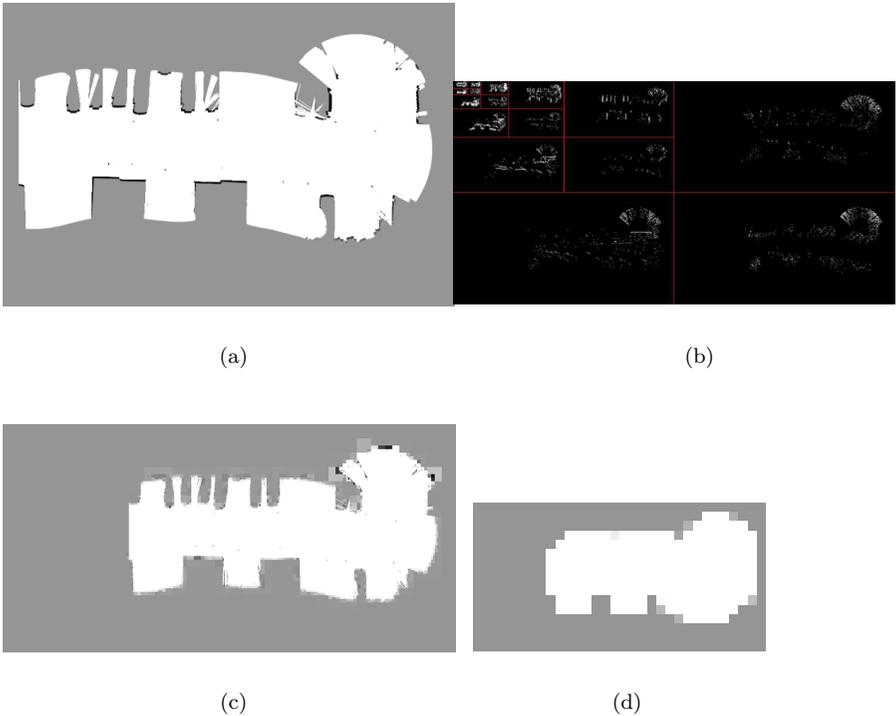


Fig. 9. (a) classical occupancy grid : 393,126 cells. (b) wavelet occupancy grid : 78,742 cells. (c) reconstructed occupancy grid from the WavOG. (d) Semantics for the coarser scaled of the WavOG.

the difference between 10^{-100} and 10^{-50} is huge in log space whereas it is negligible in a standard representation of probabilities. Then two conclusions arise: one it is necessary to work with bounded probabilities, which make the map reactive and easy to compress in wavelet space, two we must observe homogeneously near area such as the duration of observation of a cell don't make a bias in the map representation and then in the map compression. So in the development of this framework, we plan to study systematic methods to achieve both of these tasks.

7 Conclusion and Future Works

This paper introduces the structure of wavelet occupancy grids (WavOGs) as a tool for storing occupancy grids in a compact way. We have shown that WavOGs provide a continuous semantics of occupancy through scaled spaces. In accordance with the theoretical properties of wavelets, our experiments have validated that WavOGs allow major memory gains. WavOG as a com-

pact multi-scaled tool provides an efficient framework for the various algorithms that use OGs such as robot navigation, spatio-temporal classification or multiple target-tracking. In future works we plan to apply WavOGs to the monitoring of urban traffic over large areas.

Acknowledgment:

We would like to thank Cédric Pradalier in particular, for providing the experimental material of this paper, as well as an operational source code and robot.

References

1. J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1179–1187, - 1989.
2. C. Coué, T. Fraichard, P. Bessière, and E. Mazer. Using bayesian programming for multi-sensor multi-target tracking in automotive applications. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2003.
3. I. Daubechies. *Ten Lectures on Wavelets*. Number 61 in CBMS-NSF Series in Applied Mathematics. SIAM Publications, Philadelphia, 1992.
4. Alberto Elfes. Multi-source spatial data fusion using bayesian reasoning. In M. A. Abidi and R. C. Gonzalez, editors, *Data Fusion in Robotics and Machine Intelligence*, chapter 3, pages 137–163. Academic Press, 1992. ISBN 0-12-042120-8.
5. B. J. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, (119):191–233, 2000.
6. Maxim Likhachev, Geoffrey J. Gordon, and Sebastian Thrun. Ara*: Anytime a* with provable bounds on sub-optimality. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.
7. Stéphane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(7):674–693, 1989.
8. Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1998.
9. Hans P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, July/August 1988. ISSN:0738-4602.
10. D. K. Pai and L.-M. Reissell. Multiresolution rough terrain motion planning. In *IEEE Transactions on Robotics and Automation*, volume 1, pages 19–33, February 1998.
11. C. Pradalier and S. Sekhavat. Simultaneous localization and mapping using the geometric projection filter and correspondence graph matching. *Advanced Robotics*, 2004.
12. Bruno Sinopoli, Mario Micheli, Gianluca Donato, and T. John Koo. Vision based navigation for an unmanned aerial vehicle. In *Proceedings of the International Conference on Robotics and Automation*, May 2001.
13. S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 1(99):21–71, 1999.
14. S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998. joint issue with *Autonomous Robots* 5.

Further Results with Localization and Mapping Using Range from Radio

Joseph Djugash¹, Sanjiv Singh¹, and Peter Corke²

¹ Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA.
{robojoe@cmu.edu,ssingh@ri.cmu.edu}

² CSIRO ICT Centre, P.O. Box 883, Kenmore, Australia 4069.
{peter.corke@csiro.au}

Summary. In this paper, we present recent results with using range from radio for mobile robot localization. In previous work we have shown how range readings from radio tags placed in the environment can be used to localize a robot. We have extended previous work to consider robustness. Specifically, we are interested in the case where range readings are very noisy and available intermittently. Also, we consider the case where the location of the radio tags is not known at all ahead of time and must be solved for simultaneously along with the position of the moving robot. We present results from a mobile robot that is equipped with GPS for ground truth, operating over several km.

Keywords: SLAM, range-based localization, kalman filter, particle filter

1 Introduction

Many tasks for which robots are well suited require a high level of precision in localization for the application to be successful. One solution to the problem of localization in which environmental structure can't be relied upon for use in localization is to obtain absolute position via GPS. This approach is limited, however, to environments in which a clear line of sight to GPS satellites orbiting the earth, is available. Robots navigating inside buildings or underground cannot receive GPS data, and in outdoor environments nearby structures and even foliage can affect the quality of localization. Another common localization technique is dead reckoning, in which the robot's position is estimated based on measurements of distance travelled and orientation taken from wheel encoders and gyros. Since the dead reckoning position estimate degrades over time, a robot must correct position error using landmarks detected by on-board sensors. A problem that frequently arises in these cases is that of data association: sensed data must be associated with the correct landmark, even

though multiple landmarks may have similar features. Additionally, in many settings it is not possible to guarantee line of sight to the landmarks.

The method of sensing we have been using involves low-cost, low-power, radio frequency tags placed in the environment. Originally intended as a means to track assets and people in an environment equipped with special RF transponders, we invert the paradigm by fixing the tags in the environment and moving a transponder with a robot. As the robot moves, the transponder periodically sends out a query, and any tags within range respond by sending a reply. The robot can then estimate the distance to each responding tag by determining the time lapsed between sending the query and receiving the response. The advantage of such a method is that it does not require line of sight between tags and the mobile robot, making it useful in many environmental conditions that fail optical methods. Note that, since each tag transmits a unique ID number, distance readings are automatically associated with the appropriate tags, so the data association problem is solved trivially.

We would like to send a mobile robot into an environment containing these tags and have it navigate successfully while maintaining a reliable estimate of its location at all times. In this paper, we examine issues of robustness that result from noisy and infrequent range data. We also examine the issues of SLAM in this environment by presenting results from experiments in which the robot starts moving in the environment without apriori knowledge of the location of the tags.

2 Related Work

Most landmark-based localization systems use sensors that measure relative bearing or in some cases both range and bearing to distinct features in the environment. In the case that the location of these landmarks is unknown, the problem is more difficult and is generally known as Simultaneous Localization and Mapping (SLAM). Here we report on localization results with a modality in which only range to landmarks (RF tags) is measured. Some other researchers have used range to estimate position. In most cases, instead of using range, signal strength from a known transmitter is used to produce a “pseudorange” that is then used for triangulation. For instance, the Cricket System [9] uses fixed ultrasound emitters and embedded receivers in the object being located. Radio frequency signals are used to synchronize time measurements and to reject multipath readings. The localization technique is based on triangulation relative to the beacons. The RADAR system [1] uses 802.11b wireless networking for localization. This system uses the signal strength of each packet to localize a laptop. RADAR uses nearest neighbor heuristics to achieve localization accuracy of about 3 meters. The SpotOn system [4], uses radio signal attenuation to estimate distance between tags. The system localizes wireless devices relative to one another, rather than to fixed base stations, allowing for ad-hoc localization. Note that GPS also works by triangulating

ranges to multiple satellites. In some cases, GPS localization is augmented with inertial measurement and/or dead reckoning. In almost all such systems, GPS triangulation generally develops an estimate of position as well as uncertainty that is merged with a position estimate from dead reckoning. Other methods choose to train on patterns of signal strength to localize. For example, Ladd et al propose a Bayesian formulation to localize based on signal strength patterns from fixed receivers [8].

In contrast, we use a single filter to combine range measurements with dead reckoning and inertial measurements. While the range measurements are noisy and exhibit biases, we find that treatment by an extended Kalman filter (necessary because the underlying system is non-linear) after preprocessing to remove outliers and to remove systematic biases, suffices as long as the estimate doesn't get too far from the true state. This might happen if the initial condition is too far from the true state or if the filter diverges due to missing range data over an extended period of time. The Kalman filter has the advantage that the representation of the distribution is compact; a Gaussian distribution can be represented by a mean and a covariance matrix. The robot's pose estimation is maintained as a Gaussian distribution and sensor data from dead reckoning and landmark observations is fused to obtain a new position distribution.

Recent extensions of Kalman filtering allow for non-Gaussian, multimodal probability distributions through multiple hypothesis tracking. The result is a more versatile estimation technique that still preserves many of the computational advantages of the Kalman filter. Monte Carlo localization, or particle filtering, provides a method of representing multimodal distributions for position estimation [2], [11], with the advantage that the computational requirements can be scaled. The main advantage of these methods is their ability to converge from a poor initial condition. We show how a particle filter is able to recover from large offset errors that are large enough that the Kalman Filter fails. Also, we extend previous work [5], [7] in SLAM by treating the case in which the robot starts with no information about the location of the tags in the environment.

3 Approach

Our current emphasis is robustness. We would like to explicitly treat the case of noisy and missing range data in addition to requiring the robot to discover the location of the landmarks on its own. We assume only that the robot has some information about the accuracy of the range measurements as reported previously [7]. Here we use range data that is significantly less frequent and more noisy. For example, the range measurements can have a variance of upto 6 m and range measurements can be as spread out by as much as 15 seconds.

Below we discuss the use of particle filter as a method of being able to recover from large estimation errors. While the particle filter has weaker per-

formance than the Kalman filter when all is well, it shines when there is a significant break in the range data or when there is a large initial offset. We show the ability of the robot to locate the radio tags in the case that their locations are not known ahead of time.

4 Localization

4.1 Localization with Kalman Filter

Formulation

We have formulated a Kalman Filter that estimates position given measurements of odometry and heading change (from a gyro), and range measurements. Odometry and gyro measurements are used in the state propagation or the prediction step, while the range measurements are incorporated in the correction step.

Process Model. If the robot state at time k is $q_k = [x_k, y_k, \theta_k, \beta_k, \eta_k]^T$, where x_k, y_k, θ_k are the robot's position and orientation and β_k, η_k are the gyro output scale-factor error and bias at time k . The dynamics of the wheeled robot used in this experiment are well-modeled by the following set of non-linear equations:

$$q_{k+1} = \begin{bmatrix} x_k + \Delta D_k \cos(\theta_k) \\ y_k + \Delta D_k \sin(\theta_k) \\ \theta_k + (1 + \beta_k)\Delta\theta_k + \eta_k \\ \beta_k \\ \eta_k \end{bmatrix} + \nu_k = f(\hat{q}_k, u_k) + \nu_k, \quad (1)$$

where ν_k is a noise vector, ΔD_k is the odometric distance traveled, and $\Delta\theta_k$ is the orientation change. These dead reckoning measurements constitute the control input vector $u_k = [\Delta D_k, \Delta\theta_k]^T$. When a new control input vector $u(k) = [\Delta D_k, \Delta\theta_k]^T$ is received, the robot's state is updated according to the process model equation. Then we apply a standard Kalman filter, [7], to propagate the covariance matrix with the extension from our previous work to incorporate the gyro bias terms within the filter.

Measurement Model. The range measurement at time k is modeled by:

$$r_k = \sqrt{(x_k - x_b)^2 + (y_k - y_b)^2} \\ y(k) = \begin{bmatrix} r_k \\ \Delta D_k \\ \Delta\theta_k \end{bmatrix} + \omega(k) = h(q_{k+1}) + \omega(k) \quad (2)$$

where, r_k is the range measurement received at time k and (x_b, y_b) is the location of the beacon from which a measurement was received. When a measurement is obtained, using the measurement model, we compute the expected range r_k to the beacon. Then the state can be updated using standard Kalman filtering equations.

Results

In order to evaluate the performance of the filter we turn to the two commonly used error metrics, the Cross-Track Error (XTE) and the Along-Track Error (ATE). The XTE accounts for the position error that is orthogonal to the robot’s path (i.e. orthogonal to the true robot’s orientation), while the ATE accounts for the tangential component of the position error. As part of our error analysis of the path estimates, we observe the average of the absolute values of the XTE and ATE for each point in the path, as well as the maximum and standard deviation of these errors.

In the experiment illustrated here, the true initial robot position from GPS was used as the initial estimate. Furthermore, the location of each tag was known. Figure 1 shows the estimated path using the Kalman filter, along with the GPS ground truth (with 2 cm accuracy) for comparison.

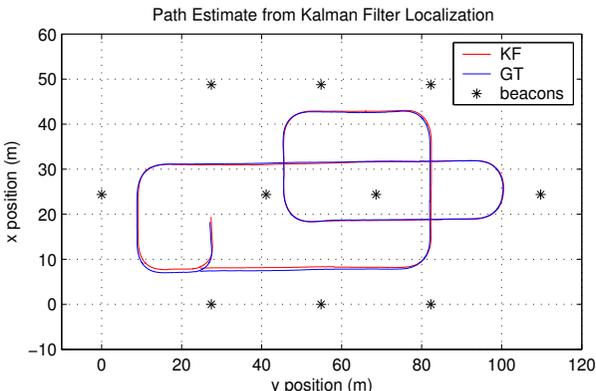


Fig. 1. The path estimate from localization (red), ground truth (blue) and beacon locations (*) are shown. The filter uses odometry and a gyro with range measurements from the RF beacons to localize itself. The path begins at (0,0) and ends at (33,0), travelling a total of 3.7 km and completing 11 identical loops, with the final loop (0.343 km) shown above. (Note the axes are flipped). Numerical results are given in Table 1.

Table 1. Cross-Track and Along-Track Errors for Kalman filter Localization estimate for the entire data set using the Kalman Filter with gyro bias compensation.

| | XTE | ATE |
|------------|----------|----------|
| Mean Abs. | 0.3439 m | 0.3309 m |
| Max. | 1.7634 m | 1.7350 m |
| Std. Dev.n | 0.3032 m | 0.2893 m |

Failure

Sensor Silence. An issue that requires attention while dealing with the Kalman filter is that of *extensive sensor silence*. When the system encounters a long period during which no range measurements are received from the beacons, it becomes heavily dependant on the odometry and its estimate diverges. Upon recovering from this period of sensor silence, the Kalman filter is misled into settling at a diverged solution. The Figure 2 shows the failure state of the Kalman filter when presented with a period of sensor silence. In this experiment, all range measurements received prior to a certain time were ignored so that the position estimate is derived through odometry alone. As can be seen in the figure, when the range data starts once again, the Kalman filter fails to converge to an accurate estimate.

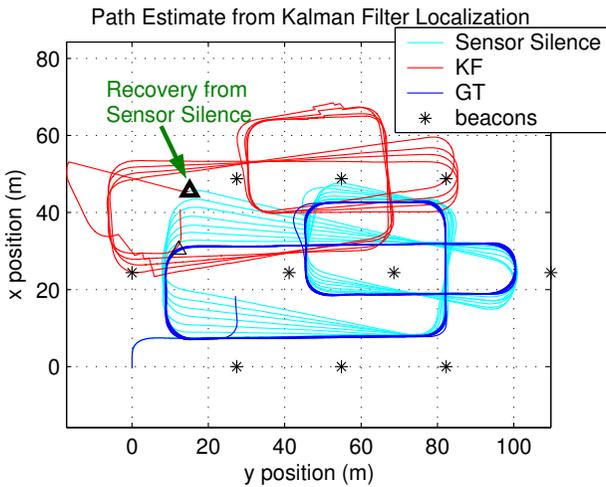


Fig. 2. The path estimate during the extended period of "simulated" sensor silence (cyan), Kalman filter's recovery from the diverged solution (red), ground truth (blue) and beacon locations are shown. (Note the axes are flipped). The filter is not able to properly recover from the diverged solution resultant of the initial period of sensor silence.

Although this is characteristic of all Kalman filters in general, this problem is especially critical while dealing with range-only sensors. Due to the extra level of ambiguity associated with each range measurement it becomes far easier for the estimate to converge at an incorrect solution.

4.2 Localization with Particle Filter

As we see above, the Kalman filter can fail when the assumptions of linearity can not be justified. In this case, it is useful to look at methods like Particle

Filters that can converge without an initial estimate. Particle Filters are a way of implementing multiple hypothesis tracking. Initially, the system starts with a uniform distribution of particles which then cluster based on measurements. As with the Kalman filter, we use the dead reckoning as a means of prediction (by drifting all particles by the amount measured by the odometry and gyro before a diffusion step to account for increased uncertainty). Correction comes from resampling based on probability associated with each particle. Position estimates are obtained from the centroid of the particle positions.

Formulation

The particle filter evaluated in this work estimates only position on the plane, not vehicle orientation. Each “particle” is a point in the state space (in this case the x, y plane) and represents a particular solution. The particle resampling method used is as described by Isard and Blake [3]. Drift is applied to all particles based on the displacement estimated by dead reckoning from the state at the previous measurement. Diffusion is achieved by applying a Gaussian distributed displacement with a standard deviation of B m/s which scales according to intersample interval. Given a range measurement r from the beacon at location $X_b = (x_b, y_b)$ the probability for the i 'th particle is

$$P(r, X_b, X_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r-|X_b-X_i|)^2}{2\sigma^2}} + P_0 \quad (3)$$

which has a maximum in a circle of radius r about the beacon with a radial cross-section that is Gaussian. The minimum probability, P_0 , helps reduce problems with particle extinction. σ is related to the variance in the received range measurements.

It was found to be important to gate range measurements through a normalized error and a range measurement band, [7]. In the event of a measurement outside the range gate an open-loop update is performed, the particles are displaced by the dead reckoning displacement without resampling or diffusion.

The location of the vehicle is taken as the probability weighted mean of all particles. There is no attempt made to cluster the particles so if there are, for example, two distinct particle clusters the mean would lie between them. Initially this estimate has a significantly different value to the vehicle's position but converges rapidly. Here we use 1000 particles, $\sigma = 0.37$, and $B = 0.03$.

Results

In the experiment illustrated here, the initial condition for the particles is based on **no** prior information, the particles are distributed uniformly over a large bounding rectangle that encloses all the beacons. The location of each

tag was known apriori. Figure 3(a) contains the plot of the particle filter estimated path, along with the GPS ground truth.

It should be noted that the particle filter is a stochastic estimation tool and results vary from run to run using the same data. However it is consistently reliable in estimating the vehicle's location with no prior information.

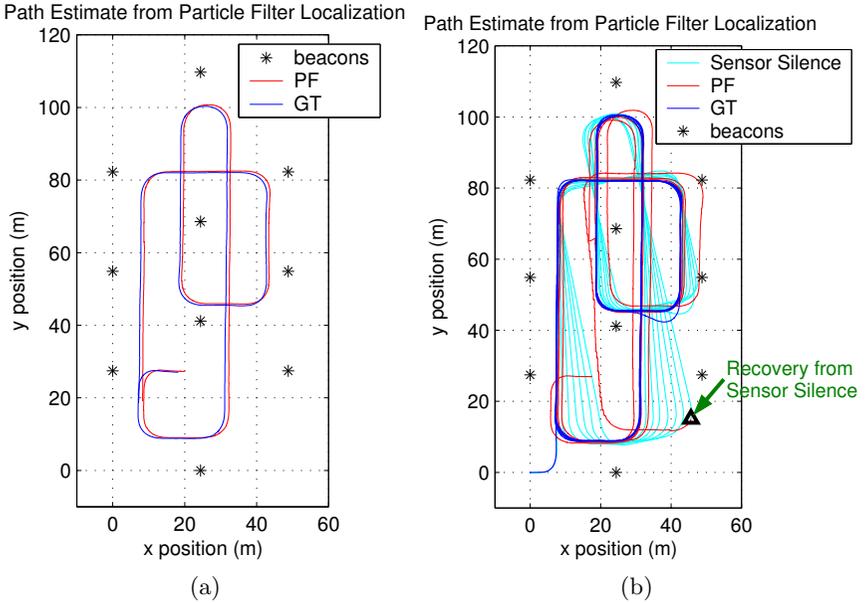


Fig. 3. (a) The path estimate from localization using a Particle Filter (red), ground truth (blue) and beacon locations (*) are shown. The filter uses the odometry and a gyro with absolute measurements from the RF beacons to produce this path estimate. The Particle Filter is not given any information regarding the initial location of the robot, hence it begins its estimate with a particle cloud uniformly distributed with a mean at (-3.6 m, -2.5 m). The final loop (0.343 km) of the data set is shown here, where the Particle Filter converges to a solution. Numerical results are given in Table 2. (b) The path estimate during the extended period of "simulated" sensor silence (cyan), Particle filter's recovery from the diverged solution (red), ground truth (blue) and beacon locations are shown. The filter easily recovers from the diverged solution, exhibiting the true nature of the particle filter.

The next experiment addresses the problem of extensive sensor silence discussed in Section 4.1. When the Particle filter is presented with the same scenario that was given to the Kalman filter earlier we acquire the Figure 3(b). This figure reveals the ability of the Particle filter to recover from an initially diverged estimate. It can be observed that although in most cases the particle filter produces a locally non-stable solution (due to resampling of the

Table 2. Cross-Track and Along-Track Errors for Particle filter Localization estimate for the entire data set.

| | XTE | ATE |
|-----------|----------|----------|
| Mean Abs. | 0.4053 m | 0.3623 m |
| Max. | 1.6178 m | 1.8096 m |
| Std. Dev. | 0.2936 m | 0.2908 m |

particles), its ability to recover from a diverged solution makes it an effective localization algorithm.

5 SLAM – Simultaneous Localization and Mapping

Here we deal with the case where the location of the radio tags is not known ahead of time. We consider an online (Kalman Filter) formulation that estimates the tag locations at the same time as estimating the robot position.

5.1 Formulation of Kalman Filter SLAM

The Kalman filter approach described in Section 4.1 can be reformulated for the SLAM problem.

Process Model: In order to extend the formulation from the localization case to perform SLAM, we need only to include position estimates of each beacon in the state vector. So,

$$q_k = [x_k \ y_k \ \theta_k \ x_{b1} \ y_{b1} \ \dots \ x_{bn} \ y_{bn}]^T \quad (4)$$

where n is the number of initialized RF beacons at time k . The process used to initialize the beacons is described later in this section.

Measurement Model: To perform SLAM with a range measurement from beacon b , located at (x_b, y_b) , we modify the Jacobian $H(k)$ (the measurement matrix) to include partials corresponding to each beacon within the current state vector. So,

$$H(k) = \frac{\partial h}{\partial q_k} \Big|_{q=\hat{q}} = \left[\frac{\partial h}{\partial x_k} \ \frac{\partial h}{\partial y_k} \ \frac{\partial h}{\partial \theta_k} \ \frac{\partial h}{\partial x_{t1}} \ \frac{\partial h}{\partial y_{t1}} \ \dots \ \frac{\partial h}{\partial x_b} \ \frac{\partial h}{\partial y_b}, \dots \ \frac{\partial h}{\partial x_{tn}} \ \frac{\partial h}{\partial y_{tn}} \right] \quad (5)$$

where,

$$\begin{aligned} \frac{\partial h}{\partial x_{ti}} &= \frac{\partial h}{\partial y_{ti}} = 0, \text{ for } ti \neq b, \text{ and } 1 \leq i \leq n. \\ \frac{\partial h}{\partial x_b} &= \frac{-(x_k - x_b)}{\sqrt{(x_k - x_b)^2 + (y_k - y_b)^2}} \\ \frac{\partial h}{\partial x_b} &= \frac{-(y_k - y_b)}{\sqrt{(x_k - x_b)^2 + (y_k - y_b)^2}} \end{aligned} \quad (6)$$

Only the terms in $H(k)$ directly related to the current range measurement (i.e., the partials with respect to the robot pose and the position of the beacon giving the current measurement) are non-zero. To complete the SLAM fomulation, P (the covariance matrix) is expanded to the correct dimentionality (i.e., $2n+3$ square) when each new beacon is initialized.

Beacon Initialization: For perfect measurements, determining position from range information is a matter of simple geometry. Unfortunately, perfect measurements are difficult to achieve in the real world. The measurements are contaminated by noise, and three range measurements rarely intersect exactly. Furthermore, estimating the beacon location while estimating the robot’s location introduces the further uncertainty associated with the robot location.

The approach that we employ, similar to the method proposed by Olson *et al* [10], considers pairs of measurements. A pair of measurements is not sufficient to constrain a beacon’s location to a point, since each pair can provide up to two possible solutions. Each measurement pair “votes” for its two solutions (and all its neighbors) within a two dimensional probability grid to provide estimates of the beacon location. Ideally, solutions that are near each other in the world, share the same cell within the grid. In order to accomplish this requirement, the grid size is chosen such that it matches the total uncertainty in the solution: range measurement uncertainty plus Kalman filter estimate uncertainty. After all the votes have been cast, the cell with the greatest number of votes contains (with high probability) the true beacon location.

5.2 Results from Kalman Filter SLAM

In this experiment, the true initial robot position from GPS was used as an initial estimate. There was also no initial information, about the beacons, provided to the Kalman filter. Each beacon is initialized in an online method, as described in Section 5.1. Performing SLAM with Kalman filter produces a solution that is globally misaligned, primarily due to the dead reckoning that had accumulated prior to the initialization of a few beacons. Since, until the robot localizes a few beacons, it must rely on dead reckoning alone for navigation. Although this might cause the Kalman filter estimate to settle into an incorrect global solution, the relative structure of the path is still maintained.

In order to properly evaluate the performance of SLAM with Kalman filter, we must study the errors associated with the estimated path, after removing any global translational/rotation offsets that had accumulated prior to the initialization of a few beacons. Figure 4 shows the final 10% of the Kalman filter path estimate after a simple affine transform is performed based on the final positions of the beacons and their true positions. The plot also includes the corresponding ground truth path, affine transformed versions of the final beacon positions and the true beacon locations. Table 3 provides the XTE and ATE for the path shown in Figure 4.

Several experiments were performed, in order to study the convergence rate of SLAM with Kalman filter. The plot in Figure 5 displays the XTE and its 1 sigma bounds for varying amounts of the data used to perform SLAM (i.e., it shows the result of performing Localization after performing SLAM on different amounts of the data to initialize the beacons).

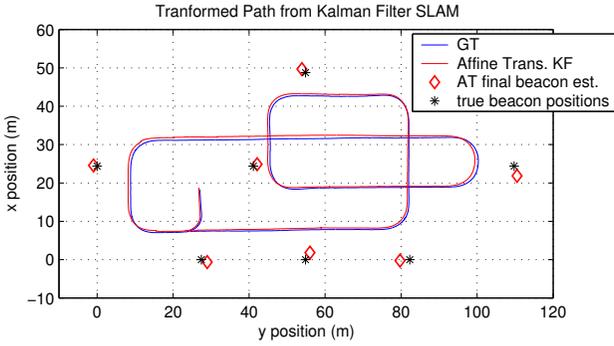


Fig. 4. The path estimate from SLAM using a Kalman Filter (green), the corresponding ground truth (blue), true beacon locations (black *) and Kalman Filter estimated beacon locations (green dimond) are shown. (Note the axes are flipped). A simple affine transform is performed on the final estimate beacon locations from the Kalman Filter in order to re-align the misaligned global solution. The path shown corresponds to the final loop (0.343 km) of the full data set after the affine transform. Numerical results are given in Table 3.

Table 3. Cross-Track and Along-Track Errors for the final loop (0.343 km) of the Data Set after the Affine Transform.

| | XTE | ATE |
|------------------|------------|------------|
| Mean Abs. | 0.5564 m | 0.6342 m |
| Max. | 1.3160 m | 1.3841 m |
| Std. Dev. | 0.3010 m | 0.2908 m |

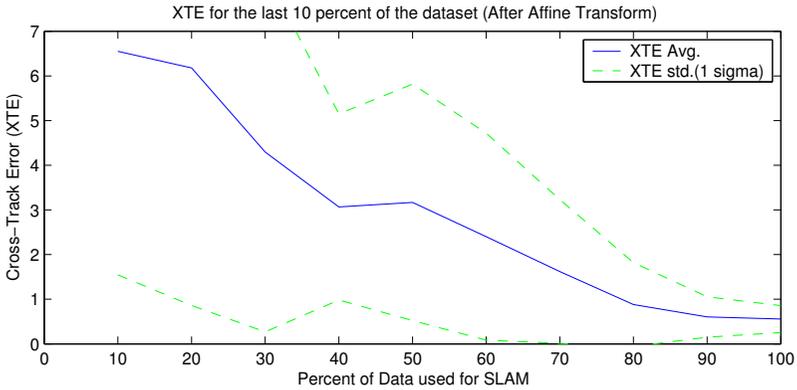


Fig. 5. Kalman Filter Coverage Graph. Varying amount of data is used to perform SLAM, after which the locations of the initialized beacons are fixed and simple Kalman filter localization is perform on the remaining data. The plot above shows the average absolute XTE and its 1 sigma bounds for various subsets of the data used for SLAM.

6 Summary

This paper has reported on extensions for increasing robustness in localization using range from radio. We have examined the use of a particle filter for recovering from large offsets in position that are possible in case of missing or highly noisy data from radio beacons. We have also examined the case of estimating the locations of the beacons when their location is not known ahead of time. Since practical use would dictate a first stage in which the locations of the beacons are mapped and then a second stage in which these locations are used, we have presented an online method to locate the beacons. The tags are localized well enough so that the localization error is equal to the error in the case where the tag locations are known exactly in advance.

References

1. P. Bahl and V. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *In Proc. of the IEEE Infocom 2000*, Tel Aviv, Israel, 2000.
2. D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1999.
3. M. Isard and A. Blake. Condensation: conditional density propagation for visual tracking. In *International Journal of Computer Vision*, 1998.
4. R. Want, J. Hightower, and G. Borriello. Spotted: An indoor 3d location sensing technology based on rf signal strength. Technical report.
5. G. Kantor and S. Singh. Preliminary results in range-only localization and mapping. In *Proceedings of IEEE Conference on Robotics and Automation*, Washington D.C., USA, May 2002.
6. D. Kurth. Range-only robot localization and slam with radio. Master's thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2004. tech. report CMU-RI-TR-04-29.
7. D. Kurth, G. Kantor, and S. Singh. Experimental results in range-only localization with radio. In *Proceedings of IROS2003*, Las Vegas, USA, October 2003.
8. A.M. Ladd, K.E. Bekris, G. Marceau, A. Rudys, D.S. Wallach, and L.E. Kavraki. Robotics-based location sensing for wireless ethernet. In *Eighth ACM MobiCom*, Atlanta, GA, September 2002.
9. A. Chakraborty, N. Priyantha, and H. Balakrishnan. The cricket location support system. In *In Proc. of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 2000)*, Boston, MA, August 2000.
10. Edwin Olson, John Leonard, and Seth Teller. Robust range-only beacon localization. In *Proceedings of Autonomous Underwater Vehicles, 2004*, 2004.
11. S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 101:99–141, 2000.

Experiments with Robots and Sensor Networks for Mapping and Navigation

Keith Kotay¹, Ron Peterson², and Daniela Rus¹

¹ Massachusetts Institute of Technology, Cambridge, MA, USA

{kotay|rus}@csail.mit.edu

² Dartmouth College, Hanover, NH, USA

rapjr@cs.dartmouth.edu

Summary. In this paper we describe experiments with networks of robots and sensors in support of search and rescue and first response operations. The system we consider includes a network of Mica Mote sensors that can monitor temperature, light, and the movement of the structure on which they rest. We also consider an extension to chemical sensing in simulation only. An ATRV-Mini robot is extended with a Mote sensor and a protocol that allows it to interact with the network. We present algorithms and experiments for aggregating global maps in sensor space and using these maps for navigation. The sensor experiments were performed outdoors as part of a Search and Rescue exercise with practitioners in the field.

Keywords: Sensor network, search and rescue, robot navigation

1 Introduction

A network of robots and sensors consists of a collection of sensors distributed over some area that form an ad-hoc network, and a collection of mobile robots that can interact with the sensor network. Each sensor is equipped with some limited memory and processing capabilities, multiple sensing modalities, and communication capabilities. Sensor networks extend the sensing capabilities of the robots and allow them to act in response to events outside their perception range. Mobile robots extend sensor networks through their ability to bring new sensors to designated locations and move across the sensor field for sensing, data collection, and communication purposes. In this paper we explore this synergy between robot and sensor networks in the context of search and rescue applications.

We extend the mapping and navigation algorithms presented in [8] from the case of a static sensor network to that of a mobile sensor network. In this algorithm, the sensor network models the sensor readings in terms of “danger” levels sensed across its area and creates a global map in sensor space.

The regions that have sensor values above a certain threshold are represented as danger. A protocol that combines the artificial potential field of the sensors with a notion of “goal” location for a mobile node (perhaps to take a high resolution picture) computes a path across the map that maintains the safest distance from the danger areas. The focus of this paper is on particular issues related to building systems of sensors and robots that are deployable in realistic physical situations. We present sensor network mapping data from our participation in a search and rescue exercise at Lakehurst, NJ. We then show how these kinds of maps can be used for navigation in two settings: (1) in a simulated scenario involving chemical spills and (2) in a physical scenario implemented on Mica Motes [3] that sense light and guide an ATRV-Mini robot.

2 Related Work

This work builds on our research agenda for providing computational tools for situational awareness and “googling” for physical information for first responders [5]. We are inspired by previous work in sensor networks [2] and robotics [6]. [7] proposes a robot motion planner that rasterizes configuration space obstacles into a series of bitmap slices, and then uses dynamic programming to compute the distance from each point to the goal and the paths in this space—this is the inspiration for our distributed algorithm. This method guarantees that the robot finds the best path to the goal. [4] discusses the use of an artificial potential field for robot motion planning. The concept of using a sensor network to generate a potential field of sensed “danger” and then using this information to generate a path of minimum danger from a start location to a goal location was proposed in [8]. In this paper, the proximity to danger is based on the number of hops from nodes which broadcast danger messages, and the total danger is the summation of the danger potentials from all nodes which sense danger. Then, given start and goal node locations, it is possible for the network to direct the motion of the agent from node to node along a path which minimizes the exposure of the agent to danger. In a related work, [1] addresses coverage and exploration of an unknown dynamic environment using a mobile robot and beacons.

3 Guiding Algorithm

To support guidance, the sensor network computes an adaptive map in perception space. The map is used by mobile nodes to compute safe paths to goal locations. The goals may be marked by a user or computed internally by the network. The map is built from locally sensed data and is represented globally as a gradient starting at the nodes that trigger sensor values denoting danger, using the artificial potential protocol described in [8]. Given such a map, we

Algorithm 1 Algorithm for following a path to the goal node.

```

1: GoalId = 19
2: QueryId = NONE
3: NextNode.Id = NONE
4: NextNode.Potential = POTENTIAL_MAX
5: NextNode.Position = (0,0)
6: Error = RobotSynchronize()
7: while !Error AND (NextNode.Id != GoalId) do
8:   if QueryId == NONE then
9:     Address = TOS_BCAST_ADDR {send query to all nodes}
10:  else
11:    Address = QueryId {send query to the next node on the goal path}
12:    NextNode.Id = NONE {set this to detect 0 responses}
13:    NodeQuery(Address, GoalId) {send the query}
14:    for all Node query responses,  $R_i$  do
15:      if ( $R_i$ .Potential < NextNode.Potential) OR (( $R_i$ .Potential ==
        NextNode.Potential) AND ( $R_i$ .Hops < NextNode.Hops)) then
16:        NextNode.Id =  $R_i$ .Id
17:        NextNode.Potential =  $R_i$ .Potential
18:        NextNode.Hops =  $R_i$ .Hops
19:        NextNode.PriorId =  $R_i$ .PriorId
20:        NextNode.Position =  $R_i$ .Position
21:      if (NextNode.Id == NONE) OR ((QueryId != NONE) AND
        (NextNode.Id != QueryId)) then
22:        QueryId = NONE {no valid response, go back to broadcast}
23:      else
24:        QueryId = NextNode.PriorId {PriorId is the next node on the goal path}
25:        Error = RobotMove(NextNode.Position) {move to position of best  $R_i$ }

```

modify the algorithm in [8] to compute safe navigation paths as shown in Algorithm 1.

Once a path query message is sent, the replies are processed to select the best path available. This is done by selecting the response with the lowest danger potential. If two or more replies with the minimum danger potential are received, the reply with the minimum number of hops to the goal is used to select the path.

4 Sensor Experiments

4.1 Search and Rescue Experiments

Experiments were conducted on February 11, 2005 at the Lakehurst Naval Air Base, NJ as part of the New Jersey Task Force 1 search and rescue training exercise. The purpose of the experiments was to validate the utility of sensor networks for mapping, to characterize the ambient conditions of a typical environment for search and rescue missions. 34 Mica Motes with light,

temperature, and acceleration sensors were manually placed on a large pile of rubble which is used to simulate the environment of destroyed buildings. The rubble consists mostly of concrete with embedded rebar or steel cable, though various destroyed appliances and scraps of sheet metal are also constituent elements. In this experiment, node locations were given to the Motes via radio messages after deployment.

The sensors were in place by 11:15am and gathered data until 12:45pm. The sensor data was stored on each Mote and downloaded at a later time. The day was cold (below freezing at the start of tests), clear and sunny, and very windy. The sensors were placed at approximately 1.2 meter intervals to form a 6x6 meter grid. People and robots were traversing the rubble while readings were taken. The particular section of rubble where the sensors were placed can be seen in Fig. 1.



Fig. 1. Wide angle view of sensors placed on rubble. Most of the sensors are behind outcroppings and hence are not visible. The circles show the locations of a few of the sensors.

To protect them from dust and weather, the sensors were placed in ziploc freezer bags. All had fresh batteries and were placed so the light sensor was generally pointing up, though most of the sensors were not on level surfaces. After about 35 minutes, sensor 23 blew off its concrete perch and fell down a two meter hole. This event is discernable in the graphs that follow. The strong winds also rearranged some of the sensors during the course of the experiment. Four sensors failed, most likely due to temperature extremes, and hence produced no data.

Sensor Radio Connectivity

The connectivity between the sensors was measured by each sensor sending and acknowledging 20 ping messages. The results are shown in Fig. 2 (left).

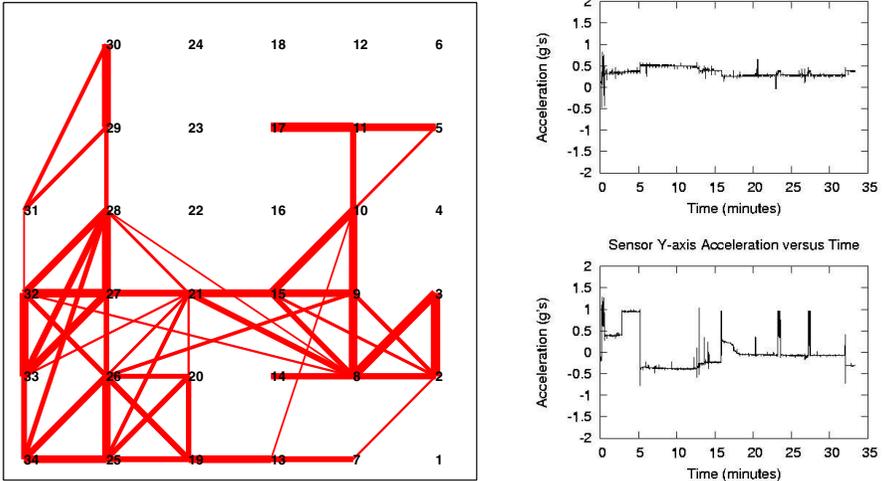


Fig. 2. Connectivity and acceleration data for sensors placed on rubble. The left image shows connectivity between sensors in the rubble field. The relative thickness of the lines indicates the connection strength. Those sensors near the top of the rubble pile had poor connectivity. The right images show X-axis (top) and Y-axis (bottom) acceleration data for Sensor A, shown in Fig. 1 as the leftmost black circle.

While the lower laying sensors, which were mostly on flat slabs of concrete, had reasonable connectivity, the sensors embedded in the more diverse rubble higher in the pile had poor connectivity. In previous experiments we have found that we get fair connectivity with a distance of two meters between sensors lying on earth. Even with the shorter distance we used between sensors here (1.2 meters) the metal in the environment has reduced connectivity dramatically. The three dimensional nature of the sensor deployment is also likely to have had an effect, since the antennas of the sensors are not on the same plane, and have varying orientations. The antennas have a toroidally-shaped reception pattern with poor connectivity along the axis of the antenna.

Light Sensing

Fig. 3 shows a two dimensional light intensity map at three different times during the experiment. The map is bilinearly interpolated from the sensed light values, with the nearest two sensors contributing to the points in the map between them. Because the light sensors were pointed at the sky and it was a bright day, they saturated at their maximum value for most of the test, even if they were in shadow, although near the end of the test shadows from the moving sun brought some of the sensors out of saturation. The light reading for sensor 23 goes down after it falls into a hole. Later, light again reached the sensor from another angle. Other sensors experienced some brief

changes in intensity due to the shadows of people walking by. The shadows from the rubble and the wind causing sensors to shift account for the rest of the changes.

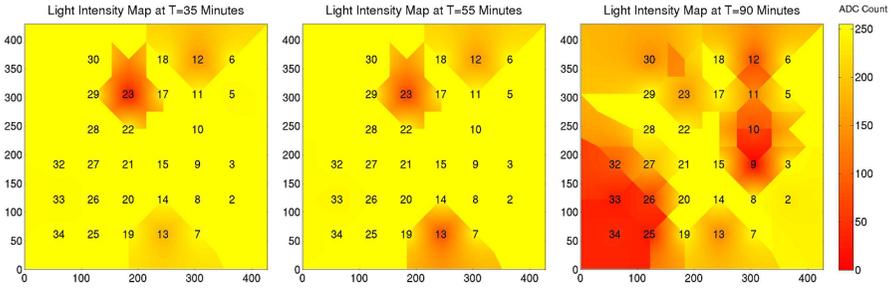


Fig. 3. Sensed light intensity map at times of 35, 55, and 90 minutes.

Temperature Sensing

Fig. 4 shows a two dimensional temperature map at three different times during the experiment. The temperature varied dramatically over time and based on sensor location. The Motes got quite warm (40C, 105F), which is surprising since the day was cold and there was a strong wind blowing, though the day warmed up gradually. The Motes were in plastic bags and the black plastic of the battery holder and the black heat sensor itself were exposed directly to sunlight. The bags acted as insulators from the cold, holding warm air inside and the sunlight on the black plastic heated the air in the bags. The black heat sensors themselves were also heated to higher than surrounding temperature by the sun. This is quite interesting since it shows that a sensor in an environment isn't necessarily sensing that environment. It needs some direct connection to the outside world, or else it is only sensing it's own microclimate.

Mote 23 fell into a hole at about 35 minutes and cools down slowly (when viewing the complete data set). The sensors near the base of the rubble and on the peak of the rubble were mostly laying exposed on flat surfaces. The in-between sensors were in amongst jumbled rubble and recorded cooler temperatures. The sensors most exposed to the sun became the warmest. The changes in temperature were caused by the sun warming the air as it rose higher, the shifting of shadows in the rubble, and sensors being shifted by the wind.

Acceleration Sensing

In addition to the light and temperature sensors, three acceleration sensors were deployed. The sensing element was an Analog Devices ADXL202E, 2-

axis device, capable of measurements down to 2 *milli-gs*, with a $\pm 2g$ range. Due to a higher data rate, we were only able to record about a half hour of readings.

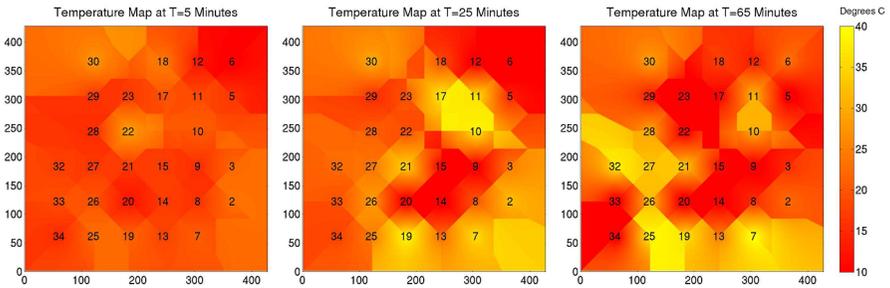


Fig. 4. Sensed temperature map at times of 5, 25, and 65 minutes.

Fig. 2 (right) shows the readings from the sensor laying on the ground, The readings at the start of each graph show the sensors being placed in position. Sensor A was picked up and then replaced by a task force member halfway through the experiment. It apparently slowly tipped over during the course of the next four minutes. The other large shifts in the readings are due to wind blowing the plastic bags the sensors were in. We have collected similar data from several other sensors placed on loose rubble at various points up the rubble pile.

Between wind, weather, shifting rubble, people moving about, lighting and temperature changes due to the motion of the sun, local variations in line of sight, and the jumbling of the radio environment by sheet metal and rebar, this is clearly a challenging environment for wireless sensing applications.

4.2 Chemical Sensing Experiment

In many first response calls the presence of deadly, invisible chemicals is first noticed when people start coughing or falling ill. Even after the presence of a gas has been verified, unless it is visible it is difficult to avoid exposure due to air motion. Chemical sensing networks can provide a first warning of nearby toxins, and more importantly, can tell us where they are, where they are moving towards, and how to avoid them.

As part of ongoing work in medical and environmental sensors for first responders, we devised a simulated air crash scenario that involves a chemical leak. The crash throws some debris into a nearby farmers field where a tank of anhydrous ammonia used as fertilizer is present on a trailer attached to a tractor. Anhydrous ammonia, when released into the atmosphere, is a clear colorless gas, which remains near the ground and drifts with the wind. It attacks the lungs and breathing passages and is highly corrosive, causing

damage even in relatively small concentrations. It can be detected with an appropriate sensor such as the Figaro TGS 826 Ammonia sensor. We ran experiments designed to map the presence of an ammonia cloud and guide a first responder to safety along the path of least chemical concentration. The sensors were Mica Motes, programmed with the same potential field guidance algorithm described above in Section 3. Light sensors on the Motes were used instead of ammonia sensors, due to the difficulty of working with ammonia gas.

The sensors were programmed with locations arranged in a grid with 50 feet between sensors. The experiment was carried out on a tabletop with the RF transmission range of the sensors reduced to match a physically larger deployment. Radio messages between sensors were limited by software to one hop, using the known locations of the sensors, to reduce message collisions. Potential field messages were repeated five times to ensure their propagation. The computed field values were read from the sensors every four seconds to update a command and control map display. It took 10 to 15 seconds for the potential field to propagate each new event and stabilize. Chemical detections were triggered at sensors 9, 20, and 32.



Fig. 5. (Left) Potential field danger map computed by sensors in response to the simulated presence of a chemical agent. (Right) Safest path computed for a trapped first responder by the sensor field.

Fig. 5 (Left) shows a danger map computed by a 38 sensor field after the ammonia has been detected. The danger in the areas between the sensors is computed using a bilinear interpolation of the stored potential field values from the nearest two sensors to each point on the map.

After the ammonia has moved into the locale of the field operations, a first responder located in the lower left corner (the + symbol there) needs guidance to safely find a way to the operations base (the + symbol in the upper right corner.) The guidance algorithm uses the potential field to compute all safest directions from one sensor to the next, and then computes the overall safest

and most direct path for the first responder to follow, which minimizes the exposure to the chemical agent. Fig. 5 (Right) shows the safest path computed by the sensor field.

Such a system can not be relied on by itself for guiding people through danger. This is due in part to the presence of obstacles which the sensors know nothing about, such as fences, bodies of water, vehicles, etc. In addition, the commander on the scene may have additional knowledge which overrides the path chosen by the sensors (e.g., if there's a tank of jet fuel which is overheating along part of the path, it may be best to try a different way, even if the chemical haze is thicker.) Thus, although the sensors guidance computations can not be the sole source for guidance, they can be a very useful addition to the knowledge which the commander and responders in the field use to choose a way to safety.

4.3 Robot Navigation Experiment

We implemented the algorithm used for the simulations in Fig. 5 and the algorithm for safe path following, Algorithm 1, on a system of physical robots and sensors. In our implementation we have a notion of “obstacle” or “danger” much like in the chemical spread simulation. The values for danger could come from any of the sensors deployed and tested as described above.

Experimental Setup

Our experimental setup consists of a network of Mica Motes suspended above the ground and an iRobot ATRV-Mini robot being guided through the network space. The network space was above a paved area surrounded by grass. The goal was to guide the robot from a start location to a goal location along a curved path, while avoiding any grassy area which corresponds to “danger”.

Network

The network is comprised of 18 Mica Motes attached to a rope net suspended above the ground (see Fig. 6). Motes were attached at the rope crossing points, spaced 2 meters apart. Deploying nodes above ground level improves radio performance and prevents damage from the robot wheels. Although the use of a net is not representative of a typical real-world deployment, it allowed research to proceed without fabricating protective enclosures for the Motes and it does not invalidate the algorithmic component of the work.

In our implementation, the location of the nodes in the network is known a priori, since the Mica Motes are not capable of self-localization without extra hardware. When the robot communicates with the next node on the path to the goal, the node passes its location to the robot. The robot then uses its compass and odometry to move to the node location.

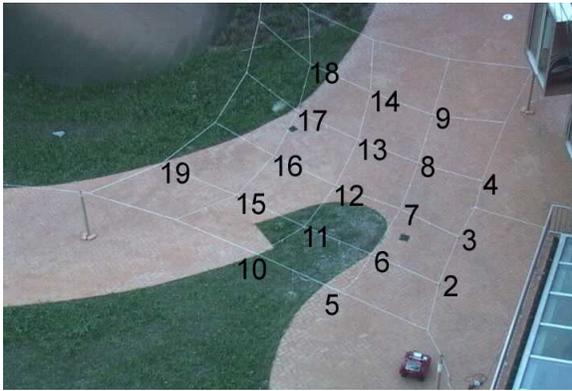


Fig. 6. The network configuration for the guided navigation experiments. 18 Mica Motes are located at the junctions of the ropes, indicated by the ID numbers. Nodes 10, 11, 12, and 18 broadcast “danger” messages, and node 19 is the goal. Node number 1 is on the robot, communicating with the network and guiding the robot. The robot is shown in the starting position for the experiments.

The presence of “danger” was detected by uncovering the light sensor on the Mica Mote sensor board. This would cause the node to broadcast five separate danger messages, which would then flood the network due to each receiver relaying the messages to its neighbors. Multiple messages were used as a means of determining “reliable” communication links—if the number of received danger messages is above a threshold based on the number of expected danger messages, then the link is determined to be reliable, and is therefore suitable to be on a guiding path [8]. In the experimental setup shown in Fig. 6, nodes 10, 11, 12, and 18 sensed danger, equivalent to being over grass in this case. The goal node was number 19, and the robot starting position is shown in Fig. 6. For these conditions, the optimal path consists of nodes 2, 3, 4, 9, 13, 16, 19.

The robot used in our experiments is an iRobot ATRV-Mini with a four-wheel skid-steer drive. It is equipped with an internal computer running Linux, as well as odometry, sonar, compass, and GPS sensors. For our experiments, the sonar sensors were only used to avoid collisions by stopping the robot if any sensor detected an object with 25cm of the robot. The GPS sensor was not used in our experiments, since its resolution was inadequate for the size of the network. Odometry was used to measure forward motion of the robot, and the compass was used to turn the robot to a new heading.

The interface to the network is by means of an onboard Mote connected to the robot by a serial cable. In fact, the onboard Mote is in command of the system and the ATRV-Mini is merely a locomotion slave. The path algorithm described in Section 3 is run on the Mote, which sends motion commands to the robot and receives acknowledgements after the move is completed.

Experimental Results

Experiments were performed with the setup shown in Fig. 6. A sequence of snapshots of an experimental run is shown in Fig. 7. Initial experiments were hampered by poor compass performance on the ATRV-Mini robot, due to the compass being mounted too close to the drive motors. Despite turning the robot drive motors off to minimize the compass deflection the compass heading was not precise, due to the large amount of steel in the adjacent building and buried electrical cables. The result is some additional deviation from the path node positions as shown in Fig. 7.



Fig. 7. Six snapshots of a guided navigation experiment. The Mica Motes are located as shown in Fig. 6. The optimal node sequence is 2, 3, 4, 9, 13, 16, 19. Because the viewing angle is not directly from above and there is some distortion in the net, the robot does not line up exactly with the node locations.

Despite the difficulties with the compass, the navigation was successfully performed many times. Although the final location of the robot is offset from node 19, the robot did follow the path of minimum danger and avoided the danger areas while being guided by the sensor network. We plan to conduct further experiments to get better statistics on the precision of this navigation algorithm.

Discussion

This implementation demonstrated robot guidance by a sensor network. The sensor network is very effective at computing useful global maps in perception space. However, the precision of the navigation system is greatly dependent on the robot hardware. Navigation by compass can be problematic if environmental factors such as electrical cables and steel structures exist. Although it

is possible to compensate for these effects in a known environment, a search and rescue scenario may not permit elaborate offline calibration.

Another option would be to use a directional antenna in place of (or in addition to) the compass. The standard Mica Mote antenna is omnidirectional, but using a directional antenna would allow the robot to determine a bearing to the next node in the goal path. This technique, coupled with the use of radio signal strength (RSSI) to determine the proximity of the robot to a node would make network localization optional, since the robot could directly sense its proximity to a node.³ Since localization is sometimes not possible in a sensor network due to the cost of additional hardware such as GPS sensors or acoustic transducers like those on the MIT Crickets, enabling the robot to move through a non-localized network is a useful feature. It is also cost effective since adding extra hardware to a small number of robots is less costly than adding localization hardware to all the nodes in a large sensor network.

Acknowledgements

This work has been supported in part by Intel, Boeing, ITRI, NSF awards 0423962, EIA-0202789, and IIS-0426838, the Army SWARMS MURI. This project was also supported under Award No. 2000-DT-CX-K001 from the Office for Domestic Preparedness, U.S. Department of Homeland Security. Points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security.

References

1. M. Batalin and G.S. Sukhatme. Efficient exploration without localization. In *Int. Conference on Robotics and Automation (ICRA)*, Taipei, May 2003.
2. D. Estrin, R. Govindan, and J. Heidemann. Embedding the internet. *Communications of ACM*, 43(5):39–41, May 2000.
3. J. Hill, R. Szwedczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. In *ASPLOS*, pages 93–104, 2000.
4. D. Koditschek. Planning and control via potential functions. *Robotics Review I (Lozano-Perez and Khatib, editors)*, pages 349–367, 1989.
5. V. Kumar, D. Rus, and S. Singh. Robot and sensor networks for first responders. *Pervasive Computing*, 3(4):24–33, December 2004.
6. J.-C. Latombe. *Robot Motion Planning*. Kluwer, New York, 1992.
7. J. Lengyel, M. Reichert, B. Donald, and D. Greenberg. Real-time robot motion planning using rasterizing computer graphics hardware. In *Proc. SIGGRAPH*, pages 327–336, Dallas, TX, 1990.
8. Q. Li, M. de Rosa, and D. Rus. Distributed algorithms for guiding navigation across sensor networks. In *MOBICOM*, pages 67–77, San Diego, October 2003.

³ Although RSSI cannot provide reliable absolute distance estimation, it still may be sufficient for determining the point of closest approach to a stationary Mote.

Applying a New Model for Machine Perception and Reasoning in Unstructured Environments

Richard Grover, Steve Scheduling, Ross Hennessy, Suresh Kumar, and Hugh Durrant-Whyte

ARC Centre of Excellence for Autonomous Systems
The University of Sydney, NSW. 2006, Australia
`r.grover,scheduling,r.hennessy,suresh,hugh@cas.edu.au`

Summary. This paper presents a data-fusion and interpretation system for operation of an Autonomous Ground Vehicle (AGV) in outdoor environments. It is a practical implementation of a new model for machine perception and reasoning, which has its true utility in its applicability to increasingly unstructured environments. This model provides a cohesive, sensor-centric and probabilistic summary of the available sensory data and uses this richly descriptive data to enable robust interpretation of a scene. A general model is described and the development of a specific instance of it is described in detail. Preliminary results demonstrate the utility of the approach in very large, unstructured, outdoor environments.

1 Introduction

The robust interpretation of sensory data represents a critical problem in the development of autonomous systems in a wide range of application areas, particularly those involving natural or unstructured environments. At its most fundamental level perception involves using sensory information to evaluate a series of decisions; understanding how various sensors interact with the surroundings; capturing and interpreting cues including location, geometry, texture, colour and other perceptual properties; and combining and manipulating the available information to improve the robustness of the processes. It is known that as the complexity of an environment increases, the level of informational abstraction required to support a given task also rises and this method aims to directly address this issue.

This paper introduces a new general model for the perception problem and highlights its efficacy through the application to the specific problem of terrain-based navigation and control of an autonomous ground vehicle (AGV). Both the practical development of the vehicle and the supporting data manipulation systems will be discussed in detail.

The proposed model is shown in Figure 1. The model has three significant characteristics:

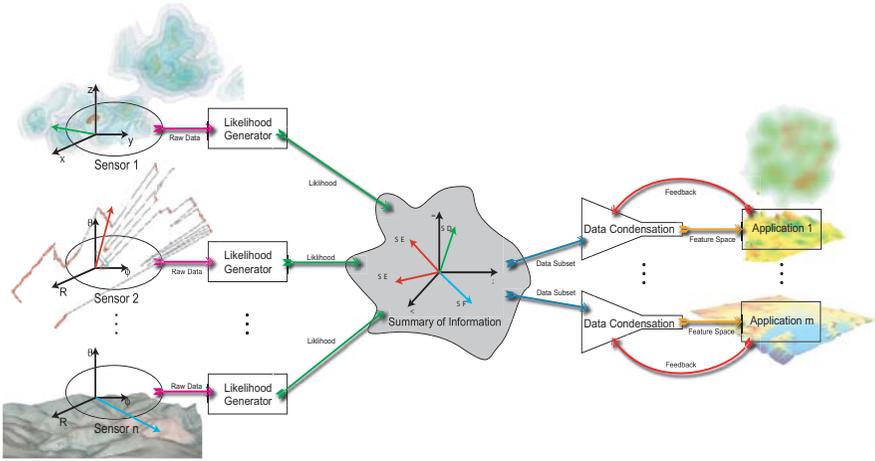


Fig. 1. A schematic view of the proposed perception model highlighting the three main characteristics: a sensor-centric information summary; explicit use of likelihood models; and delayed, application-specific interpretative stages.

- The data is modelled in a *sensor centric* manner, representing the perceptual information in terms of the observed sensory responses. Separate stimuli are described and quantified as separate degrees of freedom in a high-dimensional “sensor space” which captures, in a single structure, the data from all available sensors. No attempt is made, at this stage, to infer the existence of important characteristics or features. The sensory data itself is treated as the best available model of the external world.
- *Uncertainty and ambiguity* in sensing is captured in a probabilistic form as a likelihood function. The explicit use of a probabilistic model allows operations of temporal propagation of data and temporal fusion of information to occur through the use of the Chapman-Kolmogorov and Bayes equations respectively. The actual form used to encode these likelihoods is an important computational issue and many different techniques are appropriate under different circumstances, including: sets of particles, kernel approximations and functional representations.
- The tasks of *perception and reasoning* are interpreted as processes which abstract or compress the stored information. In this context, we focus on re-interpretation of the data for the purpose of increasing the contrast in the data relevant to some specific task. It is shown that entropy measures can be used to explicitly estimate the changes to the information content as a result of this process.

What distinguishes this approach to machine perception is this emphasis on processing data in its ‘raw’ sensory form and delaying any interpretative tasks until the time and place where they are required to achieve some goal. This is motivated by the belief that the sensory stimuli themselves represent

a robust, appropriate, and the most complete, summary of the available data. Sections 2 and 3 introduce the general problem of machine perception and reasoning and derive the general form of the solution as described in the presented model. Section 4.1 introduces the AGV system for which the practical implementation is designed and develops the requirements and characteristics necessary for successful operation. The specific instantiation of the general approaches of section 2 and 3 for this application is described in section 4.3. While the results and methods presented here are preliminary, ongoing work will be described which demonstrates that this approach is an effective implementation for the multi-sensor perception problem.

2 Modelling Unstructured Environments

This section introduces the data gathering task; outlines the justification for the utilisation of the model presented in this work; describes general methods for data manipulation, management and representation; and once the common repository of data is developed, the process of re-interpreting the data to aid task-specific processing is examined. Section 4 derives a specific example of this approach adapted to the AGV problem.

Assume that the sensor-centric representation amounts to estimating the value of m distinct properties over a physical space of dimension n . Each of the individual sensory cues would represent a separate property in this arrangement. This forms a (rank-zero) tensor field over a physical space, which is equivalent to a feature space of $(m+n)$ dimensions except that it explicitly invokes the spatial correspondences between properties. Let the vector \mathbf{y} , an arbitrary vector in \mathbb{R}^n , represent a particular location in the physical space and let the tensor function $\mathbf{x}_k(\mathbf{y})$ yield the values of the m properties at that location for the k^{th} time step.

Real sensors yield a measurement of l properties over the same n -dimensional space. These will be different to the properties to be estimated and will, for any particular sensor, correspond to a subset of the estimated properties. For example, a radar measures return intensities which infer the reflectivity at the radar wavelength. Let the tensor function $\mathbf{z}_k(\mathbf{y}) \in \mathbb{R}^l$ represent the k^{th} observation. Denote by \mathbf{Z}^k the sequence of k observations $\{\mathbf{z}_1(\mathbf{y}), \mathbf{z}_2(\mathbf{y}), \dots, \mathbf{z}_k(\mathbf{y})\}$.

The goal of the framework is to determine an appropriate estimate of the function, that is, $\hat{\mathbf{x}}_k(\mathbf{y})$ given \mathbf{Z}^k . In order to maintain this estimate, we use Bayes' rule to re-arrange the sensor model to yield

$$p(\mathbf{x}_k(\mathbf{y})|\mathbf{Z}^k) \quad (1)$$

where $\mathbf{Z}^k = \{\mathbf{z}_1(\mathbf{y}), \mathbf{z}_2(\mathbf{y}), \dots, \mathbf{z}_k(\mathbf{y})\}$, which is equivalent to the 'standard' formulation of the Bayesian problem, except that explicitly encodes the spatial dependencies of the components in the form of the tensor functions. It is possible to transform the state function $\mathbf{x}_k(\mathbf{y})$ from the spatial frame into

a function space with an appropriately defined basis set. A general form of this approach is the Kernel decompositions, where the function is treated as a linear summation of a known set of basis functions. In the sense of these kernel approximations, the functions can be re-written as

$$\mathbf{x}_k(\mathbf{y}) = \sum_{i=1}^{\infty} \mathbf{K}_i^x(\mathbf{y}, \alpha_i^{x_k}) \quad (2)$$

$$\mathbf{z}_k(\mathbf{y}) = \sum_{j=1}^{\infty} \mathbf{K}_j^z(\mathbf{y}, \alpha_j^{z_k}) \quad (3)$$

where $i, j \in \mathbb{Z}$ are summation indices; the kernels \mathbf{K}_i^x and \mathbf{K}_j^z represent the i and j^{th} functions in the basis sets for each of the equations; and the α are the parameters (including a mixture co-efficient) which define the i^{th} and j^{th} state and observation kernel function respectively.

Now, given that knowledge of the $\alpha_i^{x_k}$ and $\alpha_j^{z_k}$ uniquely defines the functions $\mathbf{x}_k(\mathbf{y})$ and $\mathbf{z}_k(\mathbf{y})$, then the estimation of these functions can be considered as the equivalent problem of estimating these parameters. Thus, we can write the equivalent of equation 1 as

$$p(\alpha^{x_k} | \mathbf{A}^k) \quad \text{where} \quad \mathbf{A}^k = \{\alpha^{z_1}, \alpha^{z_2}, \dots, \alpha^{z_k}\} \quad (4)$$

In order to make this problem tractable, however, it is necessary to select an appropriate kernel basis such that the number of parameters required to yield an acceptable approximation is finite. There are many such approximations, including: truncated Fourier series, truncated Wavelet decompositions and Gaussian mixture models, where we can limit the required number of Gaussians *a priori*. If the number of mixture components is N and M for the state and observation mixtures respectively, then we can rewrite this as,

$$p(\alpha^{x_k} | \mathbf{A}^k) \quad \text{where} \quad \mathbf{A}^k = \{\alpha^{z_1}, \alpha^{z_2}, \dots, \alpha^{z_k}\} \quad (5)$$

and $\alpha^x \in \mathbb{R}^N, \alpha^z \in \mathbb{R}^M$

3 Reasoning/Data Condensation

In this section an alternative model of the feature extraction process is presented, one which is based on the underlying assumption that the vast array of accepted methods all share common features and should, therefore, be describable as subsets of a more general approach. In this approach reasoning is treated as an abstraction (or condensing) process in that it reduces the amount of data during the operation. The overall goal of any feature extraction algorithm is to determine which subset of the sensory data best enables the identification of features, so that the system may achieve its desired goals. In this approach, consider the salient characteristics of the combined sensory

representation as ‘feature descriptors’ and therefore the basis for the identification and description of features. That is, the goal of the interpretative process is to determine firstly the *aspects* of the data which support the identification of the *elements* within it which, in turn, enable some decision process to be evaluated. We will refer to these tasks as feature promotion and extraction respectively. The overall scheme is shown in figure 2. It is beyond the scope of this paper to develop this feature-extraction process fully and a thorough treatment will be available in a forthcoming paper.

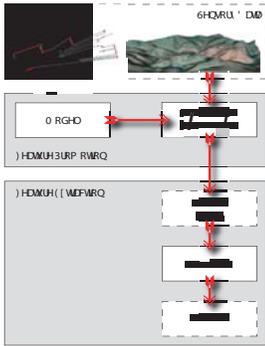


Fig. 2. Feature extraction block-diagram showing promotion and extraction phases in grey



Fig. 3. The Argo vehicle at a recent field trial

The first stage of this algorithm (incorporating the ‘generalised convolution’ and ‘model’ blocks) involves promoting relevant characteristics of the data with respect to the background. As an example, a system which relies on the reliable detection of edges in a visual field will utilise an appropriate methodology depending on the types of edges which are of greatest importance. Fundamentally, this is a contrast enhancing operation and involves the application of biases to the data, treating parts as either significant or noise. Mathematically, these biases can be conveniently encoded as functions and combined with the data, such that the output of the operation as a set of new measures (one for each model function) which locate the data in a new space. This is an explicit re-interpretation of the raw data for a specific purpose.

The final stage of the approach involves isolating and identifying the parts of the signal which are identified *by these characteristics* as distinct from background noise. As the previous stage is a generalised contrast enhancing operation, this step involves a general extension of a thresholding operation. The existence of a meaningful threshold as a feature criteria is implicit in most feature extraction algorithms, consider support vector machines (SVMs) [1] where the nonlinear process is used to generate a hyperplane classifier; or the scale-invariant keypoints of [2] where keypoints are selected at the scale-space extrema in an image space.

4 Practical Implementation

While sections 2 and 3 have introduced generalised approaches to the problem, this section describes the application of this approach to a specific application of the navigation and control of an Autonomous Ground Vehicle.

4.1 The ARGO Project

Consider the mapping of this approach to the autonomous ground vehicle (AGV) shown in figure 3. The vehicle subsystems have been developed with emphasis on two main goals: to ensure reliable operation over extended periods (greater than 24 hours continuous) and to provide a modular, scalable test platform for deployment of technologies resulting from research programmes. The vehicle has already been demonstrated to operate reliably for longer than 8 hours and has performed autonomous path-following over distances in excess of 7.5km and up to 2.5km between individual waypoints. The deployment arena is unstructured, expansive outdoor environments including desert, rural farmland and wooded areas.

4.2 Reasoning Tasks and Sensing Requirements

Reliable long-term navigation and control for this vehicle under mission scenarios including continuous day/night transitions, all-weather operation and with speeds of up to 9ms^{-1} requires a minimum set of capabilities. Most importantly the vehicle must be capable of reliable navigation under all design conditions. There are two obvious candidates: GPS/INS systems [3] and an implementation of the SLAM algorithm [4]. It has been shown, however, that reliability and fault detectability considerations suggest that neither system is suitable in isolation, however an appropriate combination can be developed. Recent results from the DARPA Grand Challenge [5] have demonstrated the effects of undetectable faults in the navigation systems.

Determining the desired path based on available information is also critical for developing a truly autonomous platform. A reliable and efficient planner will necessarily interpret sensory data with respect to a vehicle specific model. In addition to estimating the local surface geometry, characteristics such as ground type (gravel, grass, tarmac, etc.) and condition will be required. Furthermore, the traversability of different regions must be estimated at a larger scale; identifying lakes, salt-pans, rivers, cliffs and known fence-lines among the necessary tasks. Even with reliable navigation and a global plan, it is well known that local variations will prohibit blind path-following. This suggests that a vehicle will require the ability to detect and respond to obstructions (both volumetrically positive and negative) in order to modify the trajectory appropriately.

Candidate sensing modalities for these reasoning capabilities include vehicle-based GPS/INS system, near-field laser and radar scanners, and augmented

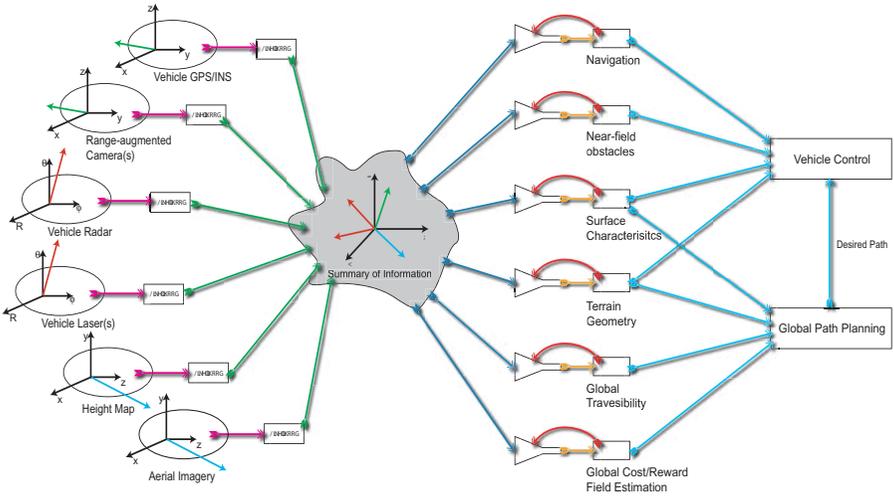


Fig. 4. High-level view of simple implementation of proposed approach to an Autonomous Ground Vehicle (AGV).

and non-augmented cameras for navigation and near-field sensing. Efficient global planning suggests the utilisation of satellite (or aerial) height data, visible and hyper-spectral imagery and surveyed meta-data (fence-lines, waypoints, beacons etc.). This method is advocated as each of these separate reasoning tasks are dependent on different subsets of the *same* sensory data. For example, geometry will depend on the direct measurement of the radar and laser reflectivities of the environment but robust feature association for navigation may require some highly abstracted combination of all the sensors. Finally, the system is able to handle asynchronous data gathering, providing the most complete available data at any point in time and is readily extensible to scenarios with multiple heterogeneous platforms. Figure 4 shows a customised version of Figure 1 with these sensors and reasoning capabilities shown.

4.3 The ‘International Rescue’ Project

In this section we introduce a practical implementation (known internally as the ‘International Rescue’ project) which enables the system of figure 4 to be constructed. This implementation provides for a common repository of sensory knowledge and allows simultaneous access to that repository for adding, changing and accessing the data. Fundamentally, the particular selection of the information to model and the operational representation for the manipulation of that data are intimately linked to the computational efficiency provided. Here we consider first an implementation used to demonstrate the applicability of this method, using a single sensor in an expansive outdoor environment.



Fig. 5. Outdoor scanning terrain imaging millimeter-wave radar unit

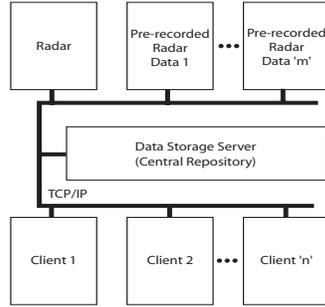


Fig. 6. The network structure used in the International Rescue Project.

The first implementation used a millimeter-wave imaging radar unit developed for outdoor unstructured environments and is shown in figure 5. The location and orientation of the scanner is obtained using high-accuracy GPS measurements throughout the data collection phase. Data from the radar is available as recorded data or ‘live’ from the unit via a TCP/IP socket connection. The network structure for the system is shown in figure 6. The Data Storage Server block in this image is the location where the probabilistic models and common representation are constructed. All the data sources connect directly to this block and the server itself is responsible for handling the incoming data. Multiple clients connected to this data server perform the application-specific tasks such as those shown in figure 1. Since there are multiple clients and sources asynchronously accessing the data server, there is a need to manage the internal data as efficiently as possible.

We treat the input space for this system to be a two-dimensional function for a set of points in a three dimensional space. That is, we will manage the data as a five dimensional vector containing: position (3), radar intensity and source ID. Since the radar provides data directly in a three-dimensional space and since each source or client will act on localised regions of that space, it is advantageous to partition the data vector into a spatial component and the property values. Furthermore, in doing this it is possible to efficiently partition and search the spatial dimensions with the utilisation of sensible data-structures.

Data Storage Structure

In this example, we developed an efficient thread-safe k-D tree. Using this approach the space is successively subdivided in regions with high data content. This ensures that regions which are sparsely known are stored efficiently, while regions where data is heavily located are divided into appropriately sized volumes. A symmetric binary subdivision process is utilised, in which a cell to be divided is divided into eight equal-volume subdivisions. This enables very fast spatial searching through the data.

Having noted earlier that there will be multiple data sources and clients accessing the common data repository it was important to accelerate, where possible, the access to the data but to protect the integrity of that data at the same time. A smallest-container, Multiple-Read, Single-Write (MRSW) mutex system was employed to achieve this. Functionally, an MRSW mutex allows an unlimited number of threads to read simultaneously from the object, but only a single thread may write to the object at any one time, and no thread may read while the object is write-locked. Figure 7 shows the structure of the spatial k-D tree, consisting primarily of the root node, from which the tree descends. Each node in the tree (that is each rectangle in this diagram) can have eight child nodes and may only contain data if it has no child nodes. If the node has no children (denoted by shaded nodes) then the node is called a 'leaf'.

While it would be possible to simply apply a single MRSW mutex to the entire tree, this would mean that the entire structure is locked at once. Since this acts contrary to the desire that the tree should provide for as efficient multi-threading as possible and since it is likely that many operations will act only on small sections of the tree, we look for a more efficient mechanism. In this example, the tree is used to add additional data to an expansive outdoor map using a radar sensor and this sensor will only interact with a small section of the map at any one time. To overcome this constraint we note the following:

1. Data may be efficiently locked at the level of a single leaf node, or a series of leaves. Additionally, a write lock is only required when the data is to be modified.
2. Each node can affect any node below it, and no others. This means that whenever a child node is being accessed in any manner then a node should be locked for reading.
3. A node will only require a write-lock when the structure of the particular node is changing, that is, the node is changing from a leaf node to an internal node. Since every operation on nodes lower in the tree than the current node require a read-lock, then all these operations will be halted until the restructuring is complete.

For these reasons, a three-tier locking structure was employed as shown in the figure. Here any operation proceeds down through the tree placing read-locks on the nodes through which it passes. When the method reaches the leaves in which it is interested it places the appropriate lock on the data. If the points are only required for reading purposes, then there is no need to retain a lock on the points as they can be passed to a calling routine using a copying method. If the structure of the tree is going to change, then the appropriate node locks are requested. As noted earlier, since the MRSW mutex will wait until all reading threads unlocks the object (but prevent any new read locks being successful) before granting the write-lock, this ensures that the minimal portion of the tree is locked during normal operation.

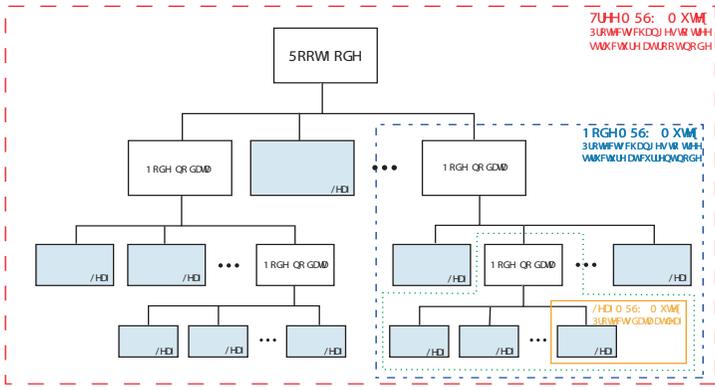


Fig. 7. The structure of the k-D tree and mutex structure for efficient multi-threaded operation. The nodes can all have eight children, corresponding to the binary division of each axis of the original volume. Only leaf nodes can contain data and are shown shaded. Multiple-read, single-write mutexes are provided at the tree, node and leaf levels for efficient access to the tree.

System Development

The structure of the previous section is implemented in a series of custom-built C++ applications. At the simplest level, these programs mimic the structure shown earlier and communicate using TCP/IP sockets to provide the internal links. The sensor data is provided by two applications which, together, mimic a real connection to the live radar, and there is no functional difference for the data storage server whether it is connected to the actual device or the data playback programs. The tree structure suggests an intuitive programmatic structure and is implemented in this manner. Operations which act on the tree therefore simply call the same operations on the child nodes, or return the requested information if the current node is a leaf.

4.4 Experimental Results

This section details the preliminary results obtained by the system. In its current configuration, the system is capable of connecting to both live and pre-recorded data streams, and of having multiple simultaneous clients connect to the data server and conduct any application specific processing. At this stage, only a single client has been developed, primarily to aid in debugging, however multiple instances of this single client can be run. In Figures 8 one example of a client can be seen. This client is taking data from the radar and processing it to put it in a form suitable for display using modern 3D graphics hardware. Figures 8 and 9 shows the system at two stages in the recursive operation, when only a small amount of data has been collected from the radar and once more data has been added. Clearly visible are both the OcTree data structure, as well as the data, which is represented in point form.

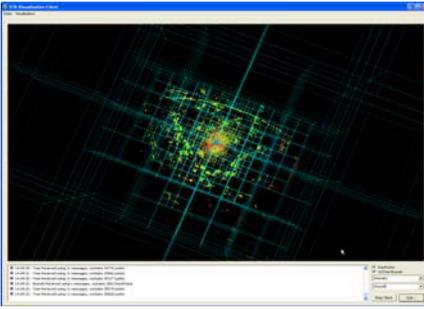


Fig. 8. Screen capture of the data visualisation client. The OcTree and the data contained in it can be seen.

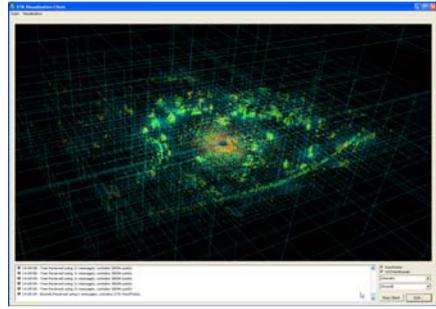


Fig. 9. Screen capture of the data visualisation client. More data has been added to the structure

4.5 Future Work

This section details the areas of ongoing research aimed at extending the framework described in this document. Some of the areas are theoretical, some are implementation related.

Generation of Probabilistic Sensor Models The discussion in this document assumed that any sensor likely to be used in the system will have a likelihood model associated with it. Most existing methods for generating these models tend to be *ad-hoc*. An important area for further research is the generation of probabilistic models for describing the way sensor data maps to world states. It is, in principle, possible to build these models purely from a detailed knowledge of the physics behind the individual sensing process.

Information and Entropy This document has described the general structure and mathematics behind much of the perception problem. One very natural issue that arises, particularly in the area of ‘feature extraction’ (which we conjecture is the most common mode of perception), is that of performance. How can one tell whether the system is extracting any useful information? It is the firm belief of the authors that this question can be directly answered through the use of metrics based on Shannon Information and Entropy. If the information content of a signal can be measured, both before and after feature extraction, then information gain gives a direct measure of performance.

Clients At this stage of the project, most of the resources have gone into the construction of the infrastructure that allows the perception paradigm suggested here to be concretely realised in a robust and rigorous manner. Even though the philosophy espoused here is that many clients can each take data from the central repository to support many individual applications, only one client has been constructed to date - that of visualising the processed data. The construction of more clients is currently underway.

The ultimate aim is to produce a concrete instantiation of the diagram shown in Figure 4.

Multiple Simultaneous Data Sources The capability currently exists to use multiple simultaneous data sources. For example, in the case of the ARGO, it would be advantageous to use camera, laser and radar as the data sources used for perception. At this stage of the project, only a single data source (radar) has been tested, both on and off line, as well as with a mix of on-line and pre-recorded data. An important area for future work is to add the information coming from other sensors. This will show conclusively the many advantages of the theory and implementation described in this document.

5 Conclusions

This paper has introduced a practical perception and reasoning system for assisting operation of an Autonomous Ground Vehicle in an unstructured, outdoor environment. The use of a new model for the analysis of the perception and reasoning problem supports the development of the system, particularly in respect to the selection of data parameterisation and manipulations used. In particular, development of a general interpretation of the tasks of modelling and reasoning provide a direct, quantifiable justification of the utilisation of an unstructured grid representation. Preliminary results show that this approach handles the difficult and expansive environments extremely well and supports the ongoing development of many advanced perception tasks.

References

1. Marti Hearst, Susan T. Dumanis, Edgar Osuna, John Platt, and Bernhard Schölkopf, "Support vector machines," *IEEE Intelligent Systems*, pp. 18–28, July/August 1998.
2. David G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
3. J. H. Kim and S. Sukkariéh, "A baro-altimeter augmented INS/GPS navigation system for an uninhabited aerial vehicle," in *Proc. 6th Int. Conf. on Satellite Navigation Technology (SATNAV03)*, July 2003.
4. R. Smith, Self M., and P. Cheeseman, *Estimating Uncertain Spatial Relationships in Robotics*, Autonomous Robot Vehicles. Springer-Verlag, 1990.
5. Chris A. Urmson, Joshua Anhalt, Michael Clark, Tugrul Galatali, Juan P. Gonzalez, Jay Gowdy, Alexander Gutierrez, Sam Harbaugh, Matthew Johnson-Roberson, Yu Hiroki Kato, Phillip Koon, Kevin Peterson, Bryon Smith, Spencer Spiker, Erick Tryzelaar, and William Red Whittaker, "High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004," Technical Report CMU-RI-TR-04-37, Carnegie Mellon University: Robotics Institute, June 2004.

Constrained Motion Planning in Discrete State Spaces

Mihail Pivtoraiko and Alonzo Kelly

Robotics Institute, Carnegie Mellon University
mihail@cs.cmu.edu, alonzo@ri.cmu.edu

Summary. We propose a principled method to create a search space for constrained motion planning, which efficiently encodes only feasible motion plans. The space of possible paths is encoded implicitly in the connections between states, but only feasible and only local connections are allowed. Furthermore, we propose a systematic method to generate a near-minimal set of spatially distinct motion alternatives. This set of motion primitives preserves the connectivity of the representation while eliminating redundancy – leading to a very efficient structure for motion planning at the chosen resolution.

Keywords: Nonholonomic motion planning lattice control set

1 Introduction

Discrete representation of states is a well-established method of reducing the computational complexity of planning at the expense of reducing completeness. However, in motion planning, such discrete representations complicate the satisfaction of differential constraints which reflect the limited maneuverability of many real vehicles. We propose a mechanism to achieve the computational advantages of discretization while satisfying motion constraints.

To this end we introduce a search space, referred to as the *state lattice*, which is the conceptual construct that is used to formulate a nonholonomic motion planning query as graph search. The state lattice is a discretized set of all reachable configurations of the system. It is constructed by discretizing the \mathcal{C} -space into a hyperdimensional grid and attempting to connect the origin with every node of the grid using a feasible path, an edge, using an inverse trajectory generator. The lattice in general is also assumed to contain all feasible paths, up to a given resolution, which implies that if it is possible for a vehicle to travel from one node to another node, then the lattice contains a sequence of paths to perform this maneuver. Hence, it is possible to conclude that this formulation allows resolution complete planning queries.

Like a grid, the state lattice converts the problem of planning in a continuous function space into one of generating a sequence of decisions chosen from distinct alternatives. Unlike a grid, the state lattice is carefully constructed such that each connection represents a feasible path. A connectivity scheme that intrinsically represents mobility constraints leads to superior motion planning results because no time is wasted either generating, evaluating, or fixing infeasible plans.

To achieve this scheme we attempt to capture *local connectivity* of the state lattice, within a limited neighborhood of any node. We discuss designing a small *control set*, a minimal set of primitive paths that, when concatenated, can re-generate any other path in the lattice. We further show that this formulation lends itself directly to building an efficient search algorithm.

2 Prior Work

The utility of the lattice is hinged on the assumption that it is possible to determine a feasible path between any two configurations in a C-space without obstacles. While this is itself a very difficult problem, it has been the objective of much research in the past century. Frazzoli et al. in [3] suggest that there are many cases where efficient, obstacle-free paths may be computed either analytically or numerically by solving an appropriate optimal control problem. A fast nonholonomic trajectory generator was described in [10]. It generates polynomial spiral trajectories, such that a path is specified by a continuous control function: curvature as a function of path length.

It was shown in [6] that through careful discretization in control space it is possible to force the resulting reachability graph of a large class of non-holonomic systems to be a lattice. However, this is usually difficult to achieve, and under most quantizations the vertices of the reachability graph are unfortunately dense in the reachable set. By using an inverse path generator, we can choose a convenient discretization in control and state space, one that makes the search more efficient. This also allows us to use continuous control functions that are natural for real systems.

The importance and difficulty of enforcing differential constraints also has a long history [1], [9], [5]. A recent trend appears to favor more deterministic variants of the PRM [11]. In [2], Quasi-PRM and Lattice Roadmap (LRM) are introduced by using low-discrepancy Halton/Hammersley sequences and a regular lattice, respectively, for sampling. LRM appeared especially attractive due to its properties of optimal dispersion and near-optimal discrepancy.

Also, a “Lazy” variant of these methods was discussed that avoided collision checking during the roadmap construction phase. In this manner the same roadmap could be used in a variety of settings, at the cost of performing collision checking during the search. An even “lazier” version is suggested, in which “the initial graph is not even explicitly represented” [2]. In this regard,

our approach of using an implicit lattice and searching it by means of a pre-computed control set that only captures local connectivity is very similar to the Lazy LRM. Our contribution is in exploring the conjecture made in that work and successfully applying it to nonholonomic motion planning.

Initial concepts of this work were explored in a successful field implementation of a nonholonomic motion planner built using the state lattice of limited size represented explicitly [4].

3 State Lattice

In this section we describe the state lattice as a generalization of a grid and present it as a search space for efficient constrained motion planning as heuristic search.

3.1 Inverse Path Generation

Among several approaches discussed in Section 1.1 that allow finding a sequence of controls from a given initial configuration to a final configuration, we evaluated the one described in [10]. This approach allows fast generation of nonholonomic trajectories. The assumed form of the solution path is a curvature polynomial of arbitrary order. The method was shown to provide good results and in principle allows optimization w.r.t. various criteria, e.g. least curvature variation. The continuous specification of paths was convenient to manipulate and execute in vehicle controllers. The method executes practically in real-time: a query is computed in about 1 millisecond.

3.2 Constructing the State Lattice

Discretization is central to defining the state lattice as a generalization of a grid. Discretization converts the motion planning problem into a sequential decision process. We adopt the typical strategy of assuming that decisions are made only at discrete states. The states are the nodes in the lattice and the motions that connect the states are the edges. While the state vector can certainly have arbitrary dimension, for this paper we have implemented the state lattice in 4 dimensions. Each node of the lattice therefore represents a 4-dimensional *posture* that includes 2D position, heading and curvature.

If the discretization exhibits any degree of regularity, then the spatial relationships between two given states will reoccur often due to the existence of other identically arranged pairs of states. A discretization exhibiting some degree of regularity leads to a set of motion options which is similarly regular.

For the balance of the paper, we adopt the assumption that the state space discretization is regular in at least the translational coordinates (x, y) . Specifically, if the path between two postures:

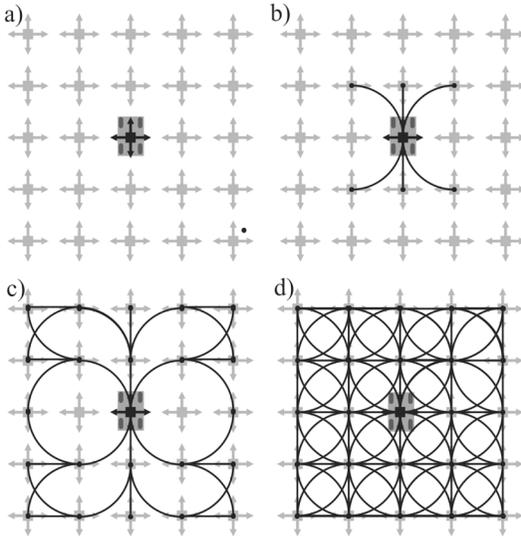


Fig. 1. Constructing the lattice for the Reeds-Shepp Car. In a) we define a discretization in C-space (an (x, y) grid is chosen here, arrows indicate allowed headings), an origin is chosen; b) for 8 neighbor nodes around the origin, feasible paths are found; c) same query is extended outward to 24 neighbors, only a few direct paths shown; d) complete lattice.

$$[x_i, y_i, \theta_i, \kappa_i] \rightarrow [x_f, y_f, \theta_f, \kappa_f]$$

is feasible, then so is the path

$$[x_i + n\Delta_l, y_i + n\Delta_l, \theta_i, \kappa_i] \rightarrow [x_f + n\Delta_l, y_f + n\Delta_l, \theta_f, \kappa_f]$$

for any integer n and (x, y) -discretization step size Δ_l . While the starting and ending states for two such motions are distinct, the motion itself (perhaps encoded as a steering function) is not.

With these properties in mind we construct the state lattice by using the inverse path generator to find paths between any node in the grid and the arbitrarily chosen origin. Fig. 1 illustrates lattice construction for the Reeds-Shepp car. By regularity, we can copy the resulting set of feasible paths to any node in the lattice. In the limit, as the lattice is built by including all feasible motions from any point, it will approach the reachability graph of the vehicle, up to a chosen resolution. Without loss of generality, we henceforth consider the state lattice to be a valid representation of the system’s reachability graph.

Since we discretize state space, it is consistent to consider discretizing paths through space in a similar fashion. We consider two paths (with identical endpoints) which are “sufficiently” close together to be *equivalent*. We define a path τ_1 to be equivalent to τ_2 if τ_1 is contained in a certain region Q around τ_2 , defined as a set of configurations within a certain distance δ_e of τ_1 , given some metric ρ :

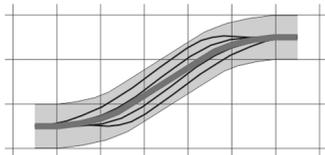


Fig. 2. Path Equivalence. A variety of paths between two configurations (thin black lines) that are contained in the boundary (grey region) are considered to be equivalent and represented by a canonical path (thick grey line).

$$\forall q \in \tau_1, \forall q' \in \tau_2, Q = \{q' | \rho(q', q) < \delta_e\} \quad (1)$$

All paths that satisfy this criterion are considered to belong to the same equivalence class (Fig. 2). It is important to note that this definition of path equivalence is consistent with applications to mobile robotics. Typically, there is a certain error of path following for realistic vehicles. By exploiting this error, motion planning can be made more efficient, as presented below.

4 Control Set

Here we present a principled method of choosing the neighborhood of the lattice that both offers practical guarantees of best exploration of the lattice and keeps the neighborhood size small to preserve search efficiency. We strive to obtain the minimal set of controls as motion alternatives that captures local connectivity of the lattice.

4.1 Path Decomposition

With the insights obtained in Section 2.2, we again look at the lattice as a concept derived from a grid. The regularity property of the grid implies that it is possible to isolate a certain representative set of connections which is repeated everywhere in the grid. As is illustrated in Fig. 3a, for the case of a 4-connected rectangular grid, it is easy to identify the minimal set of connections. The grey lines in this figure represent all possible paths in the grid. Four thick black paths in the center constitute the minimal set of paths. Any path through this grid can be decomposed into a sequence of “primitives”, paths in the minimal set. If we cast the grid in the context of motion planning, we understand that this minimal set enables us to generate arbitrarily long motion plans in the infinite grid. This concept has been used in motion planning for some time [1].

In a similar fashion, if we could identify such a control set for a lattice, we could use it to address the computational issues mentioned above and essentially create a finite representation of the lattice.

By invoking the notion of path equivalence class and some $\delta_e > 0$, we can substitute a path with two other paths such that their concatenation generates

a motion that belongs to the same equivalence class as the original path. We define path decomposition as the problem of finding two such constituents of a path (Fig. 3b).

By definition of path decomposition, the two constituent paths must meet at a lattice node. Intuitively, the longer a path is, the more lattice nodes it comes “close” to, hence the easier it is to find a decomposition because there are more “opportunities” to do so. Through a simulation study we concluded that it is possible for realistic vehicle parameters (κ_{max} and δ_e) to decompose the entire assumed infinity of motions in the state lattice, of arbitrary length, in this manner. We considered over 2000 different (relatively long) paths in the lattice and showed that all of them could be decomposed into at least two (usually more) smaller paths. Thus, the control set allows us to eliminate redundancies of the lattice both in terms of the variety of paths between nodes (through the notion of path equivalence), and in terms of generally unlimited path length (path decomposition).

4.2 Generating the Control Set

Given a method to generate the set of distinct feasible paths to a single state, the control set can be generated by a process of structured elimination. First, paths to all states one unit from the origin are generated, then, paths to all states two units from the origin, etc. When a path is considered, it is tested for passing sufficiently close to an intermediate state, and if so, it is removed from the control set because it can be decomposed into the path to this state from the origin and the path from this state to the end-point. Since we are moving radially outward, any path that can be decomposed may be removed from the control set because its “ingredients” have already been considered.

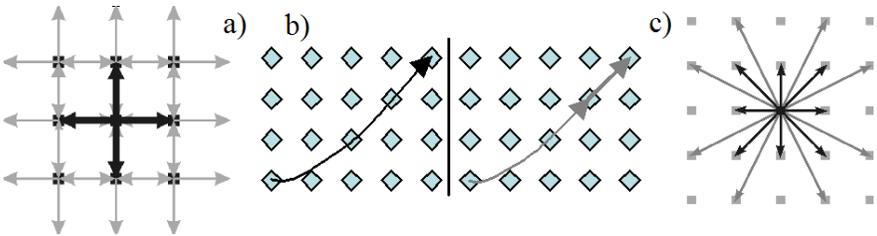


Fig. 3. a) Isolating a minimal set of paths in a 4-connected grid. b) An illustration of path decomposition. Blue diamonds are lattice nodes; the black curve on the left is some arbitrary path (that starts and ends on nodes). On the right, we show that it can be decomposed into two (grey) smaller paths that meet at another node. c) Special heading discretization that allows considering straight lines as much as possible. Black arrows show 8 equal heading intervals, grey arrows represent finer, non-uniform discretization: 8 additional intervals chosen such that the grey arrows end on nodes as well.

Each of them is either in the control set already or does not need to be because it itself is decomposable.

This process terminates at the certain radial distance from the origin when all paths at that distance can be decomposed. Through simulation studies similar to the one mentioned in Section 3.1 we verified that this termination condition is a good heuristic for obtaining a control set that spans the entire state lattice.

Although a uniform discretization of heading is an option, we found it useful to discretize heading in the non-uniform manner. The motivation for this is to enable the planner to produce straight-line paths as much as possible. For example, if the goal is a node that is to the left two nodes and up one node from the origin, then a vehicle can get there in a straight line if its heading is $\arctan(1/2) = 26.6^\circ$. Therefore, we define 8 equal heading intervals of 45° , and also 8 non-uniform intervals chosen such that there can be straight paths from the origin to all nodes within the radius of two nodes around the origin. Certainly, increasing this radius would result in finer heading resolution and increase consideration of straight paths, however experimentally we concluded that the added computational cost exceeds the gain of the radii greater than two. Fig. 3c shows our heading discretization: black lines indicate the uniform part, and grey lines show the non-uniform part.

Motion alternatives were expressed as polynomial steering (curvature) functions of cubic order:

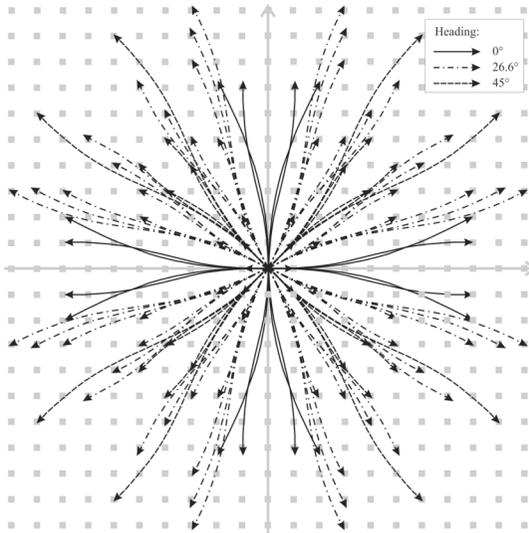


Fig. 4. An Example Control Set. Only three sets of paths with initial heading of 0 , 26.6 and 45° are specified; all others are obtained by reflection around x - and y -axes, and the two diagonals.

$$\kappa(s) = a + bs + cs^2 + ds^3$$

Such functions possess exactly the 5 degrees of freedom required to join any two poses with arbitrary initial and terminal curvatures [10]. An example control set that results is depicted in Fig. 4. Note that the shortest paths (e.g. straight up of length 1) are present in the control set, but not immediately visible.

5 Motion Planning Using Control Sets

The state lattice now possesses the properties necessary to express nonholonomic motion planning as graph search. A cost-map can be overlaid on the lattice to represent obstacles or other criteria with respect to which we want to find an optimum obstacle-free motion plan (energy, slope hazard, etc.). We also assume a path sampling procedure that returns the cost of traversal of a path given the cost of cells spanned by the path. The control set can be viewed as the neighborhood that is expanded when a particular lattice node is considered. Its minimality property that we were seeking is important to making the search in the lattice efficient. We should note that by virtue of containing all feasible motions, the lattice is a cyclic graph. Any standard systematic heuristic graph search algorithm can be applied. In this manner the state lattice can be considered a roadmap, in which the cost of traversal is considered during the search.

5.1 Estimating the Search Heuristic

A key component of heuristic search (e.g. A* implemented for this paper) is calculating the heuristic, an estimate of the cost to travel from any node in the lattice to the goal. We begin with the discussion of the issues involved in estimating the heuristic for nonholonomic vehicles and arrive at a heuristic estimation scheme, based on pre-computing a look-up table, that produces very accurate estimates, thereby improving search efficiency considerably.

The heuristic must not be an over-estimate of the true cost in order for it to be admissible (i.e. in order to guarantee that the search algorithm will find the optimal path). Ideally, we would like it to be exactly equal to true cost, such that the search can be correctly guided toward the goal. Typically, it is impossible to use standard distance metrics, e.g. Euclidean, to estimate the nonholonomic heuristic because depending on change in heading from start to goal and on direction to goal, the vehicle may have to execute an n -point maneuver. Consider re-orienting a car 180°: translation can be negligible in Euclidean sense, but the overall length of the maneuver is significant. Using Euclidean distance for such local plans would result in a gross under-estimate of the cost, such that the behavior of A* would approach that of breadth-first search, with an accompanying performance decrease.

It is important to observe that these issues are primarily relevant for local planning. When the distance to goal is much larger than the minimum turning radius of the vehicle, the under-estimation error percentage of Euclidean distance will become small, thus making this metric a viable heuristic option. We this in mind, we propose a hybrid approach to calculating the heuristic: in the close vicinity of the robot, a local estimation procedure that considers the vehicle’s kinematic model is used, whereas in the far range Euclidean distance is sufficiently accurate. Based on experimental studies, we found that a good threshold for switching between local and global heuristics is 10 minimum turning radii.

As we mentioned, it is crucial for the heuristic to be as accurate as possible for high efficiency. However, there is no known closed-form solution for calculating the local heuristic, and obtaining an accurate estimate requires solving the original planning problem. Thus, we defined an off-line pre-processing step to create the heuristic look-up table (LUT). The table is simply a compilation of the costs to travel from the origin to all lattice nodes in its local neighborhood. These costs are determined by running the planner for each possible path endpoints using simply Euclidean distance as heuristic, which is guaranteed to be an under-estimate. LUT generation could be a lengthy process, but it is performed off-line, and the agenda for future work includes developing advanced function approximators to eliminate this pre-processing. The exact values for the path costs provided by the LUT result in the dramatic speed-up of the planner as described in the next section.

5.2 Path Planner Results

In order to quantify the performance of the present path planner, we undertook a simulation study that included performing a statistically significant number of planning experiments, where initial and final path configurations were chosen at random. It was confirmed that the planner built using the control set generated in Section 4.2 for the robots in [4] is very efficient: it performs as efficiently as basic grid search. In fact, for over 90% of path planning queries, our method performs even faster than grid search. The significant result here is that this method generates optimal nonholonomic paths with no post-processing, yet can perform better than the classical grid search, the archetype of efficiency in path planning. We believe that the reason for this significant speed-up is twofold. The primitive paths can span multiple grid cells, such that by choosing a primitive, the planner may “jump” ahead, while grid search still considers one cell after another. Besides, the accurate heuristic as provided by the LUT was shown to reduce the number of required search iterations considerably.

In Figure 5 we present the timing results of our planner by considering the toughest local planning scenarios: the final state (goal) is close to the initial state and exhibits significant change in heading and direction to goal. The figure shows the results of over 1000 timing experiments for both nonholonomic

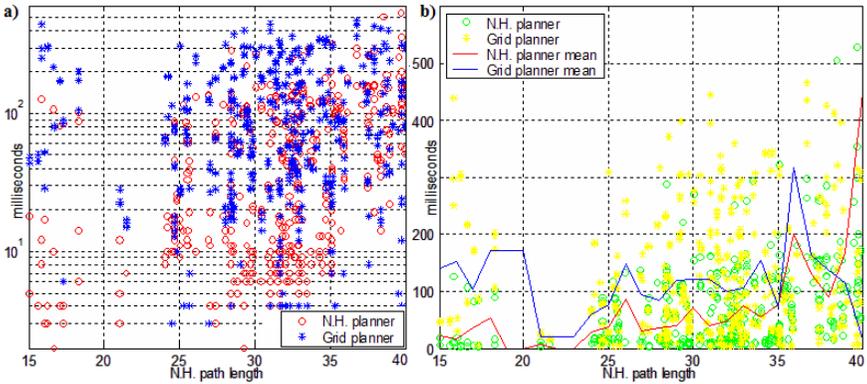


Fig. 5. Run-time results of our nonholonomic (N.H.) path planner (red datapoints) in comparison with basic grid search (blue datapoints). Vertical axis is the time of plan generation, and horizontal axis is the length of the nonholonomic paths. a) Runtimes for both nonholonomic and grid search on semi-log scale. b) Average runtimes superimposed on the same plot on linear scale.

path planner and grid search. For each experiment, a goal Q_f was chosen randomly such that the Euclidean distance between initial state and goal was the same. In this manner, the grid search had roughly the same amount of work to do, whereas nonholonomic path planner’s job could vary significantly depending on changes in orientation between initial and final states. The length of the resulting nonholonomic plan is roughly indicative of that complexity, and so the horizontal axis (in units of cell size) is intended to capture increasing nonholonomic planning complexity. Figure 5a shows the runtime versus nonholonomic path length (i.e. “complexity”) per planning query, plotted on the semi-log scale. Nonholonomic planner runtimes are denoted with circles, and the grid search runtimes – with stars. Even though the plot looks rather busy, the clustering of circles below the stars is clearly visible, indicating that on average nonholonomic planner ran faster in the same experiment (i.e. a choice of path endpoints). This trend is easier to see in Figure 5, where we superimposed the mean of runtime for both planners. The two solid lines (red for nonholonomic, and blue for grid search) clearly show that nonholonomic planner on average takes less time. The balance tilts in favor of grid search only at the right-most end of the horizontal axis, i.e. for highest planning complexity. Thus, our path planner is clearly very efficient and can compute most planning queries in less than 100ms, which deems it useful for real-time applications.

6 Applications

We discuss several important mobile robotics applications that could greatly benefit from the constrained motion planning approach presented herein. The

constrained motion planner presented here was successfully implemented on the DARPA PerceptOR program [4]. This planner guided a car-like all-terrain vehicle in its exploration of natural, often cluttered, environments. The proposed planner exhibited great performance as a special behavior that was invoked to guide the vehicle out of natural cul-de-sacs. Fig. 6b depicts an example motion plan that was generated by the vehicle on-line. The grayscale portion in the figure represents the cost-map, pink indicates obstacles, and orange is the area of unknown cost. Yellow line represents the generated plan. With this technology, the PerceptOR vehicles exhibited kilometers of autonomous traverse in very difficult natural terrain (Fig. 6a).

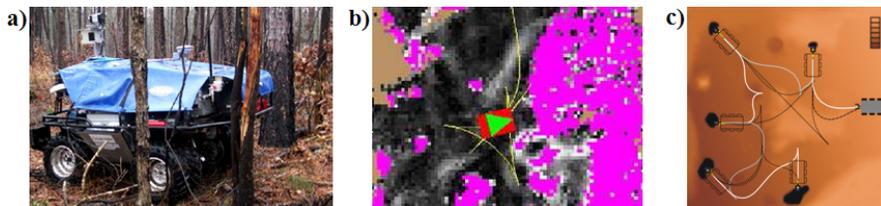


Fig. 6. Example applications: a) Robot navigation in natural cluttered environments (DARPA PerceptOR). b) A nonholonomic path computed in a natural cul-de-sac. c) Planetary rover instrument placement problem. The rover must approach five science objects at specified heading in cluttered environment on the slope of a crater.

Another important application for which the presented motion planner is suited especially well is rover navigation for space exploration. The rover instrument placement task is known to be a difficult problem both from the standpoint of motion planning and execution (see Figure 6c). The significant communication time lag is an important consideration prompting quick progress in rover autonomy. Very rough terrain and considerable wheel slip on loose terrain require an approach that can consider the model of rover motion as accurately as possible, as well as take into account the peculiarities of the terrain as it is being discovered.

Our method of motion planning is well suited for this application because it addresses all of the above issues. The inverse trajectory generator used in this approach [10] can use any kinematic rover model whatsoever, and therefore any generated path is inherently executable by the rover under consideration. The flexibility of using any cost-map, overlaid over the state lattice (implicitly represented through using control sets), enables this planner to consider an arbitrary definition of obstacles in terms of the map cost: both binary (e.g. rocks) and variable (e.g. slopes as high-cost, yet traversable). Moreover, dynamics analysis can be made to “label” regions of steep slopes or very loose terrain as untraversable.

An added benefit to specifying paths as continuous curvature functions is the possibility to define velocity planning quite easily. By defining a maxi-

imum desirable angular velocity of a vehicle, it is straight-forward to compute the maximum translational velocity as a function of path curvature. In this fashion, our path planner can become a trajectory planner with this simple velocity planning post-processing step.

7 Conclusions and Future Work

This work has proposed a generative formalism for the construction of discrete control sets for constrained motion planning. The inherent encoding of constraints in the resulting representation re-renders the problem of motion planning in terms of unconstrained heuristic search. The encoding of constraints is an offline process that does not affect the efficiency of on-line motion planning.

Ongoing work includes designing a motion planner based on dynamic heuristic search which would allow it to consider arbitrary moving obstacles, the extension of trajectory generation to rough terrain, and hierarchical approaches which scale the results to be applicable to kilometers of traverse.

References

1. Latombe J-C (1991) Robot motion planning. Kluwer, Boston
2. Branicky MS, LaValle S, Olson S, Yang L (2001) Quasi-randomized path planning. In: Proc. of the Int. Conf. on Robotics and Automation
3. Frazzoli E, Dahleh MA, and Feron E (2001) Real-time motion planning for agile autonomous vehicles. In: Proc. of the American Control Conference
4. Kelly A et al. (2004) Toward reliable off-road autonomous vehicle operating in challenging environments. In: Proc. of the Int. Symp. on Experimental Robotics
5. Laumond J-P, Sekhavat S and Lamiroux F (1998) Guidelines in nonholonomic motion planning. In: Laumond J-P (ed) Robot motion planning and control. Springer, New York
6. Pancanti S et al. (2004) Motion planning through symbols and lattices. In: Proc. of the Int. Conf. on Robotics and Automation
7. Scheuer A, Laugier Ch (1998) Planning sub-optimal and continuous-curvature paths for car-like robots. In: Proc. of the Int. Conf. on Robotics and Automation
8. Wang D, Feng Q (2001) Trajectory planning for a four-wheel-steering vehicle. In: Proc. of the Int. Conf. on Robotics and Automation
9. Hsu D, Kindel R, Latombe J-C and Rock S (2002) Randomized kinodynamic motion planning with moving obstacles. Int. J. of Robotics Research 21:233-255
10. Kelly A and Nagy B (2003) Reactive nonholonomic trajectory generation via parametric optimal control. Int. J. of Robotics Research 22:583-601
11. LaValle S, Branicky M and Lindemann S (2004) On the relationship between classical grid search and probabilistic roadmaps. Int. J. of Robotics Research 23:673-692

Vision-Based Grasping Points Determination by Multifingered Hands

Madjid Boudaba¹, Alicia Casals², Dirk Osswald³, and Heinz Woern³

¹ TES Electronic Solutions GmbH, Zettachring 8. 70567 Stuttgart, Germany
madjid.boudaba@tesbv.com

² Automatic Control and Computer Engineering Dpt.(ESAII), Technical
University of Catalonia. Campus sud-edif.U, 08028 Barcelona, Spain
alicia.casals@upc.es

³ Institute of Process Control and Robotics (IPR), University of Karlsruhe.
Engle-Bunte-Ring 8-Gebaeude 40.28, 76131 Karlsruhe, Germany
osswald@ira.uka.de, woern@ira.uka.de

Summary. This paper discusses some issues for generating points of contact on object grasping by multifingered robot hands. To address these issues, we present a general algorithm based on computer vision techniques for determining grasping points through a sequence of processes: (1) object's visual features, we apply some algorithms for extracting vertices, edges, object's contours, (3) modeling the point of contact by a bounded polytope, (3) based on these features, the developed algorithm starts by analysing the object's contour to generate a set of contact points that guarantee the force-closure grasps condition. Finally, we briefly describe some experiments on a humanoid robot with a stereo camera head and an anthropomorphic robot hand within the "Center of Excellence on Humanoid Robots: Learning and co-operating Systems" at the University of Karlsruhe and the Forschungszentrum Karlsruhe.

Keywords: Vision system, Points of contacts, Force-closure, Grasping, Linear programming implementation.

1 Introduction

Grasping by a multifingered robot hands has been an active research area in the last years. Several important issues including grasp planning, manipulation and stability analysis has been done. We refer to [1] for a general survey on grasping and to [2-3] for the theory of grasping. Much relevant work has been done in the field of grasping analysis and grasping synthesis referring to properties such as equilibrium, force-closure grasp, form closure [4-14]. Most of these researches assume that the geometry of the object to be grasped is known and the positions of the contact points are estimated based on the

geometrical constraints of the gripper. These assumptions reduce the complexity of the mathematical model of the grasping. However, much less work has been done in generating possible grasps from unknown objects. The complete grasping process requires multiple type of sensors information such as visual information or tactile (force) sensor information, and so, the grasping process should be controlled by fusing this sensory information. This requirement seems too difficult for practical applications. To overcome this problem, the authors propose different systems of sensors. In [15-16] suggested the feasibility of detecting the contact location directly by a tactile sensor instead of estimating it, using the geometrical model of the object. In [17], a vision system based on sensor fusion for grasping has been proposed, which has a parallel processing architecture. The system provides higher resolution with 1ms cycle time, being this rate within the range required for the visual servoing applications. In the most of these researches, the main concern is the detection of the contact location between the fingertips and the object, which a priori geometrical information of the object was not necessary. A grasping system suitable of performing various tasks under different operating condition is currently in the development stage. The adaptation of these theoretical techniques for real grasping objects applications is a subject of actual investigation. In this paper we address the problem of grasping unknown objects by defining a procedure based on the following three phases:

1. Visual information phase. To provide a description of the objects grasping, a set of visual features (including object size, center of mass, object's boundary, and main axis for orientation) is stored into a model graph, which preserves the topological relationships between the features.
2. Grasp points determination phase. A graphical method is used to generate a group of grasping points on the boundary of the object. Then a set of geometrical functions is analysed to find a feasible solution for grasping. There are many other methods proposed for the planning stage [7-10].
3. Grasp decision phase: This is characterized by the stability of the contact between fingers and object. To achieve this, the friction constraints of all contact points must be satisfied by finding an equilibrium between internal and external forces (see also [13] for this topic). Fig. 1 shows the experimental setup. It contains, a stereo camera head, a humanoid robot arm, and an anthropomorphic robot hand.

The rest of this paper is organised as follows: Section 2 gives some mathematical preliminaries for grasping in this direction. In section 3, a vision system framework is presented. The vision system is divided into two parts: the first part concerning to 2D grasping and the second part concerning 3D grasping. In section 4, we will describe some procedures we are using to compute feasible grasping region. In section 5, we will present the experimental results with conclusion.

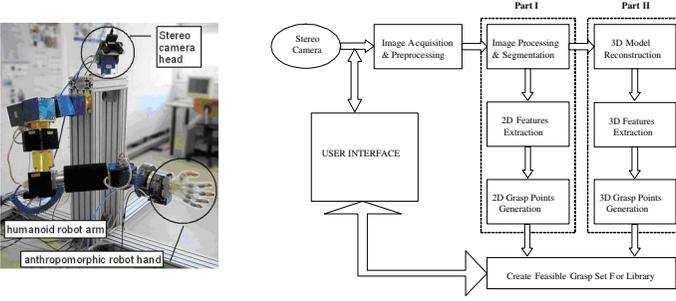


Fig. 1. The Vision System.(left side)Experimental setup with a stereo camera head, a humanoid robot arm(7DOF) and an anthropomorphic robot hand(10 DOF). (right side)Visual Processing PartI and PartII: General Flow Diagram.

2 Mathematical Preliminaries

We consider a k -hard-fingered robot hand is grasping a rigid polygonal object in a 2D workspace, as illustrated in Fig. 2. Since any curved planar object can be approximated to a polygon with any required degree of accuracy, we assume that an object to be grasped has a polygonal shape. We assume that k -fingers are positioned along n given different edges respectively. To hold the object and balance any external forces and moments within each edge, a finger i must apply a force f_i to the object, called grasping force. To assure non-slipping at the contact point p_i , the grasping force f_i must lie inside the friction sector defined by f_{li} and f_{ri} and be centered about the internal surface normal at contact point with half-angle α_i . If it lies inside the friction sector, the grasping force f_i can be expressed as a positive linear combination of these two edges vectors: $f_i = \mu_{ri}f_{ri} + \mu_{li}f_{li}$, with the coefficients $\mu_{ri}, \mu_{li} \geq 0$. The force f_i and the moment m_i acting on the object can be combined into wrench $w_i = (f_i, m_i)^T$, where $m_i = (p_i, f_i)$. By linearity, a contact wrench w_i , generated by f_i can be expressed as the same positive linear combination of the edge wrenches: $w_i = \lambda_{ri}w_{ri} + \lambda_{li}w_{li}$, where w_{ri} and w_{li} are called primitive contact wrenches. The net wrench applied to the object by k -fingers is

$$\mathbf{w}_{net} = \sum_{i=1}^k (\lambda_{ri} \mathbf{w}_{ri} + \lambda_{li} \mathbf{w}_{li}) = W\lambda \quad (1)$$

where $W = [w_{r1}w_{l1}, \dots, w_{rk}w_{lk}] \in R^{3 \times 2k}$ and $\lambda = [\lambda_{r1}\lambda_{l1}, \dots, \lambda_{rk}\lambda_{lk}] \in R^{1 \times 2k}$ are called wrench matrix and column vector, respectively. For more details in this topic, see also [1].

2.1 Modeling the Point of Contact

A point of contact with friction (sometimes referred to as a hard-finger) as defined previously, imposes non linear constraints on the force inside of its

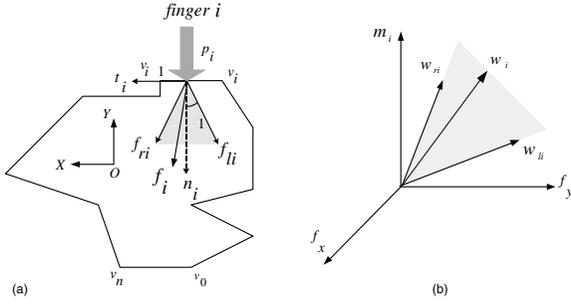


Fig. 2. Polygonal object to be grasped with a hard-finger. (a) a force f_i applied within vertices v_i and v_{i+1} . It lies inside the friction cone defined by its edge vectors f_{li} and f_{ri} . (b) The corresponding wrench space of f_i , $w_i = (f_i, m_i) \in R^3$.

friction cones. In this subsection, we simplify the problem by modeling the friction cones as a convex polytopes using the theory of polyhedral convex cones attributed to Goldman and Tucker [19]. In order to construct the convex polytope from the primitive contact wrenches, the following theorem states that a polyhedral convex cone (PCC) can be generated by a set of basic directional vectors.

Theorem 1. *A convex cone is a polyhedral if and only if it is finitely generated, that is, the cone is generated by a finite number of vectors v_1, v_2, \dots, v_m :*

$$C = \left\{ \mathbf{v}_i \in R^n : \sum_{i=1}^m \alpha_i \mathbf{v}_i, \alpha_i \geq 0 \right\} \tag{2}$$

where the coefficients α_i are all non negative. Since vectors v_i through v_m span the cone, we write (2) simply by $C = \text{span} \{v_1, v_2, \dots, v_m\}$. The cone spanned by a set of vectors is the set of all nonnegative linear combinations of its vectors. A proof of this theorem can be found in [24]. Given a polyhedral convex set C , let $\text{vert}(q) = \{v_1, v_2, \dots, v_m\}$ stand for vertices of a polytope q , while $\text{face}(q) = \{F_1, \dots, F_M\}$ denotes its faces. In the plane, a cone has the appearance as shown in Fig. 3(b). This means that we can reduce the number of cone sides, m to one face. Let's denote by q , the convex polytope of a modelled cone, and $\{v_1, v_2, v_3\}$ its three vertices. We can define such polytope as

$$\mathbf{q} = \left\{ \mathbf{x} \in R^n \mid x = \sum_{i=1}^{v_q} \delta_i \mathbf{v}_i : 0 \leq \delta_i \leq 1, \sum_{i=1}^{v_q} \delta_i = 1 \right\} \tag{3}$$

where v_i denotes the i -th vertex of q , and v_q is the total number of vertices. $n=2$ in the case of a 2D plane. Finally, we denote the intersection of a set of polytope cones (3) by

$$C_{\mathbf{q}_1}^{\mathbf{q}_3} = \text{cone}(\mathbf{q}_1) \cap \text{cone}(\mathbf{q}_2) \cap \text{cone}(\mathbf{q}_3) \tag{4}$$

The set $\mathbf{C}_{q_1}^{qs}$ is either empty or a convex polytope comes from the properties of convex polytope intersection. Since we are interested by the bounded convex

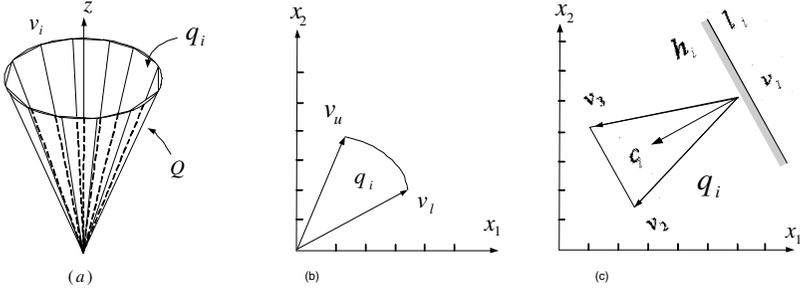


Fig. 3. Model contact: (a) Friction cone. (b) A representation of a bounded polyhedral cone in the plane (x_1, x_2) . Both vectors v_l and v_u span the cone on its lower/upper side. (c) Convex polytope and its extreme vertices. Pointed vertex v_1 joins the line segment at its midpoint and c_i is the gradient of such segment that lies inside the cone.

polytope, we know by definition that one of the its extreme points is an optimal solution. Let v_i be a pointed vertex of q , let $l_i(\phi)$ be the line through v_i that makes an angle ϕ with the positive x -axis, and h_i be the half-plane lying on $l_i(\phi)$. When ϕ increases from 0 to 2π , the line $l_i(\phi)$ (also q) rotates around v_i (see Fig. 3(c-d)). From this definition, we can suggest an observation that, if a given extreme vertex is not an optimal solution, it is possible to move along its line. To represent the vertex, v_i on the line, we introduce a scalar parameter u_i , a unit direction t_i , and an end-vertex v_{i0} so that $v_i = v_{i0} + u_i t_i$ where u_i is constrained by $0 \leq u_i \leq l_i$. The equation

$$\mathbf{c}_i^T \mathbf{x}_i, i = 1, \dots, k \quad (5)$$

represents the line containing the vertex v_i and k is the number of constraints. We denote by $c_i = [c_{ix}, c_{iy}] \in \mathbf{R}^2$, a vector of the first partial derivatives of (5), called the gradient, and it is orthogonal to the line as shown in Fig. 3(c-d) where the gradient is pointed inside of P . In Fig. 3(d), we have changed c_i to $c_i + tu_i$ where u_i is an u_p -vector varies from the lower to the upper side of the spanned cone, and t is a parameter that increases from its initial value 0 to some upper bound \bar{t} . This allows us to check whether the optimal solution remains for all c_i in the cone spanned by v_2 and v_3 . In section 4, we give a detailed procedure to refine the algorithm for (3) and how to integrate it into a whole algorithm for computing force-closure grasps.

2.2 Force-Closure Grasps

Our results on force-closure grasps are based on the following characterization of the force-closure in [4-10]. We deal with our previous definitions that the

full effect of the force and moment applied to an object is a wrench vector such as the one given in (1), where each column of the matrix is described as a three-dimensional wrench space. Using (1), given a set of k -point contacts on an object, the corresponding wrenches w_1, w_2, \dots, w_k are said to achieve force-closure, if the origin of the wrench space R^3 lies in the interior of the convex hull of the primitive contact wrenches [3] Let us rewrite (1) as a convex hull defined by

$$\mathbf{co}(\mathbf{W}) = \left\{ \sum_{i=1}^{2k} \lambda_i \mathbf{w}_i \mid \lambda_i \geq 0, \sum_{i=1}^{2k} \lambda_i = 1, \mathbf{w}_i \in W \right\} \quad (6)$$

where the superscript $2k$ is a pair of primitive wrenches as defined in the wrench matrix. Equation (6) means that the convex hull of $2k$ -primitive wrenches contains strictly the origin of R^3 . The construction of the force-closure grasps now becomes the problem of finding contact location on the boundary of the object. When a fingertip makes a contact on an edge (as shown in Fig. 2), the force direction can be in any direction located inside the friction cone with a friction coefficient, $\mu = \arctan(\alpha)$. To adapt to our previous definitions, we restrict our consideration to fingertips that make contact on an edge, and only edge forces delimiting the friction cone are considered.

3 Vision System Framework

We are currently developing a vision system and its application to two and three-dimensional object grasping which are corresponding to the part I and II, respectively (see Fig. 1). The vision system is based on STH-MD1-C digital stereo head with an IEEE 1394 (Firewire) bus interface and a SRI's Small Vision System (SVS) software for calibration and stereo correlation. The architecture of the whole system is organized into several modules implementing image processing, grasp generation and storage, which are embedded in a distributed object communication framework. For its complexity, the flow diagram of visual processing has been divided into two parts. The first part provides details of 2D object grasping. The second part is dedicated to 3D object grasping using visual information retrieval. As shown in Fig. 1, the image acquisition primarily aims at the conversion of visual information to electrical signals, suitable for computer interfacing. Then, the incoming image is subjected to processing having in mind two purposes: (1) removal of image noise via low-pass filtering by using Gaussian filters due to its computational simplicity and (2) extraction of prominent edges via high-pass filtering by using the Sobel operator. This information is finally used to group pixels into lines, or any other edge primitive (circles, contours, etc). This is the basis of the extensively used Canny edge detector. So, the basic step is to identify the main pixels that may preserve the object shape. As we are visually determining grasping points, the following subsections provide some details of what we need for our approach.

3.1 Extraction of Visual Features

A set of visual features are extracted from the left and right stereo cameras and grouped into a compact representation of the object, which preserves the topological relationships between features. As we are mainly interested on the object's contour, the pixels that belong to the object's boundary are sorted by ascending order of the rows. Let w_x and h_y be the width and height of the image in the plane, respectively. Then the upper left corner pixel coordinate is $[0, 0]^T$ and the lower right pixel coordinate vector is $[h_y - 1, w_x - 1]^T$. We denote by f a function regrouping parameters of visual features together and it is defined by

$$\mathbf{f} = \{\mathbf{p}, \mathbf{r}, \phi, \mathbf{com}\} \tag{7}$$

where p is a list of points with $p_i=(x_i, y_i)$ are the coordinates of a point i , r_i is the distance of p_i to the center of mass, $com=(x_c, y_c)$, and ϕ is the slope value connecting each pixel with the center of mass. Additional to the local features determined above, an algorithm for contour following is integrated. This algorithm follows the object's boundary from a starting point determined previously and goes counter-clockwise around the contour by ordering successively its vertices/edge points into a double linked list. The algorithm stops when the starting point is reached for the second time. The aim of this stage is to determine that all vertices/edges belong to the object's boundary which we are needed further for the determination of the grasping points position.

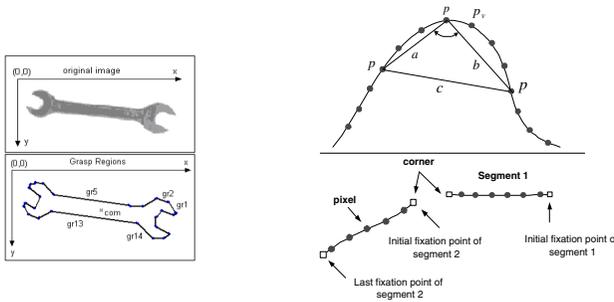


Fig. 4. Visual features of an object grasping: (top left) An original industrial object. (bottom left) The object's contour is represented by a set of segments separated by dot points and are given in counterclockwise direction. The center of mass of the object is indicated by sign (x) and com. (top right) Curves fitting and merging segments (bottom right) process applied to the object's contour. In each curve point p , a variable triangle (p^-, p, p^+) is defined. The admissible triangle is then checked by the following conditions: $d_{min} \leq |p - p^-|$, $d_{min} \leq |p - p^+|$, $\alpha \leq \alpha_{max}$, where $|p - p^-| = a$, $|p - p^+| = b$, and $\alpha = \arccos(a^2 + b^2 - c^2) / 2ab$ is the opening angle of the triangle.

3.2 Edge Linking and Segment Fitting

It was already mentioned that the contour of the object is represented by a list of consecutive edge points $p_i=(x_i, y_i)$. Thus, it can be considered as a 2D function in the Euclidian space. The edge linking list is the result that forms a list of connected edge points from a binary edge image. Let $elist=\{e_1, e_2, \dots, e_n\}$ be the list of n edges points belonging to the object's boundary with $e_i=(x_i, y_i)$ are the coordinates of an edge point i . The aim of this stage is to determine all salient contour segments that preserve the shape of contour. The curve fitting (as shown in Fig. 4 (b)(top)) describes the process of finding a minimum set of curve segments to approximate the object's contour to a set of line segments with minimum distortion. Once the line segments have been approximated, the merging method (as shown in Fig. 4(b)(bottom)) is used to merge two lines segment that satisfied the merging threshold. The final result of the algorithm is a list of consecutive line segments with a specified tolerance which preserve the object's contour. We define such list by $slist=\{s_1, s_2, \dots, s_m\}$ where a segment s_i is defined by its ending vertices $v_i=(x_i, y_i)$ and $v_{i+1}=(x_{i+1}, y_{i+1})$. m is the number of segments. Let $l_i=\{v_i, v_{i+1}, l_{gi}, d_i, \phi_i\}$ be a function defining the parameter of a segment, s_i , where l_{gi} is its center, d_i its distance to the center of mass, and ϕ_i its orientation from x-axis. Later, in the next section, we use this function for computing a feasible solution of grasping. Once the object's contour has been approximated by line segments, the compact representation of the contour of the object can be given by

$$B = (\mathbf{vlist}, \mathbf{slist}) \in R^2 \tag{8}$$

where B is a compact representation of the object's contour in 2D. We will integrate (8) in our algorithm for computing the feasible regions of grasps.

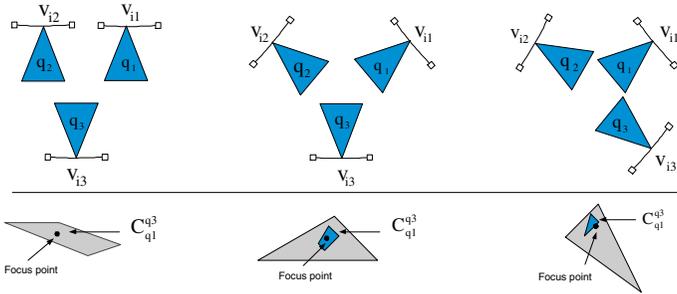


Fig. 5. 2D Grasping configuration with three-fingers and its corresponding convex polytopes. (a) Focus point of the feasible solution of (9), it does not include the area of friction of the intersection cone (4), the grasp is not achieved. The grasping is achieved in (b) and (c). (b) Focus point includes the feasible solution of (9) and area of friction cones intersection. In (c), the area of friction cones intersection is on the border of the feasible solution.

3.3 Grasping Points Determination

Once the grasping regions have been determined on the object contour, a heuristic algorithm is used to determine the grasping points where the robot fingers could be positioned. It takes as input the set of grasping regions as illustrated in the previous section and given by (8). The output is a list of k -tuple grasping points. Each region is characterized by containing one placed finger, so we have to find which triplet of regions could allow a k -finger grasp. Mainly two steps are performed. In the first step the algorithm picks the k -tuple elements from the list, and in the second it checks whether or not a candidate grasp point will satisfy the condition of force-closure grasps. Taking into account the force-closure grasping criterion, the grasping points determination must satisfy the following conditions:

- The unit normal vectors to the surfaces defined by the grasp regions positively span the plane.
- The convex hull generated by the contact points contains the origin. (See also [3] and [8] for more details).

We use an additional criterion for selecting the best grasp points. The distance between the focus point of a grasp and center of mass is used as a quality parameter to classify the stored grasps from the lower to the upper acceptance measures.

4 The Algorithm

In order to obtain procedures for computing feasible solution for grasping, we deal with our previous analysis of the shape of the object's boundary given by equation (8) which consists of a double linked list of segments and vertices, and by (3) which is the convex polytope for modelling a cone. We first calculate the extreme vertices of the convex polytope. Then we need to fix one vertex as optimal solution on the midpoint of a joining segment. We transform the double linked list of segments to the problem of solving equality and inequality constraints. We call this problem Primal Problem and can be expressed as follows:

$$\mathbf{f}(x) : \mathbf{H}(x) - \mathbf{b} \leq 0, lb \leq u(x) \leq ub \quad (9)$$

where $x = [x_1, x_2]^T$ is the vector coordinates of a point in the plane, d is a real number, $[lb, ub]$ define the lower/upper band criterion. $H = \{h_1, h_2, \dots, h_k\}$ be a linked list of half-planes defined over two variables. Each half-plane in H is defined as a closed convex set in R^2 and it is represented by giving the coordinates of its two endpoints. The intersection of a set of H , $h_k \in H$ is defined by

$$\Omega_{h_1}^{h_k} = \{h_1 \cap h_2, \dots, h_k\} \quad (10)$$

where Ω is a feasible solution of intersecting k half-planes (see Fig. 5, where $k=3$). The feasible solution for common intersection is to find a set among all the points that satisfy the following constraints problem at the same time.

$$\mathbf{h}_i = \{(\mathbf{x}, \mathbf{y}) \mid a_i \mathbf{x} + b_i \mathbf{y}\}, i = 1, \dots, k. \quad (11)$$

where k is the number of constraints. Since each half-plane is convex, and the intersection of two convex is convex, the result of a set of constraints intersection is convex. More details about convex analysis can be found in [18-19]. Since we are interested to the bounded solution, the expected running time we present here is $O(k \log k)$, where k is the number of constraints. The whole process operates as follows:

1. Compute the list of grasp regions: each region is approximated as a straight segment with its associated parameter function. Retain the grasp region that have a size value longer enough than an estimated threshold corresponding to a rayon of the finger (let said, the regions have less than three pixels are omitted).
2. Pick three grasp regions: compute the parametric form of each segment and find the midpoint as starting finger's position on each segment.
3. Model a cone as bounded polytope: by providing its vertices-representation. Fix a vertex as pointed vertex laying at the midpoint of its corresponding grasp region.
4. Polytope cone projection: compute the area of three cone intersection. Check whether the polytope is bounded by returning 1 as output, and, 0 otherwise.
5. Build a system of linear inequalities: compute the area of grasping region and check whether or not the result admits a solution.
6. Force-closure grasp: examine whether or not a feasible region of grasp is a bounded polytope region. If so, stop and save the corresponding grasping configuration. Find the focus point and measure its distance from the center of mass of the object (see Fig. 6). Keep this values into 2D array as parameter measures. We try to find as near as possible the center of mass of the object in the bounded region. This enables to decrease the effect of gravitational and inertial forces during grasping.

The last step of the algorithm consists of selecting the best grasps from a range of valid grasps from lower to upper acceptance measures by using two parameters measure: the distance between the focus point of a feasible grasp region and the center of mass, and the distance between both focus points; area of friction cone and grasp region.

5 Results and Conclusions

We have introduced an approach that combines vision and grasping. Based on the vision, visually determining grasping points is done by transforming the

grasping regions into a geometrical optimization problem. The results shown in Fig. 6 are obtained from applying the software packages in [20] to our Matlab 6.12 programming environment. In order to compute the feasible region of various grasps, we have integrated other linear programming solvers by providing a set of constraints for optimization procedure. Various grasps with three hard-fingers are tested on 2D original object and the feasible solution of grasps are determined by analysing the polytope region of grasps. The focus point inside the polytope convex and its distance from the object's center of mass are two measures used for selecting the best grasps. The most important aspects of our algorithm are how to select the grasping point set and to determine each one step of the grasping process. Three functions, pick(), insert(), and remove() are used. The initialization step picks a first grasping set. The while loop iterates by checking the feasible region of grasps and then by selecting a new candidate of grasp. A build library is used to store valid grasps by the insertion function which inserts a valid candidate grasp into library, while the remove function deletes invalid grasp from the library. The results in this paper shows the potential to combine vision and grasping in a unified way to resemble the dexterity of human manipulation.

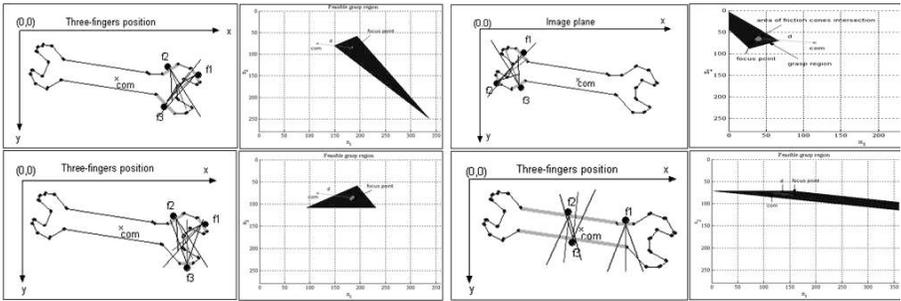


Fig. 6. Object Grasping by three fingers. These are some results obtained and positioned on their corresponding segment (shaded segments). The force-closure grasp is satisfied when the polytope region of friction cones intersection is not empty as shown in (a), (b) and (c). In (d), the force-closure grasp is not satisfied. The focus point is included into the both polytopes region as shown in (a-c). The best grasp shows in (b) is selected from its corresponding parameters measure.

The second part of our visual processing: General flow diagram will be the future work for generating 3D grasps on unknown objects includes implementation on a humanoid robot with a stereo camera head and an anthropomorphic robot hand (as shown Fig. 1).

References

1. Shimoga K. B, (1996) Robot grasp synthesis algorithms: A survey, *Int. Journal of Robotics Research*, 15(3):230–266
2. Murray R. M, Li Z, and Sastry S. S (1994) *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Boca Raton, New York
3. Mishra B, Schwartz J. T, Sharir M, (1987) On the existence and synthesis of multifinger positive grips, *Algorithmica* (3)
4. Martin B, Canny J, (1994) Easily Computable Optimum Grasps in 2-D and 3-D 2, *IEEE Int. Conf. on Robotics and Automation*, 739–747, San Diego, CA
5. Kerr J. and Roth B, (1986) Analysis of multifingered hands, *Int. J. of Robotics Research*, 4(4):3–17
6. Nguyen V. D, (1988) Constructing force-closure grasps, *International Journal of Robotics Research*, 7(3):3–16
7. Ferrari C, Canny J. F, (1992) Planning Optimal Grasps, *Int. Conf. on Robotics and Automation*, 2290–2295
8. Ponce J, Sullivan S, Boissonnat J. D, Merlet J. P, 8(1993) On characterizing and computing three- and four-finger force-closure grasps of polyhedral objects,” *International Conference in Robotics and Automation*.
9. Boudaba M, Casals A, (2000) Robot Grasps: A survey and development of a grasping procedure, Technical report, ESAII-RR-00-15, Dept. ESAII, Technical University of Catalonia, Barcelona, Spain
10. Boudaba M, Casals A, (2005) Polyhedral Convex Cones for Computing Feasible Grasping Regions from Vision, 6th IEEE CIRA Symposium, Espoo, Finland
11. Brost R. C, (1991) Analysis and planning of planar manipulation tasks, Ph.D. thesis. Carnegie Mellon University. School of Computer Science
12. Liu Y. H, (1998) Computing n-finger force-closure grasps on polygonal objects, *Proc. IEEE Int. Conf. on Robotics and Automation*, 2734–2739
13. Hirai S, Asada H, (1993) Kinematics and Statics of Manipulation using the Theory of Polyhedral Convex cones, *Int. J. of Robotics Research*, 12(5):434–447.
14. Ishii I, Nakabo Y, Ishikawa M, (1996) Target tracking algorithm for 1ms visual feedback system using massively parallel processing vision, *Int. Conf. on Robotics and Automation*, 2309–2314.
15. Maekawa H, K. Tanie H. K, Komoriya K, (1995) Tactile sensor based manipulation of an unknown object by a multifingered hand with rolling contact, *IEEE ICRA*, 743–750
16. Yoshimi B, Allen P, (1998) Visual control of grasping,” In D. Kriegman, G. Hagar, and S. Morse, Editors, *The Confluence of Vision and Control*, 195–209, Springer-Verlag.
17. Namiki A, Nakabo Y, Ishii I, Ishikawa M, (1999) High speed grasping using visual and force feedback, *Int. Conf. on Robotics and Automation*, 3195–3200.
18. de Berg M, Van Kreveld M, Overmars M, Schwarzkopf O, (1997) *Computational Geometry: Algorithms and Applications*, 2nd ed., Springer-Verlag.
19. Goldman A. J, Tucker A. W, (1956) Polyhedral Convex Cones, in *Linear Inequalities and Related Systems*, *Annals of Math. Studies*, Princeton, 38:19–40,
20. Kvasnica M, Grieder P, Boato M, Christophersen F. J, (2004) MPT2.0, <http://control.ee.eth.z/mpt>. User’s Guide, Swiss Federal Institute.

Embodied Social Interaction for Service Robots in Hallway Environments

Elena Pacchierotti, Henrik I. Christensen, and Patric Jensfelt

Centre for Autonomous Systems, Swedish Royal Institute of Technology
SE-100 44 Stockholm, Sweden {elena-p,hic,patric}@nada.kth.se

Summary. A key aspect of service robotics for everyday use is the motion in close proximity to humans. It is essential that the robot exhibits a behavior that signals safety of motion and awareness of the persons in the environment. To achieve this, there is a need to define control strategies that are perceived as socially acceptable by users that are not familiar with robots. In this paper a system for navigation in a hallway is presented, in which the rules of proxemics are used to define the interaction strategies. The experimental results show the contribution to the establishment of effective spatial interaction patterns between the robot and a person.

Keywords: Service robotics, hallway navigation, robot-human interaction, embodied social interaction.

1 Introduction

Robots are gradually entering our daily lives to take over chores that we would like to be without and for assistance to elderly and handicapped. Already today we have more than 1.000.000 robots in domestic use (Karlsson, 2004). In terms of (semi-) professional use we are also starting to see robots as courier services, and as part of flexible AGV-systems.

As robots start to enter into daily lives either in homes or as part of our office/factory environment, there is a need to endow the robots with basic social skills. The robot operation must of course be safe, but in addition we expect the robot to interact with people following certain social rules. An example of this is passage of people when encountered in the environment. When people pass each other in a corridor or on the factory floor, certain rules of encounter are obeyed. It is natural to expect that robots, at least, should follow similar rules. This is in particular important when robots interact with users that are inexperienced or have never before met a robot.

Several studies of physical interaction with people have been reported in the literature. Nakauchi & Simmons (2000) report on a system that is to stand

in line for event registration. Here the robot has to detect the end of the line and position itself so as to obey to normal queueing behavior. Althaus et al. (2004) report on a system that is to participate in multi-person interaction as part of a group. It is here important to maintain a suitable distance from the other actors and to form a natural part of the group. Passage of people in a hallway has been reported by Yoda & Shiota (1997); an avoidance algorithm has been developed, based on a human avoidance model, where two separate conditions of a standing and walking person were considered.

In this paper we study the problem of social interaction of a robot with people in a hallway setting and present an algorithm for person passage that, in contrast with the one proposed by Yoda & Shiota (1997), dynamically adapts the robot's behavior to the person's motion patterns. A overall description of the spatial interaction among people during passage is presented in Section 2, and the corresponding control strategy for the robot is presented in Section 3. The implementation of the proposed strategy is described in Section 4. The system has been evaluated in a number of different tests to show its handling of standing and moving people and the corresponding handling of regular obstacles. The experimental results are summarized in Section 5. Finally the main observations, open questions and issues for future research are presented in Section 6.

2 Human Spatial Interaction

Interaction between people has been widely studied both as part of behavioral studies and in psychology. Formal models of interaction go back to the 1960s when one of the most popular models in the literature, the *proxemics* framework, was presented by Hall (1966). The literature on proxemics is rich, but good overviews have been presented by Aiello (1987) and Burgoon et al. (1989). In proxemics the space around a person is divided into 4 categories:

- Intimate: This ranges up to 45 cm from the body and interaction within this space might include physical contact. The interaction is either directly physical such as embracing or private interaction such as whispering.
- Personal: The space is typically 45-120 cm and is used for friendly interaction with family and for highly organized interaction such as waiting in line.
- Social: The range of interaction is here about 1.2-3.5 m and is used for general communication with business associated, and as a separation distance in public spaces such as beaches, bus stops, shopping, etc.
- Public: The public space is beyond 3.5 m and is used for no interaction or in places with general interaction such as the distance between an audience and a speaker.

It is important to realize that the personal space varies significantly with cultural and ethnic background. As an example in Saudi Arabia and Japan the spatial distances to be respected in person-person interaction are much

smaller, than in countries such as the USA and the Netherlands. The passage/encounter among people does not only depend upon the interpersonal distance, but also the relative direction of motion. At the same time there are social conventions of passage that largely follow the patterns of traffic. So while in Japan, UK, Australia, ... the passage in a hallway is to the left of the objects, in most other countries it is to the right.

One could model the personal space for a human as a set of elliptic regions around a person as shown in Figure 1. Video studies of humans in hallways seem to indicate that such a model for our spatial proxemics might be correct Chen et al. (2004).

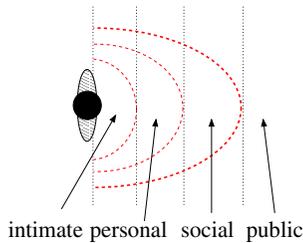


Fig. 1. The interaction zones for people moving through a hallway setting

It would be natural to assume that the robot respects the same physical boundaries as we expect from other people, if the robot has to display some level of “social intelligence”.

3 The Control Strategy

The operation of a robot in a hallway scenario is presented here. Given that proxemics plays an important role in person-person interaction, it is of interest to study if similar rules apply for the interaction between people and robots operating in public spaces. Informally one would expect a robot to give way to a person when an encounter is detected. Normal human walking speed is 1-2 m/s which implies that the avoidance must be initiated early enough to signal that the robot has detected the presence of a person and to indicate its intention to provide safe passage for her/him. In the event of significant clutter the robot should move to the side of the hallway and stop until the person(s) have passed, so as to give way. A number of basic rules for the robot behavior may thus be defined:

1. Upon entering the social space of the person initiative a move to the right (wrt. to the robot reference frame) to signal the person that has been detected.
2. Move to the right to respect a desired distance from the person (if the layout of the hallway allows) while passing the person.

3. Await a return to normal navigation until the person has passed by. A too early return to normal navigation might introduce discomfort on the user's side.

Using the rules of proxemics outlined in Section 2, one would expect the robot to initiate avoidance when the distance is about 3.5m to the person. Given a need for reliable detection, limited dynamics and early warning however, a longer distance for reaction was chosen (6 m). The avoidance behavior is subject to the spatial layout of environment. If the layout is too narrow to enable passage outside of the personal space of the user, as in the case of a corridor, it is considered sufficient for the robot to move to the right as much as it is possible, respecting a safety distance from the walls. The strategy is relatively simple but at the same time it obeys the basic rules of proxemics.

4 An Implementation

The strategies outlined above have been implemented on a Performance PeopleBot from ActivMedia Robotics (Minnie). Minnie is equipped with a SICK laser scanner, sonar sensors and bumpers (see Figure 2).



Fig. 2. The People-Bot system

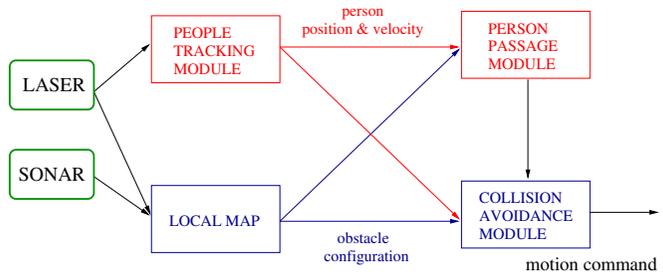


Fig. 3. The overall control system architecture

The system on-board computer runs Linux and uses the Player software (Vaughan et al., 2003) for interfacing the robot sensors and actuators. The main components of the control system are shown in Figure 3.

The laser and sonar data are fed into a local mapping system for obstacle avoidance. In addition the laser scans are fed into a person detection/tracking system. All the software runs in real-time at a rate of 10 Hz. The serial line interface to the SICK scanner runs at a rate of 5 Hz.

The tracking module detects and tracks people in the environment; the laser is mounted on the robot at a height of 33cm from the ground to perform leg detection of the persons. Information about the current position of

the people as well as their velocity is provided. Both the magnitude and the direction of the velocity are important to decide when and how to react. A particle filter, as the one presented by Schulz et al. (2001), is used which can deal with the presence of multiple persons.

The navigation system relies on a local mapper that maintains a list of the closest obstacle points around the robot. Obstacle points are pruned away from the map when they are too far from the robot or when there is a closer obstacle in the same direction. The sonar data are processed through the HIMM algorithm by Borenstein & Koren (1991) before being added to the map. The collision avoidance module can deal with significant amount of clutter but it does not take the motion of the obstacles into account as part of its planning and it does not obey the rules of social interaction. The Nearness Diagrams (ND) method by Minguez & Montano (2004) has been chosen because it is well suited for cluttered environments. The Person Passage module (PP) implements a method for navigating among dynamically changing targets and it is outlined in the next Section. During normal operation the robot drives safely along the corridor toward an externally defined goal. The goal is feed to the collision avoidance module. In parallel the person tracker runs to detect the potential appearance of a person. If a person is detected by the people tracker both the PP and the ND modules are notified. The PP module generates a strategy to pass the person. If, due to the limited width of the corridor the passage would involve entering into the personal space of the person, the ND module will override the generate motion commands and park the vehicle close the wall of the hallway, until the person has passed. Otherwise the generated motion commands are filtered through to the robot.

It is important to underline here some important assumptions that have been made in the implementation. The approach consider the presence of one person at a time; to deal with the simultaneous presence of multiple persons this strategy should be extended. It is assumed that the robot operates in a hallway wide enough to allow the simultaneous passage of the robot and the person; this means that the only impediment to the robot's maneuver is represented by the person behavior (i.e. the person's pattern of motion along the corridor). The presented method aims at achieving a low level control modality whose only competence is to determine a passage maneuver on the right of the person, when it is possible, or to stop the robot otherwise. We believe that it is crucial to stick to this simple set of rules to avoid any ambiguity in the robot behavior. In situations where the method decides to stop the robot, a high level module based on a more complete set of information (localisation of the robot on a global map of the environment, user motion model for person's behavior prediction) could determine alternative motion patterns for the robot.

4.1 Person Passage Method

The Person Passage module has been designed to perform a passage maneuver of a person, according to the previously defined proxemics rules. It operates as follows: as soon as a person is detected at a frontal distance below 6 m, the robot is steered to the right to maintain a desired lateral distance from the the user. If there is not enough space, as might be the case for a narrow corridor, the robot is commanded to move as much to the right to signal to the user that it has seen her/him and lets her/him pass.

A desired trajectory is determined that depends on the relative position and speed of the person and the environment configuration encoded in the local map. The desired trajectory is computed via a cubic spline interpolation. The control points are the current robot configuration (x_0^R, y_0^R) , the desired “passage” configuration (x_P^R, y_P^R) , and the final goal configuration (x^G, y^G) , where x is in the direction of the corridor (see Figure 4).

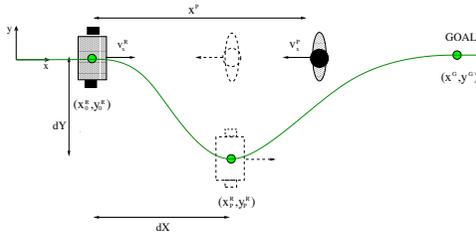


Fig. 4. Desired trajectory for the passage maneuver. The distance of the robot from the person is maximum when it is passing her/him (dashed line)

The control point (x_P^R, y_P^R) determines the passage maneuver, and is computed as follows:

$$x_P^R = x_0^R + dX \tag{1}$$

$$y_P^R = y_0^R + dY \tag{2}$$

The value of dY depends on the lateral distance LD that the robot has to keep from the person:

$$dY = LD + w_R/2 - (y^P - y_0^R) \tag{3}$$

where w_R is the robot’s width and y^P is the person’s y coordinate in the corridor frame. The value of dY may be limited by the free space on the robot right. dX is computed so that the robot maintains the maximum distance from the person when it is passing her/him, according to 4:

$$dX = v_x^R / (v_x^R - v_x^P) \times (x^P - x_0^R) \tag{4}$$

The robot starts the maneuver by clearly turning to the right to signal to the person its intent to pass on the right side, then the maneuver is updated

according to the person's current relative position x^P and velocity v_x^P (see 4), until the person has been completely passed, at which point the robot returns to its original path. The capability to adapt to the changes in the speed of the person is crucial to establish a dynamic interaction between robot and person, as will be shown in Section 5, and represents an important improvement with respect to the work of Yoda & Shiota (1997).

The adopted trajectory following controller takes into account the differential drive kinematics of our robot to define the feed forward command (driving and steering velocity) (Oriolo et al., 2002):

$$v_D(t) = \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \quad (5)$$

$$v_S(t) = \frac{\ddot{y}_d(t)\dot{x}_d(t) - \ddot{x}_d(t)\dot{y}_d(t)}{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \quad (6)$$

where $x_d(t)$ and $y_d(t)$ is the reference trajectory. The controller includes also an error feedback in terms of a proportional and a derivative term.

5 Experimental Results

The system has been evaluated in a number of different situations in the corridors of our institute, which are relatively narrow (2 m wide or less). During the experiments the “test-person” was walking at normal speed, that is around 1 m/s; the average speed of the robot was around 0.6 m/s.

5.1 Person Passage

The experiments show how the system performs in the person passage behavior, adapting to the person speed and direction of motion. Three different cases are here presented.

In the first situation a person is walking at constant speed along the corridor. Figure 5 depicts top-down four different steps of the encounter. The robot starts its course in ND mode. As soon as the robot detects the person at a front distance below 6 m, it starts its maneuver with a turn toward the right (first snapshot). This makes the person feel more comfortable and most people will instinctively move to the right too, to prepare for the passage, as it happens in the second snapshot. As soon as the person has been passed by the robot, the robot resumes its path along the center of the corridor (third and fourth frames). The steering maneuver of the robot results in an effective interaction with the user; to achieve this result, it has been crucial to perform a clear maneuver with a large advance. In the second test (see Figure 6), the person walks along the corridor and then stops. The robot starts its maneuver at the same front distance from the person as before (first frame) but then, detecting that the person has stopped (second frame) it does not turn

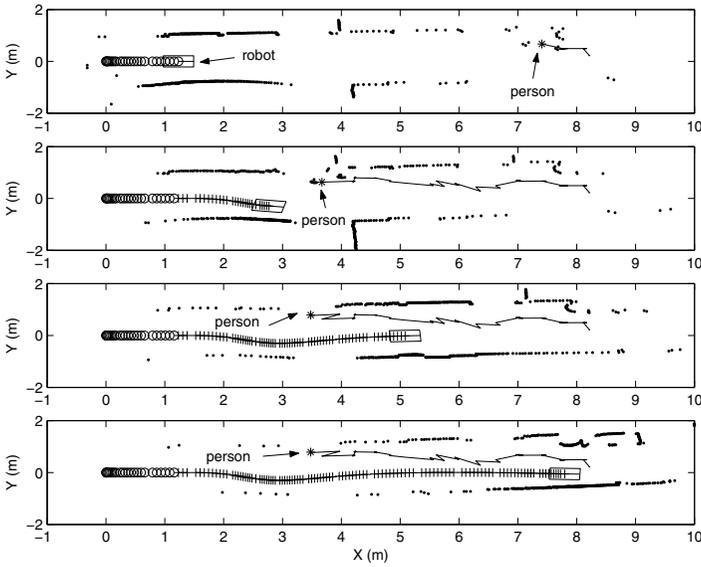


Fig. 5. A person walks along the corridor. The robot trajectory in ND (*circle*) and in PP mode (*plus*) is represented. The person trajectory (*continuous line*) and current position (*star*) together with the current obstacle points on the local map (*dots*) are also shown. The robot steers to the right to pass the person and then resumes its course

toward the center of the corridor but it continues on the right until it has completely passed the person (third frame). Then the robot resumes its path toward the goal (fourth frame). Updating on-line the desired trajectory has allowed the robot to adapt the passage maneuver to the person relative position and velocity. This is a key feature to establish an interaction with the person that perceives the robot operation as safe and “social”. In the third test (see Figure 7), the person is walking along the corridor and then turns to his left to enter in his office. The robot starts a maneuver of passage as before (first and second frames) but then, as soon as it detects the person on the “wrong” side of the corridor, it stops (third frame). Once the person is not detected any more, the robot resumes its path in ND mode (fourth frame). In this situation, the environment layout does not allow the robot to pass the person on the right and a passing maneuver on the left would be perceived by the person as not natural and unsafe, contradicting the social conventions of spatial behavior. In such a situation, it is considered as the best solution for the robot to stop.

5.2 Regular Obstacles Handling

This second set of experiments show how the robot handles regular objects in the environment. A paper bin was placed in the corridor, in the robot path.

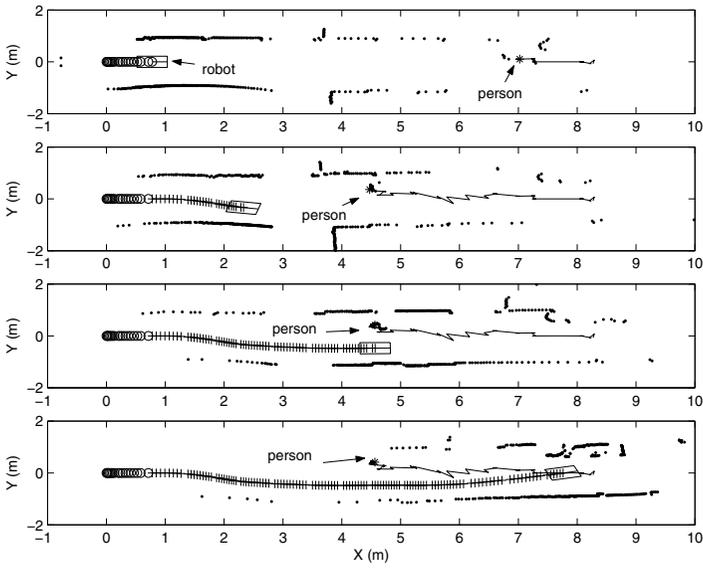


Fig. 6. The person stops. The robot waits until it has passed the person to resume its course on the center of the corridor

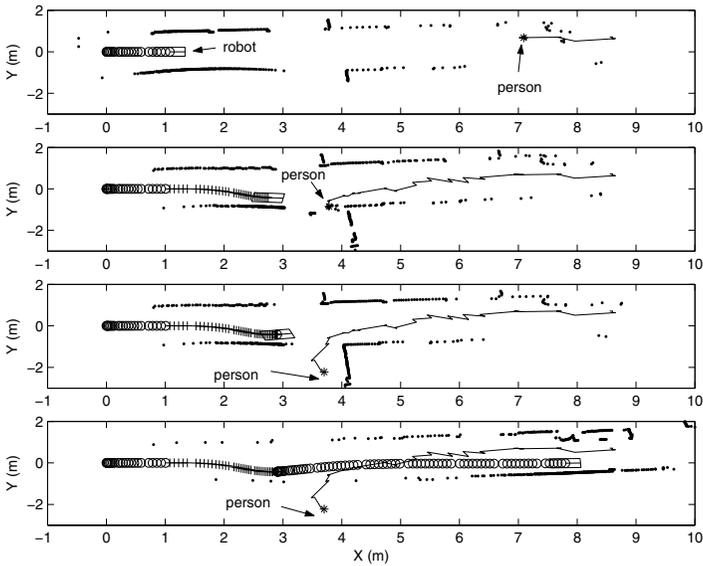


Fig. 7. The person crosses the robot path. The robots stops and wait until the person has disappeared from the field of view of the laser to resume its path in ND mode

The controller was in ND mode with a security distance of 0.6 m, because no persons were around. Two different configurations of the paper bin with respect to the corridor have been considered. In the first situation the bin is on the left of the hallway, close to the wall. The robot circumvents it on the right (see Figure 8, left). This is automatically achieved with the ND because the right is the only free direction). It is important to observe here that ND drives the robot safely around the obstacle but it does not make the robot steer to the side as early as the PP mode does, in presence of a person. A

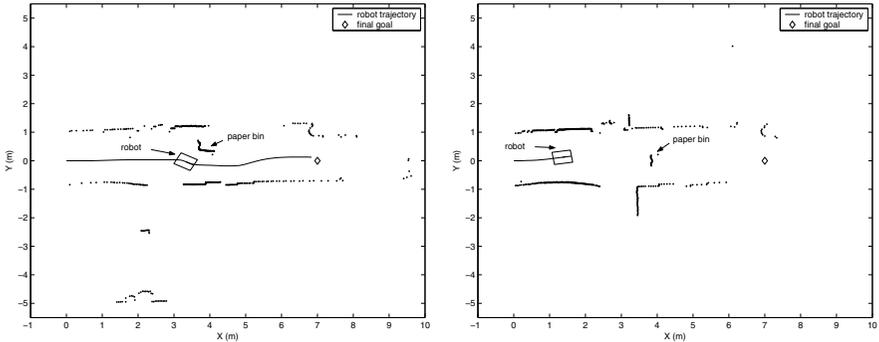


Fig. 8. Regular obstacles handling. On the left, the robot circumvents a paper bin placed on the left of the corridor. On the right, the paper bin is in the center of the corridor, the robot stops

second situation is shown on the right of Figure 8 in which the paper bin has been placed slightly to the right of the center of the hallway (wrt. to the robot). This is a potentially dangerous situation, because the object could be a non-detected person and it would be inappropriate to operate in ND mode, as ND would in most of the cases pass the obstacle on the left. The robot is not allowed to pass and it stops at a distance of 2.5 m from the object.

It may appear a strong measure to stop the robot in the center of the corridor, as in the second situation. But it is important to underline here that we are making the assumption that the corridor should normally be free from obstacles. So, if the robot detects something in the middle of the hallway it should take into account the hypothesis that this object could be a person. In this case the chosen strategy is to stop the robot, because any other attempt to steer (as moving to the side and then stopping) could be perceived, at such short distance (2.5 m), as unsafe and unpredictable by the undetected person.

5.3 Pilot User Study

To fully appreciate the value of such method and to fine-tune it to be socially acceptable there is a need for careful user studies. Some preliminary indications about the method have been achieved in a pilot user study in which 4

subjects have evaluated the acceptability of the robot motion patterns during passage with respect to 3 parameters: the robot speed, the signaling distance at which the robot starts the maneuver and the lateral distance kept from the person during passage (Pacchierotti et al., 2005). Two values of each parameters were tested by the subjects. It was clear from the users feedback that higher speeds were preferred. An explanation for this result is that the robot moves faster to the side; the higher speeds during passage were still not higher than 0.4 m/s so they were never perceived as intimidating. The lower speeds instead, were perceived as less safe or even annoying by the users. The higher value of the signaling distance was highly preferred by all the subjects. Although not necessary to avoid the user, a large signaling distance is important for the robot behavior to be clearly understood. An early maneuver allows the robot to signal its intent, so its behavior is perceived as trustworthy and safe by the user. No clear indication emerged about which value of lateral distance was preferred. The evaluation of this parameter will be addressed more extensively in further studies.

6 Summary/Outlook

As part of human robot interaction there is a need to consider the traditional modalities such as speech, gestures and haptics, but at the same time the spatial interaction should be taken into account. For operation in environments where users might not be familiar with robots this is particularly important as it will be in general assumed that the robot behaves in a manner similar to humans. There is thus a need to transfer these rules into control laws that endow the robot with a “social” spatial behavior.

In this paper the problem of passage of a person in a hallway has been studied and a control strategy has been presented, based on definitions borrowed from proxemics. The operation of the robot has been evaluated in a number of experiments in typical corridor settings which have shown how the introduction of social rules for corridor passage in the robot navigation system can give a contribution to the establishment of effective spatial interaction patterns between a robot and a person.

The hallway passage is merely one of several different behaviors that robots must be endowed with for operation in spaces populated by people. The generalization to other types of environments is an issue of current research.

Acknowledgements

The present research has been sponsored by the Swedish Foundation for Strategic Research (SSF) through its Centre for Autonomous Systems (CAS) and the EU as part of the Integrated Project “CoSy” (FP6-004150-IP). The support is gratefully acknowledged. H. Hüttenrauch and K. Severinson-Eklundh participated in discussions on the interaction strategy.

References

- Aiello, J. R. (1987). Human Spatial Behaviour. In D. Stokels & I. Altman (Eds.), *Handbook of Environmental Psychology*. New York, NY: John Wiley & Sons.
- Althaus, P., Ishiguro, H., Kanda, T., Miyashita, T., & Christensen, H. I. (2004, April). Navigation for human-robot interaction tasks. In *Proc. of the IEEE Int. Conf. on Robotics and Automation* (Vol. 2, p. 1894-1900).
- Borenstein, J., & Koren, Y. (1991, Aug.). Histogramic in-motion mapping for mobile robot obstacle avoidance. *IEEE Trans on Robotics and Automation*, 7(4), 535-539.
- Burgoon, J., Buller, D., & Woodall, W. (1989). *Nonverbal Communication: The Unspoken Dialogue*. New York, NY: Harper & Row.
- Chen, D., Yang, J., & Wactlar, H. D. (2004, October). Towards automatic analysis of social interaction patterns in a nursing home environment from video. In *6th ACM SIGMM International Workshop on Multimedia Information Retrieval* (Vol. Proc of ACM MultiMedia 2004, pp. 283-290). New York, NY.
- Hall, E. (1966). *The Hidden Dimension*. New York: Doubleday.
- Karlsson, J. (2004). *World robotics 2004*. Geneva, CH: United Nations Press/International Federation of Robotics.
- Minguez, J., & Montano, L. (2004, Feb.). Nearness Diagram Navigation (ND): Collision avoidance in troublesome scenarios. *IEEE Trans on Robotics and Automation*, 20(1), 45-57.
- Nakauchi, Y., & Simmons, R. (2000, October). A social robot that stands in line. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (Vol. 1, p. 357-364).
- Oriolo, G., De Luca, A., & Venditelli, M. (2002, November). WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Trans on Control Systems Technology*, 10(6), 835-852.
- Pacchierotti, E., Christensen, H. I., & Jensfelt, P. (2005, August). Human-robot embodied interaction in hallway settings: a pilot user study. In *Proc. of the IEEE Int. Workshop on Robot and Human Interactive Communication*. Nashville, TN.
- Schulz, D., Burgard, W., Fox, D., & Cremers, A. B. (2001, December). Tracking multiple moving objects with a mobile robot. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Kauai, HI.
- Vaughan, R., Gerkey, B., & Howard, A. (2003, Oct.). On device abstraction for portable, reusable robot code. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (pp. 2121-2127). Las Vegas, NV.
- Yoda, M., & Shiotani, Y. (1997, September). The mobile robot which passes a man. In *Proc. of the IEEE Int. Workshop on Robot and Human Interactive Communication* (p. 112-117).

Intentional Motion Online Learning and Prediction*

Dizan Vasquez, Thierry Fraichard, Olivier Aycard, and Christian Laugier

Inria Rhône-Alpes

<http://emotion.inrialpes.fr>

Summary. Motion prediction for objects which are able to decide their trajectory on the basis of a planning or decision process (e.g. humans and robots) is a challenging problem. Most existing approaches operate in two stages: a) learning, which consists in observing the environment in order to identify and model possible motion patterns or plans and b) prediction, which uses the learned plans in order to predict future motions. In existing techniques, learning is performed off-line, hence, it is impossible to refine the existing knowledge on the basis of the new observations obtained during the prediction phase. This paper proposes a novel learning approach which represents plans as Hidden Markov Models and is able to estimate the parameters and structure of those models in an incremental fashion by using the Growing Neural Gas algorithm. Our experiments demonstrate that the technique works in real-time, is able to operate concurrently with prediction and that the resulting model produces long-term predictions.

1 Introduction and Related Work

In order to successfully interact with a dynamic environment, a person, a robot or any other autonomous entity needs to reason about how the objects which populate this environment are going to move in the future. However, this knowledge about the future is often unavailable *a priori*, hence it is necessary to resort to prediction: estimate future motion based on available knowledge about the object's present and past states. This explains the importance of prediction techniques for a number of research domains like motion planning, tele-surveillance and automatic traffic control [1, 2].

This work focuses on motion prediction for objects which are able to perform trajectories as a result of an internal motion planning process or decision mechanism (*e.g.* persons, animals and robots). It is assumed that such plans

* This work has been partially supported by a Conacyt scholarship. We also want to thank the support of the french CNRS Robea ParkNav and the Predit Mobivip projects.

are made *with the intention* to reach a specific goal, thus the name *intentional motion* which will be used hereafter to designate this kind of motion.

Assuming that the object's decision mechanism as well as all the relevant variables at every time step (*e.g.* internal state, sensorial input, etc.) are known, predicting its trajectory consists in replicating the planning process in order to find the intended trajectory. However, this assumption is not realistic. Neither the planning model nor the variables are known or observable (what is the decision mechanism of a human being?) and they must be inferred from observed motion before performing prediction. This leads to the following decomposition of the problem:

- *Learning.* Construct a plan representation based on observations.
- *Prediction.* Use the representation obtained during learning to estimate future states on the basis of present knowledge.

Thus, learning consists in observing a given environment in order to construct a representation of every possible plan for it. But, how long should we observe the environment in order to construct such a "plan library"? Given the enormous number of possible plans for all but the simplest environments, there is not a simple answer. This raises an important problem of existing learning techniques (*e.g.* [3, 4]): the use of a "learn then predict" approach, meaning that the system goes through a learning stage where it is presented with a set of observations (an example dataset) from which it constructs its plan models. Then, the plan library is "frozen" and the system goes into the prediction stage.

The problem with this approach is that it makes the assumption that all possible plans are included in the example dataset, which, as we have shown, is a difficult condition to meet. This paper addresses the problem by proposing a "learn and predict" approach which is able to learn in an incremental fashion (*ie* by continuously refining its knowledge on the basis of new observations used for prediction). To the extent of our knowledge, this is the first intentional motion prediction technique in the literature to have this property.

Learning techniques used by the "learn then predict" approaches are very diverse. For example in [5] plans are modeled as series of straight motion segments which are clustered together. In [3] and [6], typical behaviors are learned by clustering whole trajectories. In [7] Bui proposes Abstract Hidden Markov Models as a way to represent plans as hierarchies of probabilistic sub-plans or policies. Although the approach does not define an automatized learning mechanism, this has been done in [4] by using the Expectation-Maximization algorithm.

In this paper, we present an approach which is able to continuously learn from observations in an incremental fashion. It models plans as Hidden Markov Models (HMM)[8] augmented with a variable which indicates the goal that the plan intends to reach². The learning algorithm is composed of

² An HMM is a stochastic finite-state automaton which models a process whose state evolves according to a transition probability at discrete time-steps. The

two modules: in the first one, the Growing Neural Gas algorithm [9] is used to estimate both the set of states in the model and the observation probabilities. The second module identifies goals and then uses a Maximum-Likelihood criterion to update the transition probability of the model. As mentioned above, the technique determines the number of goals and states in the model, thus learning the structure of the underlying HMM.

The rest of the paper is structured as follows: section 2 presents an overview of the problem. Section 3 discusses the details of our HMM-based probabilistic model and describes how it is used to represent plans. The details of the learning algorithm are presented in section 4. Section 5 discusses the experimental results. The paper ends by exposing our conclusions and explaining future research directions.

2 Problem Overview

This paper proposes an unsupervised learning algorithm which constructs plan representations by observing the motion of objects (e.g. pedestrians, vehicles, etc.) moving in a given environment. Plans are modelled as Hidden Markov Models augmented with a variable γ which is used to represent the particular goal that the object intends to reach.

The input of the learning algorithm is a continuous stream of observations $o_t = \{o_1, o_2, \dots\}$ gathered through a tracking system. In order to keep notation simple, we will assume that no more than one object is observed at the same time, noting that the approach is easily generalizable to the multi-object case. It will also be assumed that the tracking system can determine when the object has stopped or exited the environment.

Every observation $o_t = (x_t, y_t, \eta_t)$ returned by the tracker consists of an estimate of the object's position³ at time t and a binary variable η_t which indicates whether the object has reached the end of its trajectory ($\eta = 1$) or not ($\eta = 0$). A trajectory ends when the object stops moving or exits the environment.

Learning will consist in estimating the parameters of the slightly modified HMM which will be presented in the following section.

state of the process may only be observed through a noisy sensor, the probability that a measure provided by the sensor corresponds to a given state is known as the observation probability.

³ Higher-dimensional observations (ie (x_t, y_t, x'_t, y'_t)) may also be used as input by the algorithm.

3 Probabilistic Model Definition

In order to develop our model, we will start from the HMM⁴ joint probability distribution (JPD) for a single time-step, which may be written as follows:

$$p(q_t, q_{t-1}, o_t) = p(q_{t-1})p(q_t | q_{t-1})p(o_t | q_t) \quad (1)$$

Where q_{t-1} and q_t represent the state at time $t - 1$ and t , respectively, and o_t represents the observation returned by the sensor at time t . The decomposition contains the three probabilities that define an HMM: a) the state prior, or belief state $p(q_{t-1})$; b) the transition probability $p(q_t | q_{t-1})$ and c) the observation probability $p(o_t | q_t)$.

In the context of this work. Discrete states will correspond to positions in the environment and transition probabilities will depend on the particular goal that an object intends to reach. In order to account for different goals, we will augment the HMM with a variable γ which is used to represent them:

$$p(q_t, q_{t-1}, o_t, \gamma) = p(q_{t-1})p(\gamma)p(q_t | \gamma, q_{t-1})p(o_t | q_t) \quad (2)$$

This JPD has been obtained from eq. 1 by making two additional conditional independence assumptions: a) The goal does not depend on the previous state $p(\gamma | q_{t-1}) = p(\gamma)$ and b) given the state, the observation is independent of the goal $p(o_t | q_t, \gamma) = p(o_t | q_t)$.

Due to the fact that γ is not time-dependent, this may be regarded as having a different Markov model for every value of γ , where all such models share the same observation probabilities and number of states. The idea is a simplified version (*ie* without the actions) of the probabilistic planning technique known as Markov Decision Processes.

Having defined a JPD, we will proceed to specify all the model's relevant variables as well as their respective domains:

- $N \in \mathbb{N}$: The total number of discrete states in the model. These states correspond to positions in the environment.
- $q_t, q_{t-1} \in [1, N]$: The object's states at time t and $t - 1$, respectively.
- $o_t \in \mathbb{R}^2$: The object's state estimation returned by the sensor at time t . (*ie* the observation variable).
- $G \in \mathbb{N}$: The total number of goals in the model. The goals correspond to specific *places* in the environment (*ie* it may correspond to many discrete states).
- $\gamma \in [1, G]$: The goal that the object intends to reach.

Finally, we the representations we have chosen for the probability distributions:

⁴ In this section, it is assumed that the reader is familiar with Hidden Markov Models. The interested reader is referred to [8] for an excellent tutorial on the subject.

- $p(q_t | \gamma, q_{t-1})$: Table, it will be further described in §4.3.
- $p(o_t | [q_t = i])$: Gaussian $\mathcal{G}(\mu_i, \sigma_i)$.
- $p(q_0)$: Uniform $\mathcal{U}_N = \frac{1}{N}$. This probability is used to initialize the belief state for a new trajectory.
- $p(\gamma)$: Uniform $\mathcal{U}_G = \frac{1}{G}$. This probability is used to initialize the goal's belief for a new trajectory.

4 Parameter Learning Algorithm

On the basis of the model specification presented in §3 it is possible to define a learning algorithm which consists in estimating parameters from observations. Having defined the priors as uniform distributions, this leaves four parameters to be estimated: the transition and observation probabilities, and the values for N and G . It is worth noting that, by learning both N and G , the proposed technique is able to learn the structure of the model. This is an significant departure from existent techniques [3, 4], which depend on values fixed *a priori*.

Assuming that, for every observed trajectory the associated goal is known, learning may be performed using the Baum-Welch algorithm [10] which is a specialization of Expectation-Maximization [11] and has become the standard learning technique for HMM's. However, it has two problems which prevent its application to our particular problem: a) it is not incremental and b) it needs to know the number of states to be learned *a priori*. The first problem may be solved by using incremental variants of the algorithm [12, 13], but the second one is more difficult to solve and is not a trivial task. Moreover, we want to deal with the general case, where goals are not known beforehand and should be identified.

The approach proposed in this paper takes a different approach by splitting the problem in three tasks:

1. *State GNG*. The observation probability $p(o_t | q_t)$ and the number of states N are estimated using the Growing Neural Gas algorithm (§4.1).
2. *Goal GNG*. Another instance of GNG is used to estimate the number of goals G as well as their position.
3. *Viterbi Counting*. The Viterbi algorithm [14] is used to perform a maximum likelihood (ML) estimation of the transition probability $p(q_t | \gamma, q_{t-1})$. This estimation uses the outputs of tasks 1 and 2 (§4.3).

The rest of the section provides the details of the tree learning tasks.

4.1 Learning Discrete States and Observation Probabilities

The observation probability for a given state $p(o_t | [q_t = i])$ is defined as a gaussian. Therefore, the learning algorithm should estimate the mean value μ_i and standard deviation σ_i for the N states.

This rises the question of the "correct" value for N , which is an important question. The state space is continuous, when it is mapped to a finite set of discrete values an error is introduced in the representation. The number of states allows to trade off accuracy and computational efficiency. By incrementing the value of N the approximation error – also known as distortion – is reduced at the expense of additional calculation burden.

There is another way of reducing the distortion: discrete states may be placed in such a way that the mean distance between them and observed data is minimized. This is known as Vector Quantization [15].

Our approach uses a Vector Quantization algorithm known as Growing Neural Gas (GNG) [9] in order to estimate the number of discrete states of the model as well as the mean values and standard deviation of the observation probabilities. This algorithm has been chosen between many different approaches existent in the literature [15, 16, 17, 18] due to its following properties:

- It is fast. The costliest operation is of $O(N)$. This can be further optimized by using a hierarchical structure like an r-tree[19].
- The number of states is not fixed. New states are added and deleted as observations arrive.
- It is incremental. This makes it suitable to process continuous streams of observations.

The algorithm processes observation on a one by one basis. It produces a graph, where nodes representing discrete states are explicitly linked to their closest neighbors (the graph is a subset of the Delaunay triangulation). Every node i is associated to a vector μ_i known as the centroid.

The application of this structure to estimate the required parameters is straightforward: state information $\{x_t, y_t\}$ contained in each observation is used as an input for a GNG. The resulting set of nodes represents discrete states whose centroids are the mean values of the observation probabilities. The standard deviation σ_i for state i is approximated by averaging the half length of the links emanating from the corresponding node

Insertion of new states is no longer allowed when $\arg \min \sigma_i$ is less than a given threshold. This restrains the algorithm from discretizing the space below the sensor's precision. Thresholding may be regarded as defining a minimum cell size, which is similar to conventional grid approaches but with two important advantages: a) the location of the cells is not fixed *a priori* and b) only relevant cells are represented. The latter advantage depends on the ratio between the existing positions (*ie* the full grid) those which are actually visited by objects. Usually, this advantage becomes more important as the state dimension grows due to motion constraints which apply to the object (*e.g.* acceleration and speed limits, inaccessible areas, etc.).

An example of the use of GNG is presented in fig. 1. The environment is a simulator of the laboratory's entry hall. It contains a number of places which may constitute motion goals (*e.g.* the stairways in the bottom or the

two doors in the top of figure 1a). Fig. 1b presents the state of the GNG structure after processing 1000 trajectories.

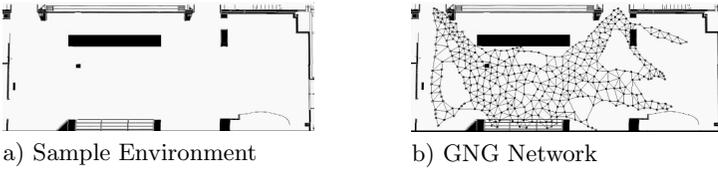


Fig. 1. Using GNG to represent discrete states in the laboratory entry hall.

4.2 Identifying Goals

The problem of automatically identifying the goals that an object seeks to reach using only observation data is very difficult since these goals are often related to information which is not present in this data (*e.g.* the presence of a billboard).

The approach taken here aims to identify goals based on a simple hypothesis: when an object stops moving (or exits the environment) it is because it has reached its goal. This leads to a simple goal identification scheme: every observation o_t having $\eta_t = 1$ (end observation) is sent to a GNG structure which clusters this information together into high-level goals.

The nodes of the resulting GNG graph corresponds to goals⁵. The graph itself may be used to identify the goal that corresponds to a given end-state observation:

$$\gamma = \min \arg_i \|(x_t, y_t) - \mu_i^g\|, \text{ for } \eta_t = 1 \quad (3)$$

4.3 Learning Transition Probabilities

Transition probabilities are updated once a complete trajectory is available, this means that all non-end observations are stored until an observation having $\eta = 1$ is received, then, expression 3 is used to compute the attained goal g . For every observation in the trajectory o_t , the Viterbi algorithm is used in order to find q_t given the past state $q_{t-1} = i$ (which has been estimated in the previous iteration)⁶:

⁵ notations μ_i^s and μ_i^g will be used henceforth in order to distinguish between state and goal GNG's

⁶ This implies iterating through the domain of q_t , meaning that the update step has cost $O(N)$.

$$q_t = \max \arg_j \left\{ p([q_t = j] \mid [\gamma = g], [q_{t-1} = i]) p(o_t \mid [q_t = j]) \right\}$$

The obtained values for g , i and j are then used as indices to update a transition count matrix A on a maximum-likelihood criterion:

$$A[g, i, j] \leftarrow A[g, i, j] + 1 \quad (4)$$

If the observation correspond to the first step of a trajectory only the current state is estimated using:

$$q_0 = \max \arg_i \{ p(o_0 \mid [q_0 = i]) \} \quad (5)$$

Transition probabilities are calculated using:

$$p([q_t = j] \mid [\gamma = g], [q_{t-1} = i]) = \frac{\mathcal{A}[g, i, j]}{\sum_h \mathcal{A}[g, i, h]} \quad (6)$$

Finally, when N or G change due to additions or deletion on the corresponding GNG structures the corresponding columns and rows are simply inserted or deleted accordingly, this is possible due to the fact that we are storing counts instead of probabilities in the transition matrix.

5 Experimental Results

In order to validate it, the proposed approach has been applied to the prediction of pedestrian motion in the entry hall of the Inria laboratory, which is a rectangular area of approximately 8 x 20 meters. As it may be seen in fig.2, the environment consists mostly of an open area without much structure.

We have performed experiments with both real and simulated data. Real data has been gathered through a vision system which tracks people using a single camera having wide-angle lenses. The system projects observations from the camera plan to the floor plan. It is worth noting that, due to the projection process, the system ends up covering only about 60% of the total area.

Simulated data consists of noisy trajectories between predefined sequences of control points.

5.1 Evaluation Criterion

In order to perform prediction, the probability $p(\gamma \mid o_0, \dots, o_t)$ has been estimated using a particle filter with a resampling step [20] to integrate new observations.

The performance of the algorithm has been evaluated by measuring the difference between the predicted and effective final destination. The first n

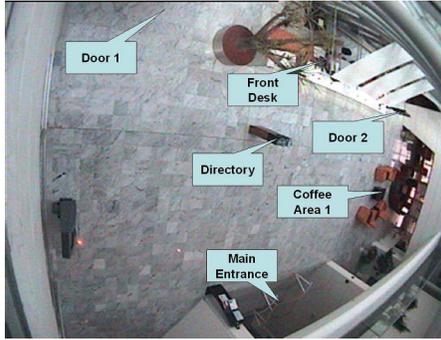


Fig. 2. The INRIA entry hall

observations of a trajectory were used to predict the most probable goal $g = \max \arg_i p([\gamma = i] | o_0, \dots, o_n)$. The global estimation error is calculated as the average of all the distance between the goal such obtained and the real end of the trajectory.

5.2 Results

We have run our experiments using datasets of 600 trajectories both for real and simulated data. The algorithm was initialized by processing 500 trajectories before starting to record the results. The remaining 100 trajectories were processed as follows: for every trajectory, the predicted goal is estimated using 10% of its length, then 20% and so on until 90%. This allows us to measure how new knowledge improves prediction.

The results obtained for both simulated and real data are presented in fig. 3.

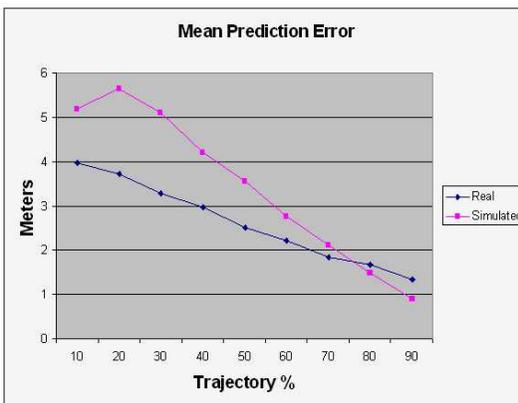


Fig. 3. Experimental results

It may be seen that both error curves decrease in near linear fashion with respect to the known fraction of the trajectories, we regard this as an encouraging result, particularly in the case of real data, which was very noisy due to the tracker's inability to adequately track the object's motion during all of its motion. However, we think that faster (non-linear) convergence rates are achievable, in particular by using a more efficient goal discovery mechanism.

It may be surprising to find that real data seems to perform better than simulated data. The reason is that the simulator produces trajectories which cover the entire entry hall, while, as we have explained above, real data is gathered only in a fraction of the environment.

It is important to mention that all the results presented here are preliminary. For example, the chosen performance measure should be improved to take into account situations where a trajectory passes through more than one goal, in this case, the system will probably predict that the object intends to reach these "intermediate" goals (which we think is fine), but, as they do not correspond to the trajectory's final position, the resulting prediction error will be high.

5.3 Real-Time

In our experiments, prediction has been performed simultaneously with learning and graphic display for the test data set. Our unoptimized implementation of the technique runs on a 2 Ghz Athlon PC at a frequency of 20–60 Hz. Even if we consider this as adequate for most real-time systems involving pedestrians, the system may be further optimized at the code level or by using a more efficient technique for searching the winner in the GNG structure, for example (§4.1).

6 Conclusions

In this paper, we proposed a method for learning motion plans from observations. Our approach represent plans as Hidden Markov Models. Learning consists of three modules: a) the Growing Neural Gas algorithm is used to estimate the total number of states N as well as the observation probability distribution; b) another GNG structure is used to estimate the number and positions of goals in the environment, and c) the Viterbi algorithm is used to perform a Maximum-Likelihood estimation of the transition probability.

The main contribution of this technique is that it follows a "learn and predict" approach, thus allowing the continuous improvement of existent knowledge on the basis of new observations. To the best of our knowledge no other technique in the literature is able to do that. A second important contribution consists on the fact that, by learning the number of states and the number of goals, this technique is able to learn the structure of the model, this distinguishes our work from techniques with fix this values *a priori*.

The technique has been implemented and applied to both real and simulated data. The experiments show that the learned model may be used to efficiently predict the intended goal of an object. Moreover, this is performed in real time.

7 Future Work

The approach presented in this paper is a first approximation to the problem. In the short term, our goal is to test the approach in a different setting: the ParkView experimental platform, which is able to track a car moving in a parking lot (fig. 4).

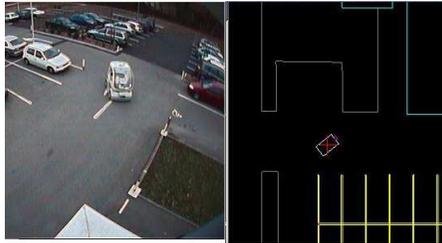


Fig. 4. The ParkView platform: *left*) camera view of the Cycab experimental car moving in the parking lot of the laboratory; *right*) the Cycab as detected on the tracking system.

In the medium term, a number of lines of work are being considered: a) including velocity and object size in the space representation; b) modelling of semi-dynamic objects such as doors which may be either open or closed; c) the extension of the algorithm to learn hierarchical plan models such as Abstract Hidden Markov Models [7].

References

1. D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell, "Towards robust automatic traffic scene analysis in real-time," in *Proceedings of the 33rd Conference on Decision and Control*, Lake Buena Vista, FL (USA), December 1994, pp. 3776–3781.
2. K. Kyriakopoulos and G. Saridis, "An integrated collision prediction and avoidance scheme for mobile robots in non-stationary environments," in *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, Nice, France, May 1992, pp. 194–199.
3. M. Bennewitz, W. Burgard, and S. Thrun, "Learning motion patterns of persons for mobile service robots," in *Proceedings of the IEEE Int. Conf. On Robotics and Automation*, Washington, USA, 2002, pp. 3601–3606.

4. S. Osentoski, V. Manfredi, and S. Mahadevan, "Learning hierarchical models of activity," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.
5. E. Kruse, R. Gusche, and F. M. Wahl, "Acquisition of statistical motion patterns in dynamic environments and their application to mobile robot motion planning," in *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Grenoble, France, 1997, pp. 713–717.
6. D. Vasquez and T. Fraichard, "Motion prediction for moving objects: a statistical approach," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA (US), apr 2004, pp. 3931–3936.
7. H. Bui, S. Venkatesh, and G. West, "Policy recognition in the abstract hidden markov models," *Journal of Artificial Intelligence Research*, vol. 17, pp. 451–499, 2002. [Online]. Available: citeseer.ist.psu.edu/bui02policy.html
8. L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Readings in speech recognition*, pp. 267–296, 1990.
9. B. Fritzke, "A growing neural gas network learns topologies," *Advances in Neural Information Processing Systems*, 1995.
10. L. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *Annals of Mathematical Statistics*, no. 37, 1966.
11. N. Dempster, A. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 9, no. 1, pp. 1–38, 1977, series B.
12. Y. Singer and M. K. Warmuth, "Training algorithms for hidden markov models using entropy based distance functions." in *Advances in Neural Information Processing Systems 9, NIPS*. Denver, CO (USA) December 2-5, 1996: MIT Press, 1996, pp. 641–647.
13. R. M. Neal and G. E. Hinton, "A new view of the EM algorithm that justifies incremental, sparse and other variants," in *Learning in Graphical Models*, M. I. Jordan, Ed. Kluwer Academic Publishers, 1998, pp. 355–368.
14. A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, no. 2, pp. 260–269, April 1967.
15. Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. COM-28, pp. 84–95, 1980.
16. T. Kohonen, *Self-Organizing Maps*, ser. Springer Series in Information Sciences. Berlin, Heidelberg: Springer, 1995, vol. 30, (Second Extended Edition 1997).
17. M. Martinetz and K. J. Schulten, "A "neural-gas" network learns topologies," in *Proceedings of International Conference on Artificial Neural Networks*, T. Kohonen, K. M??kisara, O. Simula, and e. J. Kangas, Eds., vol. I, North-Holland, Amsterdam, 1991, pp. 397–402.
18. G. Carpenter, S. Grossberg, and D. Rosen, "Fuzzy art: An adaptive resonance algorithm for rapid, stable classification of analog patterns," in *Proc. Int. Joint Conf. Neural Networks*, vol. II, Seattle, USA, 1991, pp. 411–420.
19. A. Guttman, "R-trees: A dynamic index structure for spatial searching." in *SIGMOD Conference*, 1984, pp. 47–57.
20. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, feb 2002. [Online]. Available: citeseer.ist.psu.edu/article/arulampalam01tutorial.html

Design and Locomotion of a Semi-passive Mobile Platform

Amir Shapiro¹ and Shraga Shoval²

¹ Department of Mechanical Engineering, Ben Gurion University of the Negev, Beer Sheva 84105, Israel. ashapiro@bgu.ac.il

² Department of Industrial Engineering & Management, College of Judea and Samaria, Ariel 44837, Israel. shraga@yosh.ac.il

Summary. This paper presents a novel design and a motion planner for a semi-passive mobile robot. The robot consists of an upper circular body and three identical semi-passive driving mechanisms. Each mechanism consists of a passive wheel that can freely roll, a rotation actuator along the normal axes and a linear actuator for motion along the radial direction of the upper body center. The robot is equipped with an inclinometer to measure the surface slope. Each wheel is also equipped with a rotational encoder to measure roll. Using an odometric model, data from these encoders determines vehicle position. Kinematic analysis provides tools for designing a motion path that steers the robot to the desired location, and determines the singular configurations. Due to the passive roll, there is no longitudinal slippage, and lateral slippage is determined from the kinematic and odometric models. This enables accurate and reliable localization even with slippage. A gait pattern planer for downhill, as well as horizontal and uphill surfaces is presented. A prototype robot has been built and field tested. Experimental results verify the suggested models.

Keywords: Passive motion planning, skid steering, slippage.

1 Introduction

Wheel slippage is one of the dominant features that affect the efficiency, reliability, feasibility and stability of mobile robot motion. Uncontrolled slippage causes undesired motions that result in erroneous position and orientation. The most common method for autonomous relative position estimate - odometry, is subject to unbounded errors due to slippage [2], and requires an additional positioning system (e.g. Map-Matching, GPS, Beacon-Based Triangulation). This problem becomes critical when no absolute positioning system is available (e.g. space, underground or indoor missions). Furthermore, additional tasks such as trajectory planning and obstacle avoidance cannot be reliably performed in the presence of uncontrolled slippage. Many researchers deal with robot-surface interaction, particularly on slippery terrains. Bidaud et. al. [1] deal with wheel-soil interaction models. Iagnemma et. al. [7]

describe terrain estimation and sensing methodology using visual, tactile and vibrational feedback. Ferretti et. al. [5] exploits high resolution encoders to compensate for non linear friction terms. Physics based motion control that involves a model of traction mechanics with the consideration of force distribution among the wheels is discussed in [3]. In this approach the wheel-soil contact angle and the distribution of the load on each wheel are considered, and a control system maximizes traction between the vehicle and the terrain. Yoshida and Hammano [11] investigate the tire-soil traction mechanics as well as the body-suspension-wheel dynamics of a mobile robot. Conventional locomotion uses legs or powered wheels to generate motion. In contrast, our robot relies on relative motion of the joints to generate motion of the central body similar to the motion of a downhill skier on an icy surface, or locomotion on rollerblades. A novel robot design that has the ability to switch between skating and walking modes is the Roller-Walker [4]. This quadruped robot has the ability to switch between walking and skating modes. Passive wheels at the end of each leg fold flat to allow the robot to walk. In the skating mode, the wheels are rotated into place to allow the robot to carry out skating motion. Another example is the ROLLERBLADER [6]. This robot is different from the Roller Walker in its ability to raise the rollerblades off the ground. This allows the use of gaits that mimic those used by human rollerbladers. Shimizu [10] developed both a skiing robot and a snowboarding robot that can model how humans perform turns on skis or snowboard.

Semi-passive driving mechanism has several advantages over regular powered mechanisms. First, no longitudinal slippage occurs, given minimal friction between the wheels and the surface. The minimal friction is required for overcoming the rotational friction between the wheels and their housing, which is significantly reduced even by conventional bearings. The elimination of longitudinal slippage is essential for accurate and reliable odometry, particularly for outdoor missions. Using semi passive driving mechanism, combined with intelligent motion planner can significantly reduce the power consumption of the vehicle. Utilizing the powered mechanism only when required (e.g. traveling uphill or on horizontal surface), while changing to passive motion when possible (downhill travel or using the vehicle inertia) can save significant energy consumption, and therefore increase the autonomy of the system. Finally, while replacing the passive wheels with ice skating blades it can move on slippery surfaces such as ice utilizing the unique characteristic of small friction forces on the longitudinal direction and large friction forces on the lateral direction.

The paper is organized as follows: In Section 2 we describe the robot design. Section 3 provides kinematic analysis and geometrical insights of singular configurations under no lateral slippage conditions. Section 4 suggests an odometric model for localization and lateral slippage detection. Section 5 describes motion patterns for downhill, uphill and horizontal locomotion. Section 6 presents experimental results that verify the motion planner and the odometric model. Section 7 provides the conclusions.

2 Robot Description

The robot consists of an upper circular body and three identical semi-passive driving mechanisms shown in Fig. 1(a). Each driving mechanism consists of a passive wheel that rolls freely along its longitudinal direction. The mechanism has two actuators: a rotational and a linear actuators shown in Fig. 1(b). Both actuators use Pittman DC servo motors. The linear actuators use a lead screw mechanism with two parallel slide guides and linear bearings. The rotational actuator uses timing belt mechanism to reduce the total robot height and lower the center of gravity. The robot is equipped with studded-like tires to increase traction and reduce slippage. The wheels can be easily replaces by ice skating blades or skis for motion on icy or snowy surfaces. The robot is equipped with an inclinometer to measure the surface slope.

The robot is equipped with six degrees of freedom, allowing for changes in the internal configuration which are required for various motion patterns (as described in the following sections). The rotation along the normal to the central body determines the longitudinal rolling direction of the wheel. Since wheels are passive, we assume no longitudinal slippage (lateral slippage is permitted)³. Each of the passive wheels is equipped with a rotational encoder to measure rolling. Data from these encoders is used by the odometric model for relative position and orientation estimation of the robot. Furthermore, based on the kinematic and odometric models (discussed in Sections 3 and 4), the amount of lateral slippage on each wheel can be determined.

3 Kinematic Analysis

The robot's c-space (configuration space) contains nine parameters, $\mathbf{q} = (x_b, y_b, \theta_b, d_1, d_2, d_3, \theta_1, \theta_2, \theta_3) \in \mathbb{R}^9$, out of which only six are actuated. Hence the robot's central base is un-actuated. The goal is to design a motion path for the actuated joints such that it steers the entire robot to a desired location. To begin, we compute each wheel's center point location and velocity using rigid body transformation and its time derivative:

$$\mathbf{p}_i = \mathbf{d}_b + R_b d_i \mathbf{r}_i \quad \text{and} \quad \dot{\mathbf{p}}_i = \dot{\mathbf{d}}_b + R_b \dot{d}_i \mathbf{r}_i - \dot{\theta}_b J R_b d_i \mathbf{r}_i \quad \text{for } i = 1, 2, 3 \quad (1)$$

where $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ and R_b is the rotation matrix of the central base angle θ_b . Assuming no lateral slippage, the wheel's center point velocity has no component along the lateral direction. Therefore:

$$\mathbf{c}_i^T \dot{\mathbf{p}}_i = \mathbf{0} \quad \text{for } i = 1, 2, 3 \quad (2)$$

where $\mathbf{c}_i = R_b R_i (1, 0)^T$ is the lateral direction of the i^{th} wheel, and R_i is the wheel's rotation matrix of angle θ_i . Let us make the following definitions:

³ This assumption is valid only for wheels with small inertia and for relatively small accelerations.

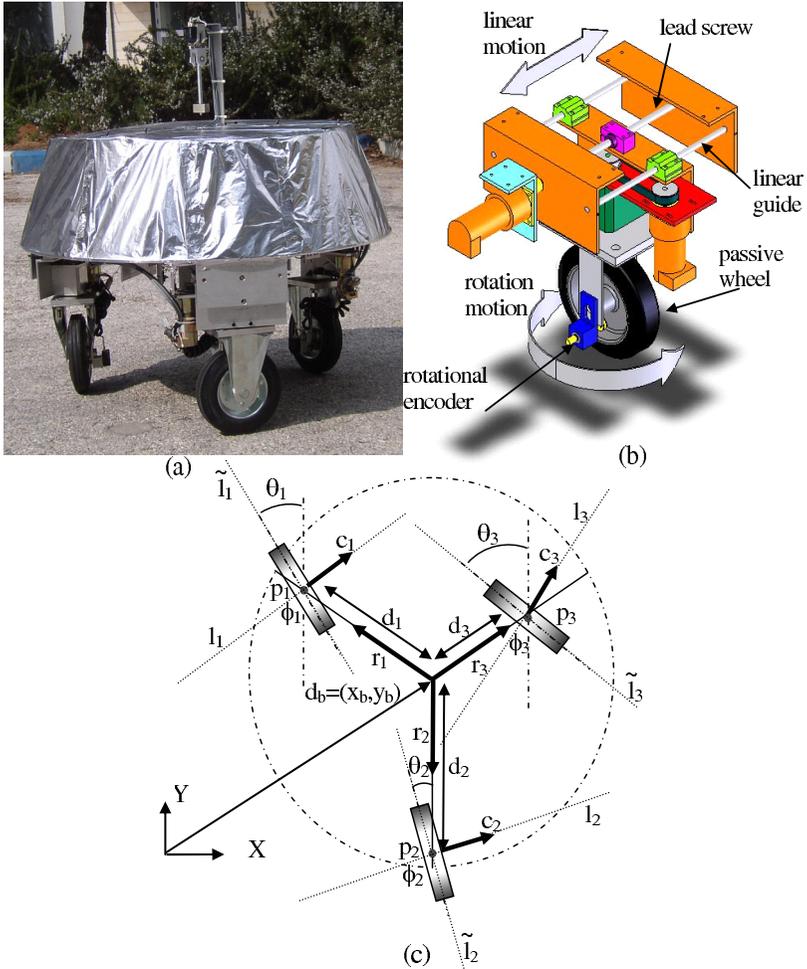


Fig. 1. (a) Prototype of the three wheeled robot, (b) design model of the wheel mechanism, and (c) the robot's parameters.

$$V(\mathbf{q}) = -diag(\mathbf{c}_1^T R_b \mathbf{r}_1, \mathbf{c}_2^T R_b \mathbf{r}_2, \mathbf{c}_3^T R_b \mathbf{r}_3) \in \mathbb{R}^{3 \times 3} \quad (3)$$

and

$$K(\mathbf{q}) = \begin{bmatrix} \mathbf{c}_1^T & -d_1 \mathbf{c}_1^T J R_b \mathbf{r}_1 \\ \mathbf{c}_2^T & -d_2 \mathbf{c}_2^T J R_b \mathbf{r}_2 \\ \mathbf{c}_3^T & -d_3 \mathbf{c}_3^T J R_b \mathbf{r}_3 \end{bmatrix} \in \mathbb{R}^{3 \times 3}.$$

The no-slippage constraint (2) can now be written in matrix form as follows:

$$\begin{pmatrix} \dot{\mathbf{d}}_b \\ \dot{\theta}_b \end{pmatrix} = G(\mathbf{q}) \begin{pmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{pmatrix} \quad \text{where } G(\mathbf{q}) = K^{-1}(\mathbf{q})V(\mathbf{q}) \in \mathbb{R}^{3 \times 3}. \quad (4)$$

This constraint depends on velocities as well as on the configuration. Therefore, (4) introduces three non-holonomic constraints, and the robot is said to be a *non-holonomic, under-actuated system*. Let $\mathbf{u} = (\mathbf{u}_d, \mathbf{u}_\theta)^T \in \mathbb{R}^6$ be vector of control inputs, where $\mathbf{u}_d = (\dot{\mathbf{d}}_1, \dot{\mathbf{d}}_2, \dot{\mathbf{d}}_3)^T$ and $\mathbf{u}_\theta = (\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3)^T$. Then the robot's kinematic system is:

$$\dot{\mathbf{q}} = \begin{pmatrix} G(\mathbf{q})\mathbf{u}_d \\ \mathbf{u} \end{pmatrix}. \quad (5)$$

The central base velocity is uniquely determined by the actuators' velocities only if $G(\mathbf{q})$ has full rank. Moreover, existence and uniqueness of solution to the robot's kinematic system is assured only if $\text{rank}(K(\mathbf{q})) = 3$. Matrix $K(\mathbf{q})$ is of full rank if, and only if, the three lines l_1, l_2 , and l_3 do not intersect in a single point and are not mutually parallel. These lines are given by $\mathbf{l}_i = \mathbf{p}_i + t_i \mathbf{c}_i$ for $i = 1, 2, 3$, and t_i is a length parameter along the i^{th} line (Fig. 1(c)). If the robot is not in a singular configuration, the central base velocity is fully controllable using the linear actuators velocities. Later on we use this fact to conduct uphill motion. However, singular configuration can be used for free slide in downhill motion.

4 Odometric Model and Slippage Detection

In this section we describe the odometric model of the robot and a method for slippage detection. As previously discussed, each wheel is equipped with a rotational encoder to measure the passive roll - ϕ_i , (Fig. 1(c)). Given the central body velocity, the wheels' center point velocity is computed in (1). Taking the derivative of ϕ_i and multiplying by the wheel radius - W_r , gives the i^{th} wheel center point longitudinal velocity. Equating the latter term with the i^{th} wheel center point velocity projected on the longitudinal direction, denoted $\tilde{\mathbf{c}}_i = -J\mathbf{c}_i$, results in:

$$\tilde{\mathbf{c}}_i^T \dot{\mathbf{p}}_i = \dot{\phi}_i W_r \quad \text{for } i = 1, 2, 3 \quad (6)$$

Based on (6) it is possible to evaluate the central base velocities while measuring the passive wheels' rotation velocities and the actuators' positions and velocities. Let us define the 3×3 matrix $\tilde{K}(\mathbf{q})$ as follows:

$$\tilde{K}(\mathbf{q}) = \begin{bmatrix} \tilde{\mathbf{c}}_1^T & -d_1 \tilde{\mathbf{c}}_1^T J R_b \mathbf{r}_1 \\ \tilde{\mathbf{c}}_2^T & -d_2 \tilde{\mathbf{c}}_2^T J R_b \mathbf{r}_2 \\ \tilde{\mathbf{c}}_3^T & -d_3 \tilde{\mathbf{c}}_3^T J R_b \mathbf{r}_3 \end{bmatrix} \in \mathbb{R}^{3 \times 3}, \quad (7)$$

then central base velocity is determined by:

$$\begin{pmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{\theta}_b \end{pmatrix} = \tilde{K}^{-1}(\mathbf{q}) \begin{pmatrix} \dot{\phi}_1 W_r - \dot{d}_1 \tilde{\mathbf{c}}_1^T R_b \hat{\mathbf{r}}_1 \\ \dot{\phi}_2 W_r - \dot{d}_2 \tilde{\mathbf{c}}_2^T R_b \hat{\mathbf{r}}_2 \\ \dot{\phi}_3 W_r - \dot{d}_3 \tilde{\mathbf{c}}_3^T R_b \hat{\mathbf{r}}_3 \end{pmatrix}.$$

Numerical integration of the central base velocity along the motion path determines the robot's central base position. Note that as long as $\tilde{K}(\mathbf{q})$ is not singular it is possible to calculate the base position *even under lateral slippage*. Matrix $\tilde{K}(\mathbf{q})$ is of full rank if, and only if, the three lines \tilde{l}_1, \tilde{l}_2 , and \tilde{l}_3 do not intersect in a single point and are not mutually parallel, where $\tilde{l}_i = \mathbf{p}_i + \tilde{t}_i \tilde{\mathbf{c}}_i$ for $i = 1, 2, 3$, and \tilde{t}_i is a length parameter along the \tilde{l}_i line (Fig. 1(c)).

Slippage detection method: After evaluating the central body velocities, it is possible to compute each wheels' center point velocity according to (1). This velocity vector, $\dot{\mathbf{p}}_i$, can be divided into two components: the longitudinal component, $\tilde{\mathbf{c}}_i^T \dot{\mathbf{p}}_i$, and the radial component,

$$\dot{s}_i = \mathbf{c}_i^T \dot{\mathbf{p}}_i \quad \text{for } i = 1, 2, 3.$$

Note that \dot{s}_i is the lateral velocity of the wheel center point. If the three lines \tilde{l}_1, \tilde{l}_2 and \tilde{l}_3 (Fig. 1(c)) do not intersect in a single point and are not mutually parallel, then *we can explicitly compute the amount of lateral slippage of each wheel*.

5 Motion Patterns

In this section we describe the motion patterns of the robot. Since motion is based on a semi-passive mechanism, there are two major patterns: Uphill and horizontal locomotion, and downhill motion.

5.1 Uphill and Horizontal Locomotion

In horizontal or uphill locomotion, gravitational force cannot be used to conduct motion. Rather, the robot actuators produce the required central body velocity. The motion planning problem is as follows: For a given path, $\boldsymbol{\alpha}(t)$ of the robot's central body, what should be the actuators' velocities. The Lafferriere and Sussmann method [9] is an example of such a motion planning method for under-actuated non-holonomic systems. The Lafferriere and Sussmann method requires the system to be nilpotent (i.e. high order of Lie products vanish). However, our system contains trigonometric function whose derivatives never vanish and therefore is not nilpotent. Eq.(4) shows that, in order to provide the robot's central body with any desired velocity, the linear joints should supply the joint's velocities

$$\mathbf{u}_d = V^{-1}(\mathbf{q})K(\mathbf{q}) \begin{pmatrix} \dot{\mathbf{d}}_b \\ \dot{\theta}_b \end{pmatrix}_{desired} \quad \text{where} \quad \begin{pmatrix} \dot{\mathbf{d}}_b \\ \dot{\theta}_b \end{pmatrix}_{desired} = \dot{\boldsymbol{\alpha}}(t). \quad (8)$$

Applying the velocities described in (8) to the linear actuators provides the central body with the desired velocity and it precisely follows the $\boldsymbol{\alpha}(t)$ path. This motion is limited by the linear actuators' stroke. When one of the linear actuators reaches its limit, all actuators stop. Next, the linear actuators return

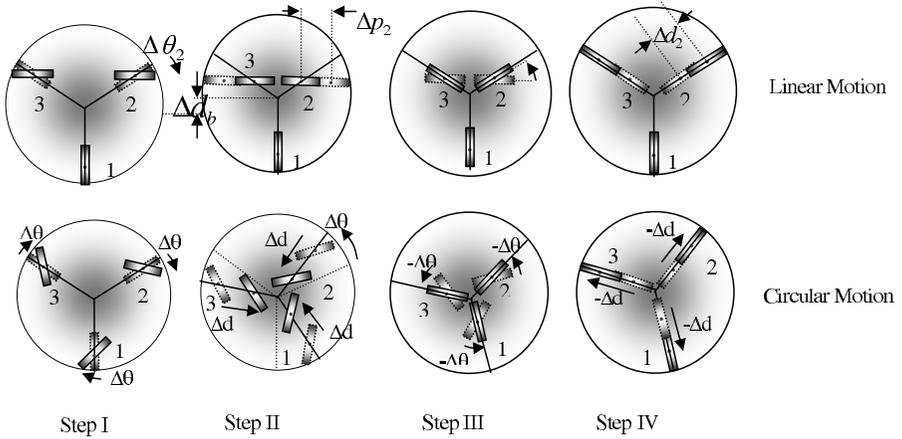


Fig. 2. Horizontal and uphill motion patterns

to their initial configuration without causing the robot's central body to move. From (3) we notice that if the matrix $V(\mathbf{q})$ is the 3×3 zero matrix, motion of the linear actuators will not affect motion of the robot's central body. $V(\mathbf{q})$ is a diagonal matrix with the terms $\mathbf{c}_i^T R_b \mathbf{r}_i$ on the diagonal. The $\mathbf{c}_i^T R_b \mathbf{r}_i$ terms vanish if each \mathbf{c}_i is perpendicular to $R_b \mathbf{r}_i$. This happens only when the wheels are in the radial directions.

Figure 2 illustrates the principle of our motion patterns for linear motion and for circular motion around the robot's center. Motion consists of four steps, in which some or all wheels change their angular and/or linear configuration resulting in the desired path for the robot's central body. Other trajectories can be generated using similar patterns.

Linear motion pattern consists on four phase motion: First the front two wheels (wheels 2 and 3) rotate $\Delta\theta_2$ and $\Delta\theta_3$ to the required configuration (Step I). Next, the linear actuators of wheels 2 and 3 move to provide the desired velocity to the central body (step II). The actuators move Δd_2 and Δd_3 , resulting in a longitudinal motion of wheels 2 and 3 of Δp_2 and Δp_3 , and a central body linear motion of Δd_b . It should be noted that $\Delta p_2 = \Delta p_3 = \Delta d_2 \cos \Delta\theta_3$, and $\Delta d_b = \Delta d_2 \sin \Delta\theta_3$. Once the linear actuators reach their maximum stroke, the wheels rotate such that their longitudinal axes coincide with the radial direction to the base center (in our case $-\Delta\theta_2$ and $-\Delta\theta_3$). Finally, the linear actuators return to their initial configuration.

Circular motion pattern: In the first step all wheels simultaneously rotate $\Delta\theta$ at the same direction. Next, all linear actuators move simultaneously the same distance Δd . This linear motion generates tangential forces that rotate the robot's body $\Delta\theta_b$ around its center. Once the linear actuators reach their limit, the wheels rotate such that their longitudinal axes coincide with the radial direction to the base center and the linear actuators return to their

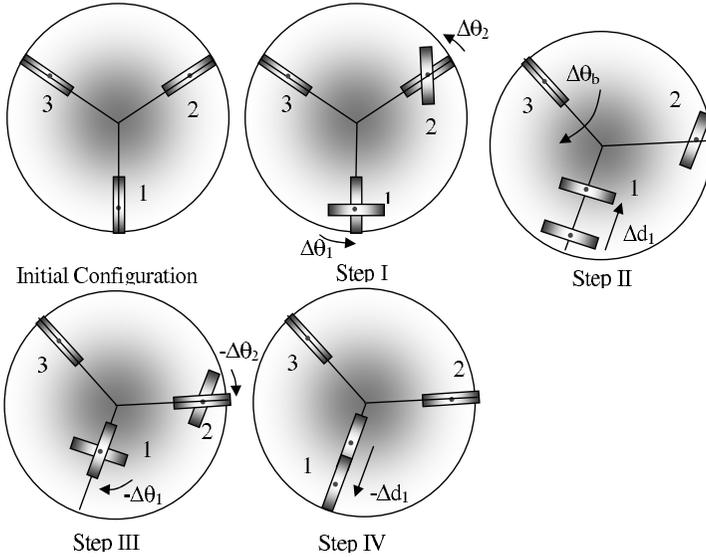


Fig. 3. Motion pattern for rotation about wheel 3.

initial configuration. The rotation around the robot center $\Delta\theta_b$ is given by

$$\Delta\theta_b = \frac{\Delta d}{d} \tan(\Delta\theta)$$

where d is distance between the robot center and the wheels. According to this equation, larger rotation angle of the wheels $\Delta\theta$ in step I increases the rotation of the robot's body in step II. However, $\Delta\theta = 90^\circ$ is a singular configuration in which the body can rotate freely with around its center. It also should be noted that as the wheels approach the robot's center, the rotation rate increases for the same $\Delta\theta$. However, the friction forces required for this rotation increase, and eventually break the static friction constraint, resulting in a lateral slippage of the wheels.

Other trajectories can be generated using similar patterns. For example, a rotation around one of the robot's wheels is shown in Figure 3 (in this figure around wheel 3). In step I wheel 2 rotate 60° and wheel 3 rotate 90° to the configuration shown. Next, the linear actuator of wheel 1 generates the rotation of the body by moving d_1 . In step III the wheels rotate back to the radial configuration and in step IV the linear actuator of wheel 1 returns to the initial configuration.

5.2 Downhill Locomotion

In downhill motion the gravitational force is used for dragging the robot downwards. For circular motion the lines l_i 's intersect in a single point and the

matrix $K(\mathbf{q})$ is singular. In this case the robot is constrained to move along an arc shaped path. The center of the arc is in the intersection point of the $\mathbf{l}_1, \mathbf{l}_2$ and \mathbf{l}_3 lines. Since the robot is an Euler-Lagrange system and since there is friction in the wheels' bearings, the system is passive and governed by gravitational potential energy. According to Koditschek [8] the configuration in which the system's potential energy is minimal is an asymptotical stable equilibrium point of the system. According to this observation, we find the radius and center of curvatures at each point of the desired motion path. Then we continuously set the \mathbf{l}_i 's intersection point at the center of curvature of the desired path by changing the robot's configuration. This way, the robot passively glides along the desired path. In the "snow-plough" motion two wheels are rotated in a "snow-plough" configuration, while the third wheel is used for steering. In this mode, speed is controlled according to the slide angle of the wheels relative to the motion direction. Figure 4 shows these two patterns for downhill motion.

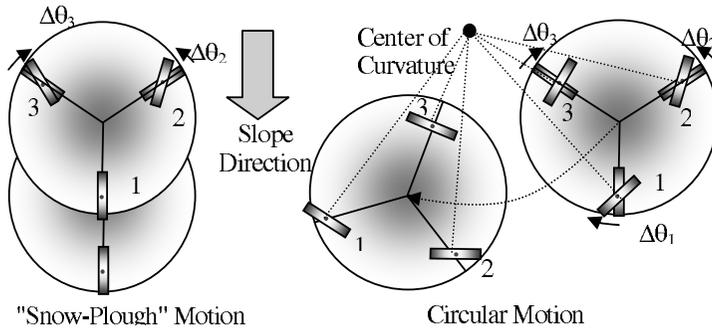


Fig. 4. Downhill motion patterns

6 Experimental Results

In this section we describe the experiments conducted with our autonomous robot, shown in Figure 1(a). In the first experiment we examine the linear motion pattern on a horizontal surface. Figure 5 shows the robot configuration (rotation angle and linear actuator of all wheels) during motion. Wheels 2 and 3 perform the required rotation and translation as shown in figure 2, while wheel 1 remains passive. The second part of figure 5 shows the actual wheels locations during motion as determined by our odometric model. Although the nominal path of the robot center is linear, actual path is not linear and bends to the right. This is expected as the experiment is conducted on a non-homogenous surface, and lateral slippage occurs, especially during stage II. This is also the reason for the oscillated motion of wheel 1, which nominally remains passive during that motion. However, the slippage is clearly detected by the odometric model.

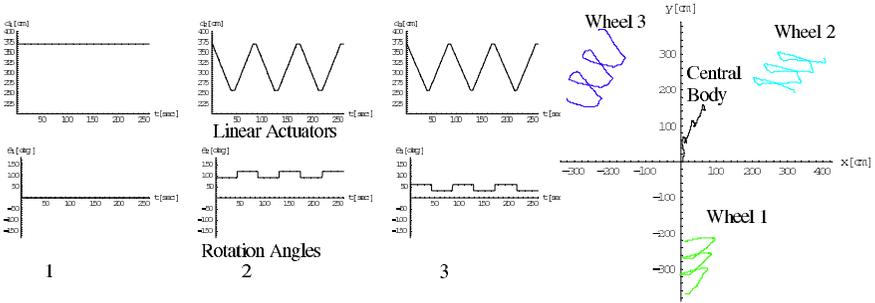


Fig. 5. Actual paths determined by the odometric model in horizontal linear motion

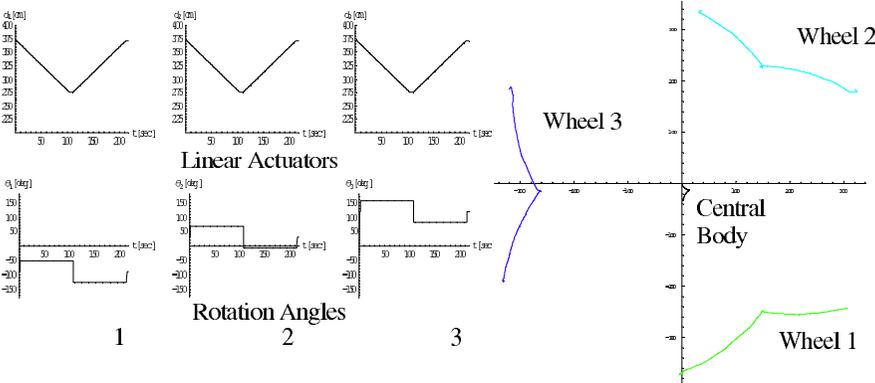


Fig. 6. Actual paths determined by the odometric model in horizontal circular motion

Figure 6 shows the robot configuration (rotation angle and linear actuator of all wheels) during circular motion around the center of the robot. Instead of returning to initial configuration (Step III in Figure 2) all wheels are rotated $-2\Delta\theta$ before the linear actuators return. This way rotation of the central body continues during Step IV, resulting in a double rotation angle for a full motion period.

Figure 7 shows a downhill motion using the "snow-plough" method. In this motion all linear actuators remains stationary, and wheels 2 and 3 are rotated until motion starts. Returning to the initial radial configuration stops the motion. The odometric model shows identical, parallel and near-linear motion of all wheels and robot's body. The non-linearity of the path occurs at the beginning and end of motion due to rotation of the wheels.

Finally we show an experiment for downhill rotation. In the experiment shown in figure 8, the robot rotates around wheel 3 according to the pattern shown in figure 4. In this pattern wheels 1 and 2 rotate $\Delta\theta_i$ such that I_1 and I_2 (lines through wheels 1 and 2 in the lateral direction) intersect at the contact

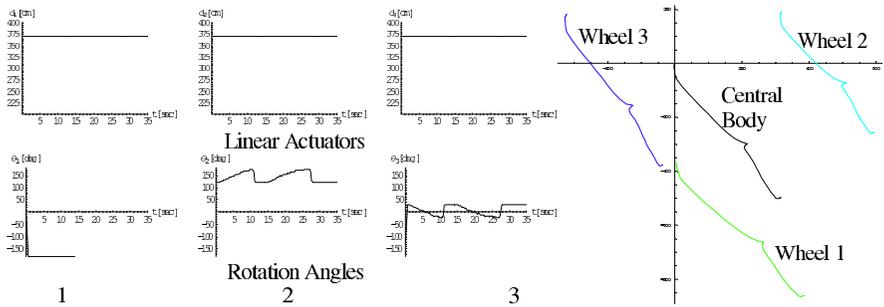


Fig. 7. Actual paths determined by the odometric model in linear downhill motion

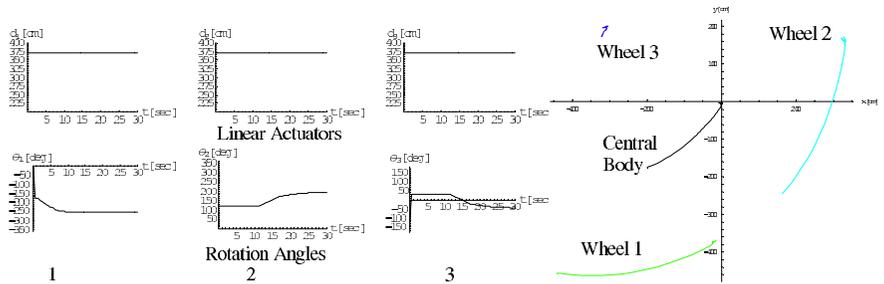


Fig. 8. Actual paths determined by the odometric model in linear downhill motion

point of wheel 3. The robot rotates about wheel 3 and stops when it reaches a minimal potential energy position.

7 Conclusions

In this paper we present a mobile robot, designed for motion on slippery surfaces. Motion is performed by changes in the internal configuration of the robot, using passive rolling studded-like wheels. Kinematic model determines the required joints' velocities that steer the robot to a target position. Odometric model accurately determines the robot's position even in the presence of slippage. A method for evaluating the lateral slippage based on the odometric and kinematic models is presented. Gait patterns for motion up and down hills, as well as on horizontal surface are presented. Experimental results verify our models and slippage estimate, and show the reliability and accuracy of motion on slippery surfaces. Field experiments for the suggested gait patterns on various slopes and terrains have been carried out using our prototype model. In future work we intend to develop a dynamic model and investigate the effect of various terrain types on the suggested gait patterns.

References

1. P. Bidaud, R. Chatila, G. Andrade-Barroso, and F. Ben Amar. Modeling robot-soil interaction for planetary rover motion control. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 576–581, Victoria B.C., Canada, October 1998.
2. J. Borenstein and L. Feng. Gyrodometry: A new method for combining data from gyros and odometry in mobile robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 423–428, Minneapolis, Minnesota, April 1996.
3. S. Dubowsky and K. Iagnemma. Mobile robot rough-terrain control (rtc) for planetary exploration. In *Proceedings of ASME DETC/CIE: 26th Biennial Mechanisms and Robotics Conference*, Baltimore, Maryland, September 2000.
4. G. Endo and S. Hirose. Study on roller-walker (multi-mode steering control and self-contained locomotion). In *IEEE Int. Conf. on Robotics and Automation*, pages 2808–2814, San Francisco, CA, April 2000.
5. G. Ferretti, G. Magnani, G. Martucci, P. Rocco, and V. Stampacchia. Friction model validation in sliding and presliding regimes with high resolution encoders. In *Experimental Robotics VIII B. Siciliano and P. Dario Eds.*, pages 328–337. STAR Springer, Heidelberg, 2002.
6. C. Frederik, W. Heger, and V. Kumar. Design and gait control of a rollerblading robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 3944–3949, New Orleans, LA, April 2004.
7. K. Iagnemma and S. Dubowsky. Vehicle wheel-ground contact angle estimation: with application to mobile robot traction control. In *7th Int. Symposium on Advances in Robot Kinematics*, Piran-Portoroz, Slovenia, June 2000.
8. D. E. Koditschek. The application of total energy as a lyapunov function for mechanical control systems. In *J. Marsden, Krishnaprasad, and J. Simo, editors, Control Theory and Multibody Systems, AMS Series in Contemporary Mathematics*, 97:131–158, 1989.
9. G. Lafferriere and H. J. Sussman. A differential geometry approach to motion planning. In *Nonholonomic Motion Planning Z. Li and J. F. Canny Eds.*, pages 235–270. Kluwer, 1993.
10. S. Shimizu, K. Hasegawa, and T. Nagasawa. Alpine ski robot. *Journal of Robotics Society of Japan (JRSJ) special issue on Amusement Robot*, 8(3):126, June 1990.
11. K. Yoshida and H. Hamano. Motion dynamics of a rover with slip-based traction model. In *IEEE Int. Conf. on Robotics and Automation*, pages 3155–3160, Washington D.C., May 2002.

Wheel Control Based on Body Configuration for Step-Climbing Vehicle

Daisuke Chugo¹, Kuniaki Kawabata², Hayato Kaetsu³, Hajime Asama⁴ and Taketoshi Mishima⁵

¹ The University of Tokyo, 2-11-16, Yayoi, Bunkyo-ku, Tokyo, Japan
`chugo@iml.u-tokyo.ac.jp`

² RIKEN (The Institute of Physical and Chemical Research), 2-1, Hirosawa, Wako-shi, Saitama, Japan `kuniakik@riken.jp`

³ RIKEN (The Institute of Physical and Chemical Research), 2-1, Hirosawa, Wako-shi, Saitama, Japan `kaetsu@riken.jp`

⁴ The University of Tokyo, 5-1-5, Kashiwanoha, Kashiwa-shi, Chiba, Japan
`asama@race.u-tokyo.ac.jp`

⁵ Saitama University, 255, Shimo-Ookubo, Saimata-shi, Saitama, Japan
`mishima@me.ics.saitama-u.ac.jp`

Summary. In our current research, we are developing a holonomic mobile vehicle which is capable of running over the step. This system realizes omni-directional motion on flat floor using special wheels and passes over the step in forward or backward direction using the passive suspension mechanism. This paper proposes a new wheel control method of the vehicle according to its body configuration for passing over the step. The developed vehicle utilizes the passive suspension mechanism connected by two free joints that provide to change the body configuration on the terrain condition. Therefore, it is required to coordinate the suitable rotation velocity of each wheel according to its body configuration. In our previous work, the vehicle motion during step-climbing was discussed and moving velocity of each wheel was derived. In this paper, we adapt these results to wheel control and derived rotation velocity reference of each wheel. The performance of our proposed method is verified by the computer simulations and experiments using our prototype vehicle.

Keywords: Omni-Directional Mobile System, Passive Linkage Mechanism, Step-Climbing, Wheel Control

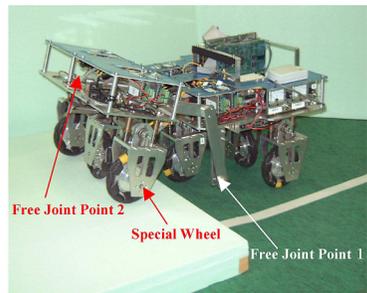
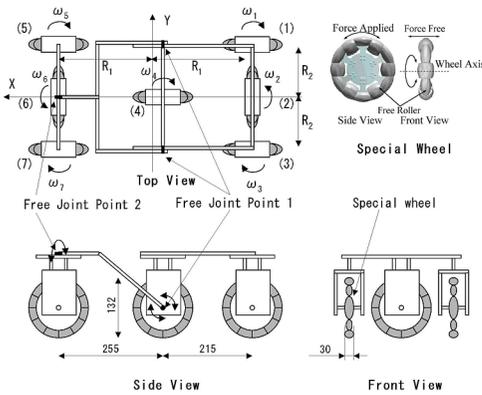
1 Introduction

In recent years, mobile robot technologies are expected to perform various tasks in general environment such as nuclear power plants, large factories, welfare care facilities and hospitals. However, there are narrow spaces with small barriers such as steps and the vehicle is required to have quick mobility

for effective task execution in such environments. The omni-directional mobility is useful for moving in narrow spaces, because there is no holonomic constraint on its motion. [1] [2] Furthermore, the step-overcoming function is necessary when the vehicle runs in the environment with barriers. In related works, various types of omni-directional mobile systems are proposed (legged robots, ball-shaped wheel robots, crawler robots, and so on). The legged robot [3] [4] can move in all directions and passes over rough terrain. However, its energy efficiency is not so good because the mechanisms tends to be complicated and the robot need to use its actuators in order to only maintain its posture. The robot with ball-shaped wheels can run in all directions [5], however, it cannot run on the rough grounds. The special crawler mechanism [6] is also proposed for the omni-directional mobile robot, but which can climb over only small steps. Therefore, there is still a lack of well-adapted mobile system for both narrow spaces and irregular terrain operation and we are developing a holonomic omni-directional vehicle with step-climbing ability. [7]

Our prototype mechanism consists of seven special wheels with free rollers (Figure 1) and a passive suspension mechanism. The special wheel equips twelve cylindrical free rollers [8] and applies the traction force only in advance direction. All special wheels are actuated and generate the omni-directional motion with suitable wheel arrangement and wheel control.

Furthermore, our mechanism utilizes new passive suspension system, which is more suitable for the step than general rocker-bogie suspensions. [9] [10] The free joint point 1 is in the same height as the axle and this system helps that the vehicle can pass over the step smoothly when the wheel contacts it in forward or backward direction. [7] No sensors and no additional actuators are equipped to pass over the barriers on the floor.



(a) Overview of the mechanism

(b) Prototype vehicle

Fig. 1. Our prototype mechanism

Our prototype has redundant actuators, therefore the vehicle controller calculates the control reference of each wheel based on the kinematic model [11] and controls each actuator to take the coordination among the wheels using PID-based control. [12] When the vehicle with passive linkages overcomes the step, the moving velocity of each wheel is different because of the change of the body configuration and its kinematic model. Therefore, it is required to modify the wheel control reference referring to the change of its body configuration. However, in many cases, fixed control reference which is derived without the consideration of body configuration is adapted and it causes the wheel slippage or rotation error. [13]

In our previous work, we developed PID based control scheme with coordination among the wheel [12] and derived its control reference referring to its body shape. [14] However, wheel control reference tends to become too extraordinary, especially during step-climbing, because it is derived from only vehicle's body configuration without consideration of the balance among wheel velocities. Too extraordinary control reference causes wheel slippage and rotation error. Wheel slippage disturbs the vehicle mobile performance and it is important to reduce it for maximizing traction force. [15] Therefore, it is required to derive the suitable wheel control reference not only referring the body configuration but also maintaining the balance among the wheel rotation velocities. In this paper, we developed the adjusting method of wheel control reference referring to the modification of the body shape for reducing wheel slippage and rotation error, and increasing the mobile performance.

2 Control System

2.1 Kinematics

Our vehicle has the passive suspension mechanism in its body and the body configuration changes according to the terrain condition when the vehicle passes over the non-flat ground. Therefore, it is required to modify the wheel control referring to its configuration.

In this section, we consider the relationship of rotation velocity vector of each wheel and the change of the body configuration on general passive linkage vehicle model. We assume that the vehicle has n passive linkages and all wheels have grounded and actuated. When the vehicle passes over the barrier as shown in Figure 2 (a), the velocity vector of wheel $i + 1$ is calculated by the velocity vector of wheel i and the rotation vector of wheel $i + 1$ in equation (1). These vectors are expressed by three dimensions in their local coordination system.

$$\mathbf{v}_{i+1} = {}^i\mathbf{v}_i + {}^i\sigma_i \times {}^i\mathbf{P}_{i+1}^i \quad (1)$$

where i is the number of wheel ($i = 1 \cdots n$), ${}^i\mathbf{v}_i$ and ${}^i\sigma_i$ are the velocity vector and the rotation vector of wheel i on the coordinate system i , respectively. ${}^i\mathbf{P}_{i+1}^i$ is the position vector from the wheel i to wheel $i + 1$ on coordination i .

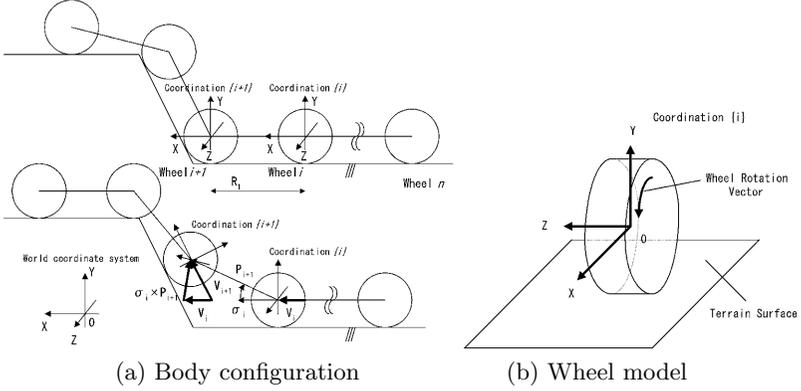


Fig. 2. Relationship between the velocity vector and the vehicle body

The coordinate system of each wheel is defined as shown in Figure 2(b).

- The x-axis is defined in the drive direction of the wheel.
- The y-axis is defined in the perpendicular direction to the ground.

Thus, the x-direction ingredient of the velocity vector in the coordination $\{i\}$ is derived as the control reference value. The control reference of the wheel $i + 1$ (ω_{i+1}) is derived from equation (2) and (3).

$$\omega_{i+1} = \frac{\|{}^{i+1}\mathbf{v}_{i+1}\|_x}{r} + \|{}^i\sigma_i\|_z \tag{2}$$

$${}^{i+1}\mathbf{v}_{i+1} = {}^{i+1}\mathbf{R} \cdot {}^i\mathbf{v}_{i+1} \tag{3}$$

$\|{}^{i+1}\mathbf{v}_{i+1}\|_x$ is x ingredient of the wheel $i + 1$ velocity vector, r is the radius of the wheel and ${}^{i+1}\mathbf{R}$ is the conversion matrix from the coordination $\{i\}$ to $\{i + 1\}$. Thus, when the velocity vector and rotation vector of wheel i are defined as ${}^i\mathbf{v}_i$ and ${}^i\sigma_i$, the control reference of wheel $i + 1$ is expressed as equation (4).

$$\omega_{i+1} = \frac{\|{}^{i+1}\mathbf{R} \cdot ({}^i\mathbf{v}_i + {}^i\sigma_i \times {}^i\mathbf{P}_{i+1}^i)\|_x}{r} + \|{}^i\sigma_i\|_z \tag{4}$$

All wheels have grounded, therefore we can assume that each wheel grounds the plane as shown in Figure 2(b). When the angle between the x-axis of coordination $\{i\}$ and the one of coordination $\{i+1\}$ is α , the conversion matrix is derived as equation (5). α fulfills the equation (6).

$${}^i{}^{i+1}\mathbf{R} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$\|{}^{i+1}\mathbf{v}_{i+1}\|_y = 0 \tag{6}$$

2.2 Adaptation to Our Prototype

In previous section, we discuss the general vehicle kinematics referring to the body configuration. In this section, we adapt it to our prototype vehicle and derive the velocity vector of each wheel. Our vehicle measures the change of body configuration using its attitude sensors and generates the wheel control reference with this information.

Our vehicle has two potentiometers on each passive joint and tilt sensors which are attached on the rear part of the vehicle body as shown in Figure 3(a). We can measure the following angles using these sensors.

- The roll angle θ_1 and pitch angle γ_1 from potentiometers.
- The roll angle θ_2 and pitch angle γ_2 from tilt sensors.

Our developing vehicle has 7 wheels and all wheels has actuated. Figure 1(a) shows the definition of the wheel number (We display as wheel i : $i = 1 \cdots 7$), the coordinates, the length of each links, and the rotate speed of each wheel, respectively. R_1 and R_2 indicate the length of each links and $\omega_1, \cdots, \omega_7$ are the rotation velocity of each wheel.

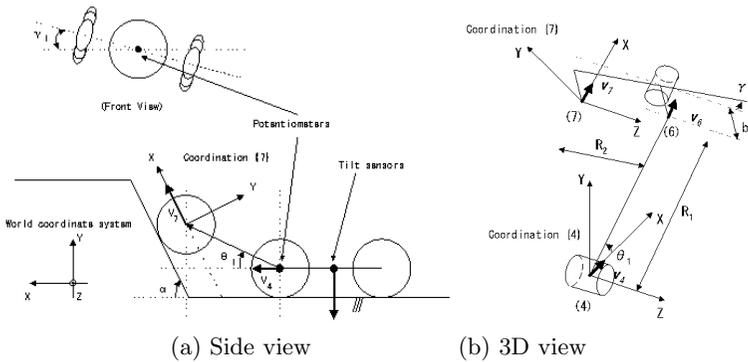


Fig. 3. Coordination and parameters of our prototype

When the vehicle runs at v_0 in x-direction on the coordination $\{4\}$, the velocity vector of wheel 7 on the coordination $\{4\}$ is derived as equation (7) by equation (1). The kinematic relationship among the wheels is shown in Figure 3(b).

$$\begin{aligned}
 {}^4\mathbf{v}_7 &= {}^4\mathbf{v}_6 + {}^4\sigma_6 \times {}^4\mathbf{P}_7^6 = ({}^4\mathbf{v}_4 + {}^4\sigma_4 \times {}^4\mathbf{P}_6^4) + {}^4\sigma_6 \times {}^4\mathbf{P}_7^6 \\
 &= \begin{bmatrix} {}^4v_{7x} \\ {}^4v_{7y} \\ {}^4v_{7z} \end{bmatrix} = \begin{bmatrix} v_0 + \dot{\theta}_1 \{-R_1 \sin \theta_1 + R_2 \cos \theta_1 \sin \gamma_1 - b \cos \theta_1 (1 - \cos \gamma_1)\} \\ \dot{\theta}_1 \{R_1 \cos \theta_1 - R_2 \sin \theta_1 \sin \gamma_1 + b \sin \theta_1 (1 - \cos \gamma_1)\} - \dot{\gamma}_1 (R_2 \cos \gamma_1 - b \sin \gamma_1) \\ -\dot{\gamma}_1 (R_2 \cos \theta_1 \sin \gamma_1 - b \cos \theta_1 (1 - \cos \gamma_1)) \end{bmatrix} \quad (7)
 \end{aligned}$$

As the same, the velocity vectors of wheel 1, 3 and 5 are derived from equation (8), (9) and (10), respectively.

$${}^4\mathbf{v}_1 = \begin{bmatrix} {}^4v_{1x} \\ {}^4v_{1y} \\ {}^4v_{1z} \end{bmatrix} = \begin{bmatrix} v_0 - \dot{\theta}_2 \{R_1 \sin \theta_2 + R_2 \cos \theta_2 \sin \gamma_2 - b \cos \theta_2 (1 - \cos \gamma_2)\} \\ \dot{\theta}_2 \{R_1 \cos \theta_2 + R_2 \sin \theta_2 \sin \gamma_2 - b \sin \theta_2 (1 - \cos \gamma_2)\} + \dot{\gamma}_2 (R_2 \cos \gamma_2 - b \sin \gamma_2) \\ \dot{\gamma}_2 (R_2 \cos \theta_2 \sin \gamma_2 - b \cos \theta_2 (1 - \cos \gamma_2)) \end{bmatrix} \quad (8)$$

$${}^4\mathbf{v}_3 = \begin{bmatrix} {}^4v_{3x} \\ {}^4v_{3y} \\ {}^4v_{3z} \end{bmatrix} = \begin{bmatrix} v_0 + \dot{\theta}_2 \{-R_1 \sin \theta_2 + R_2 \cos \theta_2 \sin \gamma_2 - b \cos \theta_2 (1 - \cos \gamma_2)\} \\ \dot{\theta}_2 \{R_1 \cos \theta_2 - R_2 \sin \theta_2 \sin \gamma_2 + b \sin \theta_2 (1 - \cos \gamma_2)\} - \dot{\gamma}_2 (R_2 \cos \gamma_2 - b \sin \gamma_2) \\ -\dot{\gamma}_2 (R_2 \cos \theta_2 \sin \gamma_2 - b \cos \theta_2 (1 - \cos \gamma_2)) \end{bmatrix} \quad (9)$$

$${}^4\mathbf{v}_5 = \begin{bmatrix} {}^4v_{5x} \\ {}^4v_{5y} \\ {}^4v_{5z} \end{bmatrix} = \begin{bmatrix} v_0 - \dot{\theta}_1 \{R_1 \sin \theta_1 + R_2 \cos \theta_1 \sin \gamma_1 - b \cos \theta_1 (1 - \cos \gamma_1)\} \\ \dot{\theta}_1 \{R_1 \cos \theta_1 + R_2 \sin \theta_1 \sin \gamma_1 - b \sin \theta_1 (1 - \cos \gamma_1)\} + \dot{\gamma}_1 (R_2 \cos \gamma_1 - b \sin \gamma_1) \\ \dot{\gamma}_1 (R_2 \cos \theta_1 \sin \gamma_1 - b \cos \theta_1 (1 - \cos \gamma_1)) \end{bmatrix} \quad (10)$$

On the other hand, the rotation vector of each wheel is derived using the roll and pitch angle as shown in equation (11) and (12). In equation (11), The rotation vectors of wheel 5 and 7 are same because these wheels are connected by same linkages. As the same, the rotation vectors of the wheel 1 and 3 are same in equation (12).

$${}^4\sigma_5 = {}^4\sigma_7 = [{}^4\sigma_{7x} \quad {}^4\sigma_{7y} \quad {}^4\sigma_{7z}]^T = [\dot{\gamma}_1 \quad 0 \quad \dot{\theta}_1]^T \quad (11)$$

$${}^4\sigma_1 = {}^4\sigma_3 = [{}^4\sigma_{3x} \quad {}^4\sigma_{3y} \quad {}^4\sigma_{3z}]^T = [\dot{\gamma}_2 \quad 0 \quad \dot{\theta}_2]^T \quad (12)$$

2.3 Derivation of Wheel Control Reference

In previous section, we derive the velocity vectors and rotation vector of each wheel when the vehicle runs at v_0 on the coordination $\{4\}$. However, these vectors are only calculated by kinematical relationship of wheels and the velocity vector of wheel, which passes over the step, tends to become extraordinary because of change of body shape. If we derive the wheel control reference by these extraordinary vectors using equation (4) simply, wheel control references become also extraordinary and cause wheel slippage and rotation error. In order to derive suitable wheel control references from results of previous section, we consider the following points.

- When the vehicle passes over the step at V_0 in advanced direction on the vehicle coordination as shown in Figure 1 (a), we set V_0 as the velocity in x-direction on the coordination $\{4\}$.

- All wheel velocity based on V_0 referring to body configuration must be smaller than V_0 .

Our proposed scheme is shown in Figure 4. When the vehicle passes over the step at V_0 on the vehicle coordination as shown in Figure 1 (a), we set the velocity in x-direction on the coordination $\{4\}$ as equation (13) temporary and we derive the velocity vectors of all wheels from equation (7)-(10).

$${}^4v_{4x} = V_0 \quad (13)$$

The coefficient c_i of wheel i is determined by equation (14).

$$c_i = \begin{cases} \frac{|V_0|}{|{}^4\mathbf{v}_i|} & \text{if } |{}^4\mathbf{v}_i| > |V_0| \\ 1 & \text{if } |{}^4\mathbf{v}_i| \leq |V_0| \end{cases} \quad (14)$$

where ${}^4\mathbf{v}_i$ indicates the calculated velocity vector of wheel i . i ($= 1, \dots, 7$) means sub-number for identification of the wheel.

The velocity vector on the coordination $\{4\}$ is determined by equation (15). Using equation (15), all wheel velocity vectors are calculated within the range of V_0 .

$$\mathbf{v}_i^{out} = c \cdot {}^4\mathbf{v}_i \quad (15)$$

where $c = \min \{c_1, \dots, c_7\}$.

Now, we derive the wheel control references from wheel velocity vector and rotation vector derived in previous section. When we set v_0 as the velocity vector \mathbf{v}_4^{out} of x-direction, the velocity vector of wheel i on the coordination $\{i\}$ is shown in equation (16). As shown in equation (5), we assume that the obstacle is the α -degree slope about each wheel as shown in Figure 3(a).

$${}^i\mathbf{v}_i = {}^i\mathbf{R} \cdot {}^4\mathbf{v}_i = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^4v_{ix} \\ {}^4v_{iy} \\ {}^4v_{iz} \end{bmatrix} = \begin{bmatrix} \cos \alpha \cdot {}^4v_{ix} - \sin \alpha \cdot {}^4v_{iy} \\ \sin \alpha \cdot {}^4v_{ix} + \cos \alpha \cdot {}^4v_{iy} \\ {}^4v_{iz} \end{bmatrix} \quad (16)$$

From equation (4) and (13), the control reference of wheel i is expressed in equation (17).

$$\omega_i = \frac{\cos \alpha \cdot {}^4v_{ix} - \sin \alpha \cdot {}^4v_{iy}}{r} + {}^4\sigma_{iz} \quad (17)$$

The α -degree is defined in equation (18), because the x-axis is defined in the drive direction of the wheel and the velocity vector is parallel to the drive direction as equation (6).

$$\sin \alpha \cdot {}^4v_{ix} + \cos \alpha \cdot {}^4v_{iy} = 0 \quad (18)$$

Our vehicle controls each wheel based on this control reference using PID based control system. [8]

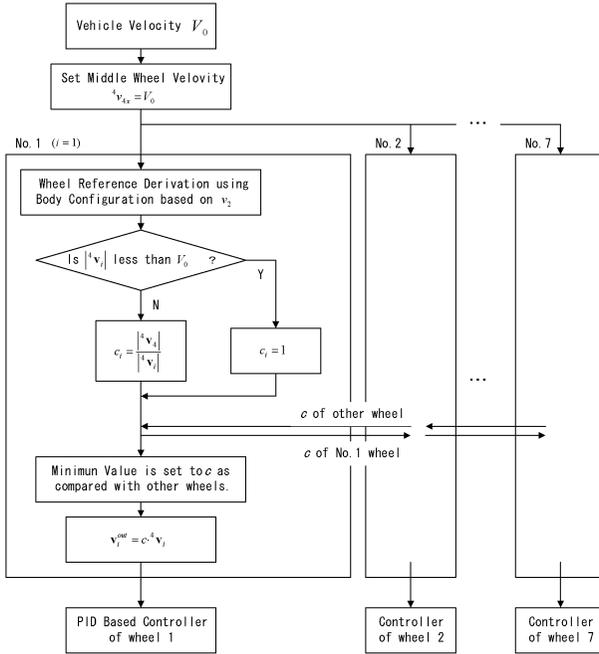


Fig. 4. Flow chart of the wheel control reference derivation

3 Experiments

3.1 Computer Simulation

We verify the effectiveness of our proposed control reference by computer simulations. We adapt our proposed control reference to test vehicle model and compare the result of proposed reference with the result of fixed reference which does not consider the body configuration. As initial conditions, simulation parameters of test vehicle model are chosen from our prototype model. The parameters are shown in Table 1.

Table 1. Parameters of prototype

| | |
|-------------------------|---|
| Number of Linkages | 2 |
| Length of Linkage | Front Part 195[mm], Rear Part 400[mm] |
| Body Weight | Front Part 7.8[kg], Rear Part 13.8[kg] |
| Wheel Diameter | 132[mm] |
| Distance between Wheels | Front-Middle 255[mm], Middle-Rear 215[mm] |
| Friction Coefficient | Static 0.3, Dynamic 0.25 |
| Running Speed (V_0) | 0.25[m/sec] |

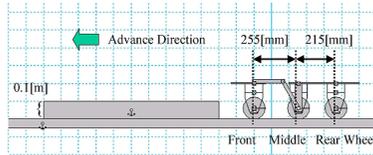


Fig. 5. Simulation setup

In this simulation, the vehicle passes over the step at advance direction as shown in Figure 5. The vertical gap of step is 0.1[m].

We use the Working Model 2D as a physical simulator and MATLAB as a controller. Working Model calculates the vehicle conditions dynamically using Kutta-Merson integrator and outputs the sensing data for MATLAB such as rotational velocities of each wheel and angle of free joint. MATLAB calculates the output value of each actuator using these values with our developed PID based controller [12] and returns the output value for Working Model. Both applications are linked by Dynamic Data Exchange function on MS Windows.

3.2 Simulation Results

As the result of the simulation, when the prototype vehicle passes over the 0.1[m] height step, control references of each wheel are derived as shown in Figure 6. These references are within the range of V_0 and we verify these control references are suitable.

Figure 7 shows the slippage ratio and Figure 8 shows rotation error ratio of the wheels during step climbing. The slippage ratio of the wheel decreases 54[%] and the rotation error ratio of the wheel decreases 55[%] by our proposed control method as shown in Table 2. The slip ratio and the rotation error ratio of wheel are calculated by equation (19) and (20), respectively.

$$\hat{s} = \frac{r\omega - v_\omega}{r\omega} \quad (19)$$

$$\hat{d} = \frac{\omega_{ref} - \omega}{\omega} \quad (20)$$

where ω is the rotation velocity of the wheel and ω_{ref} is the reference value of wheel rotation velocity. r and v_ω indicate the radius of the wheel and the vehicle speed, respectively.

From these results, our proposed method reduces the slippage and the rotation error of the wheels. Therefore, our proposed control method is effective for increasing the mobile performance of the vehicle during step-climbing.

3.3 Experiments

Here, we verify the mobile performance of proposed wheel control method by the experiments using our prototype. In this experiment, the vehicle passes

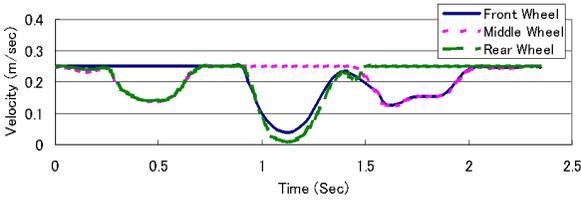
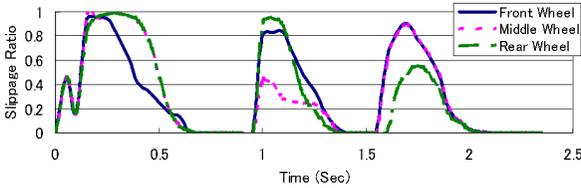


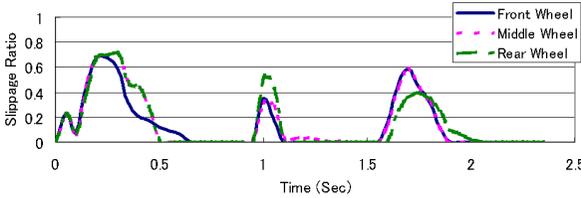
Fig. 6. Wheel control reference during step-climbing

Table 2. Slippage and rotation error ratio of wheel [%]

| | Method | Front Wheel | Middle Wheel | Rear Wheel | Average |
|----------------------|----------|-------------|--------------|------------|---------|
| Slippage Ratio | Standard | 31.3 | 30.3 | 29.6 | 30.3 |
| | Proposed | 13.1 | 14.3 | 14.2 | 13.9 |
| Rotation Error Ratio | Standard | 16.5 | 16.3 | 16.6 | 16.5 |
| | Proposed | 7.4 | 7.2 | 7.9 | 7.5 |



(a) With standard control method

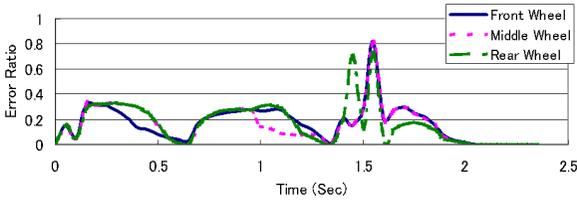


(b) With proposed control method

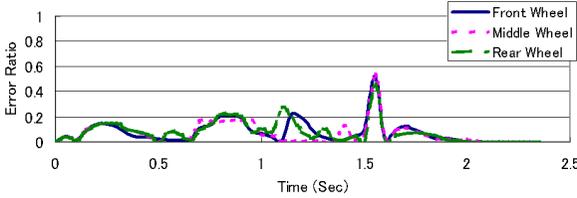
Fig. 7. Wheel slippage ratio

over the step in advance direction and we verify the height of the step which the vehicle can climb up. Experimental conditions are same as the prototype vehicle parameters shown in Table 1. We compare the results by our proposed method with the result utilizing standard PID controller, which doesn't consider the body configuration.

As the result of the experiment, the vehicle can pass over the 0.152[m] height step with our proposed wheel control method as shown in Figure 9. With standard method, the vehicle can pass over the only 0.072[m] height



(a) With standard control method



(b) With proposed control method

Fig. 8. Wheel rotation error ratio



Fig. 9. Passing over the 152[mm] height step

step. From this result, mobile performance of the vehicle increases using our proposed wheel control method.

4 Conclusion

In this paper, we propose the wheel control method of the vehicle according to change of its body shape during step-climbing. We discuss the kinematic model referring to the body configuration and adjusting method of the wheel control references when the vehicle passes over the step changing the body shape.

We verified the effectiveness of our proposed method by the computer simulations and experiments. Utilizing our proposed method, the slippage ratio and the rotation error ratio of wheels reduces when the vehicle climbs the step

and its step-overcoming performance is improved. As the results, our vehicle realizes 152[mm] height step-climbing performance with wheel of 132[mm] diameter. Its performance is useful for omni-directional wheeled vehicle. Our proposed wheel control method can utilize for the vehicle which has passive suspension mechanism.

References

1. G. Campion, G. Bastin and B.D. Andrea-Novel. Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots. In: IEEE Trans. on Robotics and Automation, Vol.12, No.1, pp.47–62, 1996.
2. M. Ichikawa. Wheel arrangements for Wheeled Vehicle. Journal of the Robotics Society of Japan, Vol.13, No.1, pp.107–112, 1995.
3. G. Endo and S. Hirose. Study on Roller-Walker: System Integration and Basic Experiments, In: Proc. of the 1999 IEEE Int. Conf. on Robotics & Automation, pp.2032-2037, 1999.
4. T. McGeer. Passive dynamic walking, The Int. Journal of Robotics Research, vol.9, No.2, pp62-82, 1990.
5. M. Wada and H. Asada. Design and Control of a Variable Footpoint Mechanism for Holonomic Omnidirectional Vehicles and its Application to Wheelchairs. In: IEEE Trans. on Robotics and Automation, Vol.15, No.6, pp.978-989, 1999.
6. S. Hirose and S. Amano. The VUTON: High Payload, High Efficiency Holonomic Omni-Directional Vehicle. In: Proc. of the 6th Symp. on Robotics Research, pp.253-260, 1993.
7. D. Chugo, *et al.* Development of omni-directional vehicle with step-climbing ability. In: Proc. of the 2003 IEEE Int. Conf. on Robotics & Automation, pp.3849–3854, 2003.
8. H. Asama, *et al.* Development of an Omni-Directional Mobile Robot with 3 DOF Decoupling Drive Mechanism. In: Proc. of the 1995 IEEE Int. Conf. on Robotics and Automation, pp.1925–1930, 1995.
9. Stone, H. W., Mars Pathfinder Microrover: A Low-Cost, Low-Power Spacecraft, In: Proc. of the 1996 AIAA Forum on Advanced Developments in Space Robotics, 1996.
10. Y.Kuroda, *et al.* Low Power Mobility System for Micro Planetary Rover Micro5. In: Proc. of the 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS99), pp.77–82, 1999.
11. Brian Carisle, An Omni-Directional Mobile Robot. Developments in Robotics 1983, IFS Publications Ltd., pp.79–87, 1983.
12. D. Chugo, *et al.* Development of Control System for Omni directional Vehicle with Step-Climbing Ability. In: Proc. of the 4th Int. Conf. on Field and Service Robotics, pp.121–126, 2003.
13. P. Lamon, *et al.* Wheel torque control for a rough terrain rover. In: Proc. of the Int. Conf. on Robotics and Automation, pp.4682–4687, 2004.
14. D. Chugo, *et al.* Vehicle Control Based on Body Configuration. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp.1493–1498, 2004.
15. K.Yoshida and H.Hamano. Motion Dynamic of a Rover With Slip-Based Traction Model. In: Proc. of the 2002 IEEE Int. Conf. on Robotics & Automation, pp.3155–3160, 2001.

Ball-Shaped Robots: An Historical Overview and Recent Developments at TKK

Jussi Suomela and Tomi Ylikorpi

Helsinki University of Technology, P.O. Box 5500, 02015 TKK, Finland
Jussi.Suomela@hut.fi, Tomi.Ylikorpi@hut.fi

1 Introduction

Sphere is "the set of all points in three-dimensional space lying the same distance (the radius) from a given point (the centre)." (Encyclopedia Britannica Online)

The shape of a sphere provides complete symmetry and a soft, safe and friendly look without any sharp corners or protrusions, which is advantageous when a robotic device is dealing with people. In terms of robotics, a spherical structure can freely rotate in any direction and all positions are stable. While the propulsion system is located inside the ball it can be hermetically sealed to provide the best possible shield for the interior parts. The spherical shape maximizes the internal volume with respect to surface area and provides an optimal strength against internal overpressure or underpressure, which is an important feature for underwater and space applications. The greatest technical challenges are its limited off-road capability and challenging controllability. Step-climbing capability is defined by the radius of the ball and the ratio of the masses of the cover and the unbalanced mass. Typically, the static step-climbing capability is less than $0.25 \times R$. The possibility of rotation in all directions makes the control of the ball challenging. Ball oscillation during the movement is difficult to handle, while the control system requires powerful actuators to compensate the oscillations.

Ball-shaped autonomously moving vehicles have a long history, and recent studies have described a variety of applications in different environments, including marine, indoors, outdoors, zero-gravity and planetary exploration. This article describes briefly the history of American patents of self-moving balls and recent developments carried out at Helsinki University of Technology (TKK) and elsewhere in the world.

2 History of Self-Propelled Movable Balls in View of U.S. Patents

Engineers are often advised not to invent the wheel again. However, a quick search of the U.S. Patent office database immediately reveals more than 50 patents related to the autonomous mobility of a ball-shaped object. These patents date from 1897 to 2003 and all comprise a motorized counterweight that is used to generate ball motion. Obviously, the number of related patents in the USA and worldwide is much larger than found by this quick search. The number of similar one-wheeled and two-wheeled counterweight-based vehicles is even larger.

The first vehicles were small spring-powered toys with one fixed axis of rotation. The patents concentrate on methods of storing and converting spring energy with different mechanical solutions. Adding steering capability to the toys has been a challenge from early times. In 1906, B. Shorthouse patented a design that offered the possibility of manually adjusting the position of the internal counterweight in order to make the ball roll along a desired curved trajectory instead of a straight path (U.S. Patent 819,609). Ever since, mechanisms have been patented to produce more or less irregular rolling paths for self-propelled balls. The toy shown in Fig. 1 dates back to 1909 and shows one innovative way of producing a wobbly rolling motion for an amusing toy.

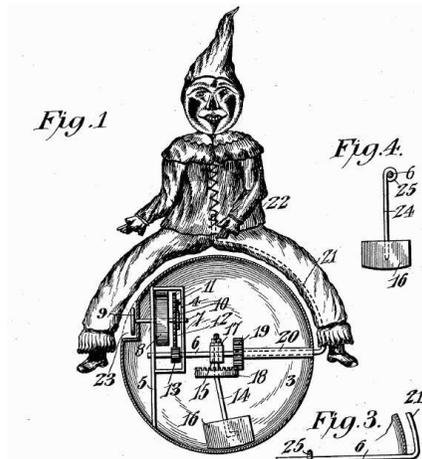


Fig. 1. Mechanical Toy by E.E. Cecil, (U.S. Patent 933,623)

The counterweight was usually constructed with a lever rotating around the ball's axis. Mobility was provided by generating torque directly to the lever. The amount of torque needed from the power system was directly proportional to the mass of the counterweight and length of the lever arm. In 1918, A.D. McFaul patented a "hamster-ball" design (a derivative of a ham-

ster running wheel), where the counterweight was moved by friction between the ball's inner surface and traction wheels mounted on the counterweight (Fig. 2, left). In this construction, the length of lever arm does not any more affect the required power-system torque, and similar mobility can be achieved with less internal torque. Obviously this is of great benefit to spring-driven toys, at least if they have a large diameter.

A mechanical spring as a power source was displaced by a battery and an electric motor in a patented design by J.M. Easterling in 1957 (U.S. Patent 2,949,696). Consequently, electric motors were introduced with several different mechanical solutions that were already at least partly familiar from earlier spring-driven inventions. Further development introduced shock and attitude sensing with mercury switches that would control motor operation and rolling direction, as well as adding light and sound effects.

An active second freedom for a motorized ball was introduced by McKeehan in 1974, as shown in Fig. 2, (right). In addition to reversible rolling motion, upon impact against an obstacle, the ball would also change its axis of rotation with the aid of additional motors. This opened the way towards radio-controlled (introduced in 1985 in U.S. Patent 4,541,814) and, finally, computer controlled, ball-robots. As (radio-controlled) toy-cars became more common following 1984, they were frequently inserted inside the ball to provide a fully steerable 2-dof. rolling toy (U.S. Patent 4,438,588).

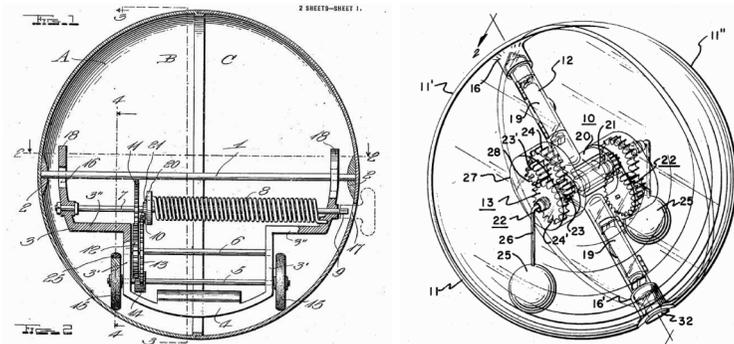


Fig. 2. (Left) Early "hamster-ball" by A.D. McFaul, (U.S. Patent 1,263,262). (Right) A 2-dof. ball by R.W. McKeehan, (U.S. Patent 3,798,835).

Spherical vehicles to carry people were first developed for marine applications, like the one of W. Henry in 1889 (Fig. 3, left). This vehicle, with its passenger floating in the water, was balanced by ballast mass and the weight of the passenger. The vehicle would move in a manner very similar to the toys described above with balanced mass inside and with their outer surface rolling. Steering would be achieved by tilting the axis of rotation by moving the passenger mass inside the vehicle. In 1941, J.E. Reilley patented a ball-shaped car (Fig. 3, right) and later different types of chairs were inserted

inside the spherical vehicle. In some cases, a person would enter a ball and operate it directly without any additional means, like a hamster inside his running wheel.

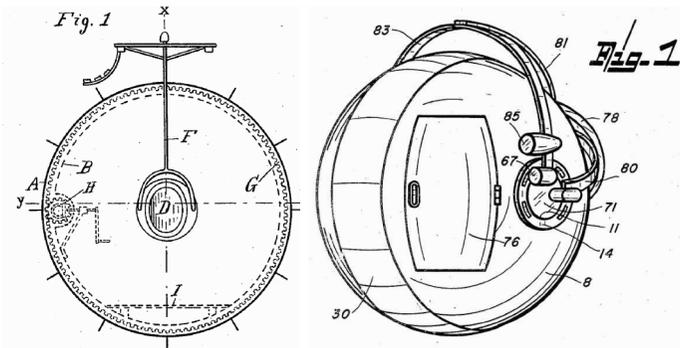


Fig. 3. (Left) A marine vessel by W. Henry (U.S. Patent 396,486). (Right) A Spherical vehicle by J.E. Reilley (U.S. Patent 2,267,254).

The most recent inventions introduce new novel solutions to alter the position of the ball's centre-of-gravity. One example is the Spherical Mobile Robot by R. Mukherjee, patented in 2001, that uses several separate weights that are moved with the aid of linear feed systems (U.S. Patent 6,289,263).

3 Fascinating Shape of a Sphere

The apparent large number of patents (and obviously a much larger number of patent applications) raises the question: "Why is the mobile ball so popular?" One aspect is visible already in the names given for the device, such as "self-propelled toy", "magic ball", "squiggle ball" and "randomly self-propelled spherical toy". The power source and principle of action of the device is not obvious and it appears to move by itself, like a living thing. Apparent random action increases the fascination of people watching the ball moving around. This fascination is real and of a great importance when the ball is turned into a useful robot that operates together with people at home. Another aspect can be seen in the name "self-propelled continuously moving toy", which indicates that this device never stops. Partly because of its design, and partly through added functions, this ball-shaped vehicle has an amazing capability to back-off and change direction upon contact with an obstacle. It is hard to get this thing jammed, except in unsuitable terrain that could be too resistive or too slippery. By definition, the ball is round; it has no corners and its outer dimensions are similar in all possible directions. Wherever it enters, it can also exit, regardless of its orientation. This being so, the robustness of the ball in real environments is excellent.

4 TKK Family of Ball-Shaped Robots

During recent years, the Automation Technology Laboratory of Helsinki University of Technology has developed several different ball-shaped robots. It may be stated that the laboratory has a special interest in ball-shaped devices. Although these robots have been developed for several purposes and their functioning is very different, they have, however, a common feature - their ball-shaped structure - which originates from their common need for mobility. Although the ball is not easy to guide, and its kinematics and dynamics are quite challenging, the ball-shape provides a robust and fault-tolerant envelope. It has also been observed that the spherical shape fascinates the general public, which is a fact of great importance when the robot is to be operated with human beings in a common environment. The robots are presented below in chronological order.

4.1 Submar

Submar is a robotic member of a multirobot society of several small ball-shaped floating robots. The robots can be inserted inside a fluid circulation system of, for example, a paper factory. There, the robots would move along the fluid circulation and measure several properties of the fluid itself or of piping containing it. Submar robots measure the internal state of the process and can perform small tasks such as injecting reagent and taking samples. The robots move semi-actively along the process flow, changing their vertical position through the use of an internal pump and diving tanks (see Fig. 4). A pressure sensor gives information for vertical positioning.

The aim of this project was to create a multirobot system, i.e. a robot society operating in a 3D process environment. Members of the society communicate with each other and so provide an overall view of the state of the complete fluid circulation system. The benefits of this robot society concept are fault tolerance, flexibility and simplicity. The society structure compensates the disadvantages of the partly random behaviour of the robots and of their incomplete mobility capabilities. The spherical shape provides the advantage of obstruction-free mobility.

A third generation prototype of Submar has a 10.8-cm outside diameter. A 0.5W DC motor controls functions such as the diving tank and chemical tank. Typical sensors will measure internal and external temperatures, conductivity, pressure and tilt. The CPU is a Siemens 80C166, 32 MHz, with a 128 kbytes flash EPROM memory and 128kbytes SRAM. Communication between robots and with the operator is provided by a Radiometrix BIM, 9600 baud, 433 MHz.

The Submar robots have a wide range of possible applications, including various industrial processes, water purification plants, and environmental monitoring cases. The novel feature in Submar sensor/actuator robots is that they can follow the state of a process from the inside.

One possible use of Submar robots in environmental monitoring is to anchor it to bottom of a sea or a lake. The robot would then move up and down along the rope, measuring interesting parameters. Measured values would be either saved in Submar's memory or sent by radio to the operator. Another method might be to let a herd of robots drift along a river flow collecting data and catching them in a net at the end of the experiment. [1] [2]

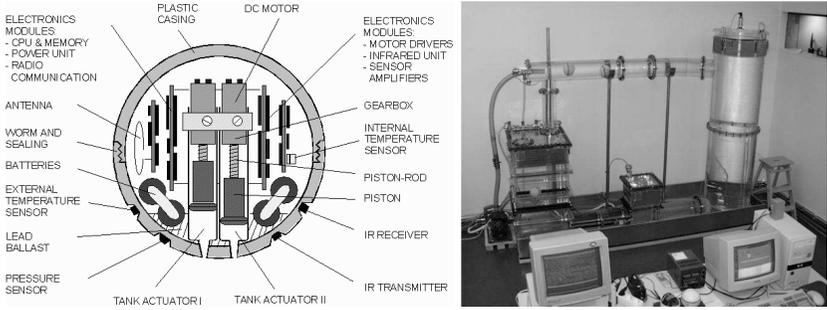


Fig. 4. Submar construction and test environment. (TKK)

4.2 Rollo

Rollo is a motorized 2-dof. spherical robot intended to operate at home with people. It can act as a real mobile telephone, event reminder and safety guard. Rollo has also performed live on-stage in a theatre in Helsinki. During its development, several different mobility systems have been developed. Fig. 5 illustrates the three generations of Rollo-robot. The first two generations were of the "hamster-ball" type, while the third is based on a rotational rim gear carrying a rolling axis. An instrument body hangs suspended on the rolling axis.

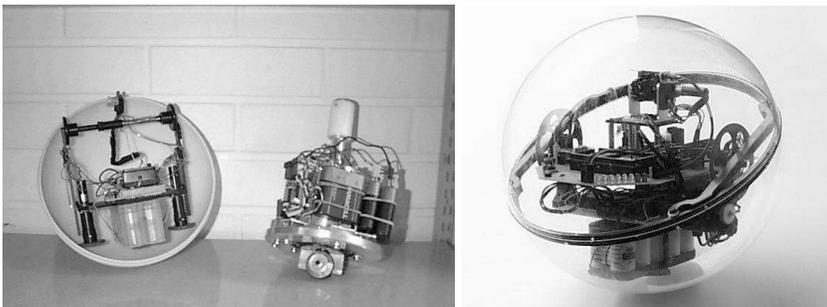


Fig. 5. 2nd, 1st and 3rd generation of the Rollo. (TKK)

The radio-controlled second generation hamster ball was provided with two completely independent freedoms and full steerability with easy control, but required an expensive spherical cover that was accurate and fitted precisely. The third generation allows a softer and transparent cover easily available from industry. The kinematics and dynamic behaviour of this design are quite challenging. Rolling direction is selected by turning the rolling axis along the rim gear, which must then lie in the horizontal position. However, during rolling, the rim gear also rotates around the axis and there are only two positions where the robot can select the rolling direction (i.e., the rim gear lies horizontally). In these two cases, a similar motor rotation yields to opposite directions of rotation along the rim gear. The robot always has to advance a full number of half-revolutions, after which it needs to determine which direction along the rim gear is the correct one. Revolutions of the rim-gear are counted by means of an inductive sensor. Continuous steering of the robot is also possible in theory, but in practice it would be a very demanding task.

Contrary to the Rollo-robot, the Roball, and the Rotundus and The This-tle ([6], [9], [4]) for example, are able to use continuous steering. In these designs, the rolling axis is not rotated in a horizontal plane to choose the rolling direction, but is tilted sideways to make the ball adopt a curved trajectory. This is in nature a dynamic behaviour, which has been studied in relation to Roball-development ([6]).

Another challenging property of the hard-surfaced, unbalanced ball on a smooth floor is that it behaves like a pendulum. Any change in motor torque (acceleration, deceleration), or disturbances from its surroundings, easily make the ball to oscillate; this attenuates very slowly. Oscillations around the rolling axis are controlled in Rollo with a closed-loop system that controls the drive-motor torque. The control loop is equipped with attitude sensors and gyroscopes that measure forward and backward motion of the payload mass. A more difficult task is to control the sideways oscillation, since we do not possess any actuators in this direction. So far, no active instrumentation has been included for this, but, in future, passive dampers or an active closed-loop controlled movable counter-weight or pendulum may be considered.

The Rollo-robot is equipped with a camera, microphone and a video link. Communication to the control station is achieved using a radio modem. The robot is equipped with a Phytec MiniModul-167 micro controller board using a Siemens SAB C167 CR-LM micro controller. The robot has sensors for temperature, pan, tilt and heading of the inner mechanics and pulse encoders for motor-rotation measurement. The local server transmits controls to the robot using commands that are kinematics invariant (i.e., they use the work-environment variables only). The commands include heading, speed and running time/distance. Coded graphical signs mounted on the ceiling are utilized by means of the on-board camera to determine absolute robot location when necessary. The system has an automatic localization command, which causes the robot to stop, wait for some time to smooth out oscillations, turn

the camera to the vertical position, find the visible beacons and automatically calculate the position, which is then returned to the control station.

The robot can be programmed as an autonomous device or it can be teleoperated via the Internet. The user interface contains a virtual model of the remote environment where the video input and virtual models are overlaid to produce the augmented reality for robot guidance. Augmented reality provides an efficient medium for communications between a remote user and a local system. The user can navigate in the virtual model and subsequently use it as an operator interface.

As one application, an educational system has been developed for virtual laboratory exercises, which university students can do over the Internet. The overall experimentation system includes versatile possibilities to set up interactive laboratory exercises from an elementary level to more advanced levels. Topics include mechatronics, robot kinematics and dynamics, localization and navigation, augmented VR-techniques, communication systems and Internet-based control of devices.

A second application, The Home Helper system, provides a mobile multimedia platform for communications between home and outside assisters. The system is connected to various networked devices at home. The devices provide possibilities for remote security surveillance, teleoperation of the devices, and interactive assistance to people living at home [3].

4.3 The Next-Gen Robot Society

The Next-gen Robot Society consists of a mother robot and several small ball-shaped robots shown in Fig. 6. The aim of the project is to realize a functional robot society of several robots autonomously changing information and working together. A practical application of the system might be a "Night Watchman" that involves the agents of the multi-robot system patrolling the corridors of the laboratory and asking people moving around there for identification. The small robot contains a novel mechanism that changes the robot shape from a ball into two separated hemispheres. Simultaneously, as the hemispheres separate, a mechanical tail extends from its interior to provide better balance under motion. The kinematics of this robot does not present a ball but a two-wheeled rover, although it can be driven also in ball-shaped configuration. In the latter case, steerability would be limited, though. As this robotic system is also intended for indoor operation with human beings, the spherical shape provides a friendly look. Further, the ball shaped rovers can be handled and stored in a robust way by the mother robot that will serve as an energy refilling station and, when needed, giving a piggyback ride for some or all of them.

Communications form the basis for any multi-robot system. The robots are able to form a network by themselves, without the aid of an operator. This requires dynamic network formation and management. Each robot will also have a sufficient level of autonomy to remain in operation when they go out

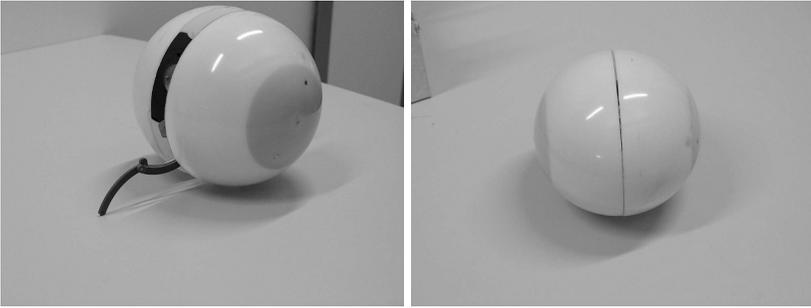


Fig. 6. A member of the Next-gen Robot Society, closed and open. (TKK)

of communications range. Communications between the human operator and the robots can either be achieved through the mother robot or by contacting the robot member directly.

In the robot are two processors, a phyCORE 167CR, functioning as the main controller and having an Infineon 16-bit C167CR microprocessor. It also features several communication channels, including two 10/100 Mbit Ethernet connections, a USB 1.1 host, two serial ports and a compact flash-card slot housing a Bluetooth or WLAN communications module. A CMOS camera is connected to the USB host port. It is used, among other things, in recognition of objects and optical flow measurements. Three VTI SCA 610 accelerometers / inclinometers are used for inclination and acceleration measurements. A silicon ring gyroscope is used to monitor the angular speed of the robot, to stabilize its driving and keep track of the current angle and the relation to the world coordinates.

A real challenge with the project was to get all the things to fit inside the 16-cm ball, including the drive mechanics and separation mechanism. The drive mechanism utilizes two independent motors. The separation mechanism takes actuation from the other drive motor via a one-way clutch. This way, we were able to reduce the number of the motors, with the limitation that the other motor may run only in one direction during normal operation. When the motor running direction is changed, the mechanism extends 25 mm along with in-built linear guides. The opening is realized by a special shuttle-device that automatically closes the structure again when the motor rotation is continued. A 5-bar linkage is operated by the motion of separate body parts and extends a two-piece tail to give further support for the device upon rolling. This support is needed to allow higher motor torque when over-passing obstacles like carpet edges or door steps. The tail can also be used for automatic connection to a battery charging station on-board the mother vehicle.

The opening action of the robot requires careful positioning of the masses inside. The system must be stable both in closed and open positions and the camera must have an unobstructed view in the open position.

4.4 Thistle

This study, funded by the European Space Agency under ARIADNA-program, focused on new innovations derived from nature to develop a novel system to provide a robust and efficient locomotion system to be used for exploring foreign planets. Thistle is a large low-mass wind-propelled ball inspired by the Russian Thistle plant, as shown in Fig. 7. The 1.3-meter ball represents a model of a larger 6-meter version that was proposed to operate on the surface of Mars for autonomous surface exploration. In order not to be fully dependant on occasional wind-energy, the Thistle was equipped with a 2-dof. drive system that provided full steerability and motorized locomotion.

Without additional equipment and mass, assuming a low drag coefficient due to the open structure of the ball, a terrestrial 5 m/s wind is supposed to propel the roughly 4-kg prototype over obstacles 10 cm high. When actively driven by the motorized 5.1 kg ballast mass we expect the prototype to roll over 4.3 cm obstacles. Driving tests with the Thistle show that locomotion is quite clumsy and somewhat chaotic. Structural flexibility and sectional circumference make the ball proceed in short bursts. If a tilt angle is introduced by means of the steering system, during rolling the Thistle follows a spiral-like path in which the radius of curvature decreases towards the end of the motion. In practice, it is possible to make the Thistle turn in a very limited space, but controlling the length of the turn is more difficult.

The torque margin of the drive system allows the ballast mass to be rotated a complete revolution around the axis of rotation. This means, that when the Thistle stops against an obstacle, the ballast mass finally travels over the upper dead centre and, in consequence, the Thistle autonomously backs off by half revolutions. Due to instability, the Thistle also, usually simultaneously, turns slightly. This behaviour enables the Thistle to circumvent obstacles autonomously, and without any active steering. The Thistle was also tested on a snow bed during Finnish winter conditions. The soft structure of the snow effectively damped out the structural vibrations of the Thistle, while driving and steering was clearly easier and overall behaviour was more predictable.



Fig. 7. The tumbleweed plant and inspired Thistle rover. [4]

5 Other Recent and Related Development

In addition to robots presented, there are several other similar devices, mostly intended for demonstration or simply for toys. Some recent and related developments are shown in Fig. 8. The 1.5-meter diameter scale models of the Tumbleweed Rover and Windball are intended for Mars exploration. Both of them are purely wind-driven, the only mobility-related actuation being re-shaping the structure by inflation/deflation (Tumbleweed) or with the aid of shape-memory alloys (Windball). On Mars, 6-meter versions of these models would be used to carry out scientific tasks like surface mapping and atmospheric measurements. The 15-cm Roball performed an important role in a study of interaction between the robot and small babies. It is anticipated that the 15 cm Cyclops and 50 cm Rotundus will be used to inspect and guard industrial plants. The Sphericle is used as an educational tool for learning dynamics and control of a ball-shaped robot. The Gyrorover of Carnegie Mellon University, although not acting as a ball but a wheel, is stabilized and steered by means of an internal, mechanical gyroscope. [11] This approach might very well also be adapted for ball-shaped robots.

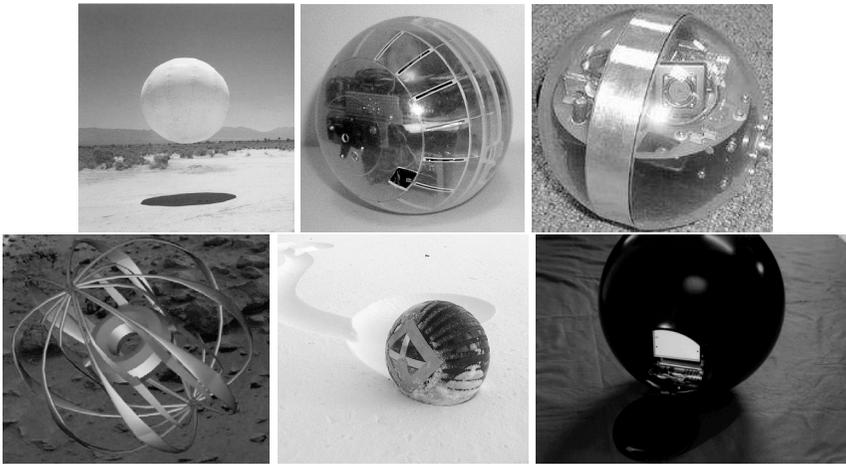


Fig. 8. Top Row: Tumbleweed Rover, Roball Rolling Robot and Cyclops. [5], [6], [7]. Lower Row: Windball, Rotundus and Sphericle. [8], [9], [10].

6 Conclusions and Future of Ball-Shaped Robots

Throughout history, ball-shaped toys have been quite popular and they do still exist. Development in computer technology, wireless data transfer and digital cameras has given them many advanced operational capabilities. The autonomous ball-shaped robots are being introducing back into modern homes,

this time, not only as toys, but also as serving and guarding robots. Future work in this field will concentrate on analyzing and developing the dynamics and control of the ball, as well as on applications and interaction with environment and people.

The utilization of large wind-propelled balls for Mars-exploration has been widely studied in many separate institutions. The main advantage is the large size, low mass and autonomous mobility; this accompanies the disadvantage of limited steerability. Although the balls appear efficient in covering large distances, a common requirement of full controllability is unlikely to be met in these ambitious scenarios. It would require exceptional open-mindedness to include a Windball, Tumbleweed or Thistle even as a piggy-pack along with other Mars-exploration instruments. Only the future will tell if the current expensive exploration missions will be followed by low-cost autonomous missions utilizing these more simple technologies.

References

1. Vainio, M. (1999) Intelligence through interactions - underwater robot society for distributed operations in closed aquatic environment. Espoo, Teknillinen korkeakoulu, 131 p.
2. Appelqvist, P.(2000) Mechatronics design of a robot society - A case study of minimalist underwater robots for distributed perception and task execution. Espoo, Teknillinen korkeakoulu, 86 p.
3. Wang, Y. and Halme, A. (1996) Spherical rolling robot, Espoo, Teknillinen korkeakoulu, 16 p.
4. Peter Jakubik, Jussi Suomela, Mika Vainio, Tomi Ylikorpi, (2004) Biologically inspired solutions for robotic surface mobility. ARIADNA AO4532-03/6201 Final Report, Helsinki university of technology, Finland.
5. Low Cost Mars Surface Exploration: The Mars Tumbleweed. NASA/TM-2003-212411, August 2003.
6. Francois Michaud and Serge Caron (2001) Roball, the Rolling Robot. LABORIUS - Research Laboratory on Mobile Robotics and Intelligent Systems, Universite de Sherbrooke, Canada.
7. Brian Chemel, E. Mutschler, H. Schempf Cyclops: Miniature Robotic Reconnaissance System. Field Robotics Center, Robotics Institute, Carnegie Mellon University, Pittsburgh, USA.
8. Moritz Heimendahl, Thomas Estier, Pierre Lamon, Roland Siegwart, (2004) Windball. Swiss Federal Institute of Technology Lausanne, Autonomous Systems Laboratory.
9. Rotundus AB, c/o UUAB, Uppsala Science Park, SE-751 83 Uppsala, Sweden.
10. A. Bicchi(*), A. Balluchi(*), D. Prattichizzo(*), A. Gorelli(**) (1997) Introducing the Sphericle: an Experimental Testbed for Research and Teaching in Nonholonomy, (*)Centro E. Piaggio, Universita di Pisa, Italia, (**) Facolta di Ingegneria, Universita di Siena, Italia.
11. S. Tsai, E. Ferreira, and C. Paredis, Control of the Gyrover: A Single-Wheel Gyroscopically Stabilized Robot, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99), Vol. 1, October, 1999.

Development of a Water-Hydraulic Self-Propelled Robotic Drill for Underground Mining

Michael N. Wendt¹ and Garry A. Einicke²

¹ CSIRO Australia, PO Box 883, Kenmore, QLD 4069 Michael.Wendt@csiro.au

² CSIRO Australia, PO Box 883, Kenmore, QLD 4069 Garry.Einicke@csiro.au

Summary. The design and control of high power water hydraulic robotic systems presents unique challenges. This paper describes some of those challenges and the techniques used to overcome them to develop a unique system for an autonomous robotic drilling machine for underground mining.

Keywords: Mining Water-Hydraulics Electrohydraulic

1 Introduction

A robotic self-propelled underground drilling system is under development that uses high-pressure potable water to provide power to drive hydraulic systems for cutting and propulsion. The drill string, that normally provides both torque resistance, fluid flow and thrust is replaced by a flexible hose with a self-propelled down-hole motor. In operation the drill is a small scale (0.1m diameter) version of a tunnel boring machine except that friction with the wall is used to create thrust rather than pushing from a casing. Removal of the drill string enables a tight radius of turning and provides greater flexibility of the system to optimise hole trajectories.

Water, alone or with additives, is commonly used in drilling to provide power to a downhole motor. The fluid is subsequently dumped into the hole to recover the cuttings and return them to the surface. With a self-propelled system, water is also needed to power hydraulic clamping, thrusting and steering systems before being dumped into the hole. Although essential for this system, water hydraulics for power transmission is making a return to modern hydraulic systems [1,2] because of its advantages as a power transmission fluid, i.e., low cost, environmental friendliness, high system stiffness, fire safety and chemical neutrality.

2 Overview of Design

A drawing of the machine is shown in Figure 1. A cutting head is attached via a high-torque, compact universal joint to a shaft which travels the length of the machine and is splined into a water-powered electronically controlled wobble plate motor located at the rear. Thrust and steering is performed by three axial cylinders located in the main body 120° apart. Torque and thrust from the cutting head is transferred from the motor to the walls via a set of clamps located near the centre of the machine. Water is pressurised by a positive displacement pump located on the surface and supplied to the drill by a flexible hose attached to the rear of the unit together with communications and electronics power lines. Water is used both as the working fluid for the motor and hydraulics as well as the transport medium to flush cuttings to the surface.

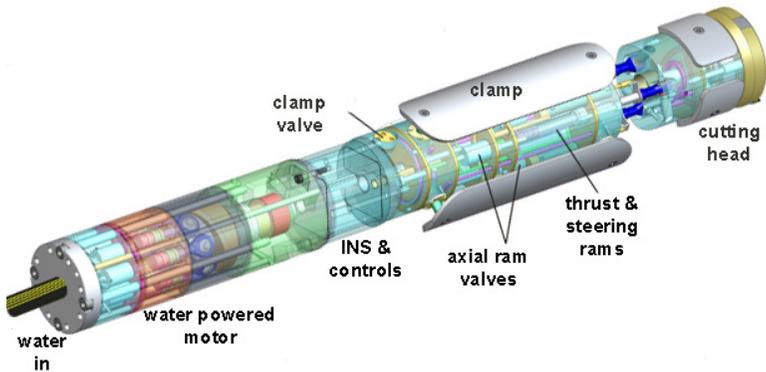


Fig. 1. Overview of robotic drill.

The operation of the machine occurs in four stages. Firstly, the rear clamps engage the wall and the front clamps are depressurised. Secondly, the three axial cylinders are pressurised to provide thrust for the cutting bit. Steering is enabled by differential pressure on the axial cylinders causing the drill to veer toward the cylinder with the lowest force. Thirdly, once the axial rams are fully extended, the front clamp is engaged and pressure is released to the rear clamps. Pressure on the other side of the three axial rams is then used to drag the hose and body forward to the next position. Finally, the front clamps are disengaged, the rear clamps engaged and the cutting cycle recommences. The cutting head runs continuously throughout the operation. The reverse process can be used to back out of the hole if the hole prematurely collapses during retrieval with water jets from the motor to assist with displacement of any material that may remain in the hole. Otherwise, tension on the hose is used to retrieve the drill.

2.1 Water Hydraulics

The difficulties associated with water as a power transfer medium include its poor lubrication properties due to low viscosity, its low compressibility and its potential for corrosion. These properties lead to issues with leakage of seals and valves, a propensity for cavitation and water hammer and a restriction on material selection to prevent corrosion. Low viscosity generally makes it difficult to use conventional proportional control valves due to the tolerances needed to reduce leakage especially at the very high pressures (34MPa) used. Instead, bi-state valves with a metal-to-metal seal are employed here.

Two different designs are incorporated within the drill for the design of the hydraulic cylinders and pistons. The axial pistons use 316 stainless steel for both the piston and bore material, whilst the clamps use aluminium bronze pistons and stainless bores. Although adequate corrosion resistance is provided by 316, its wear properties are poor and it is essential that two surfaces not touch. Wear bands, made from bronze impregnated PTFE are added to the piston on either side of the double acting seal. Due to the short length and lower speed of the clamp pistons, a solid aluminium bronze piston was used to provide the bearing surface. A drawing of both pistons are provided in Figure 2. The seal used for the axial rams was an 'O' ring backed PTFE band to ensure suitable performance at higher speeds. A bronze impregnated PTFE piston was used for the clamp piston seal. PTFE bearings were also used to support and seal the rod. A Nitrile wiper was used to keep cuttings from entering the ram.

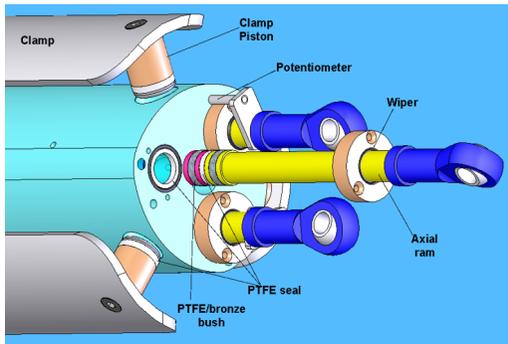


Fig. 2. Detailed piston assembly.

2.2 Construction

The drill is constructed as a series of sections sliced across the axial plane. Seals between each surface separate the multiple flow passages that contain the high pressure supply water, control pressures, lubricating oil, electronics and shaft.

These seals also exclude high external pressure water and rock slurry from entering the machine. These seals take two forms: a) protruding edges (0.2mm high by 0.5mm wide) are machined onto the surface and these act on copper gaskets, or b) 'O' rings with a compression to 96% of the groove are used between lapped surfaces to minimise the extrusion gap. High strength bolts are used to clamp the entire assembly together. Materials for construction consist primarily of 316 stainless steel as this provided both adequate corrosion resistance and strength. Inner components of the motor are made from EN26 steel which has high strength and surface hardness. This material has low corrosion resistance but the components are completely immersed in oil for lubrication and corrosion protection.

3 Electrohydraulic Wobble Plate Motor

While some novel water powered motors [3] and other commercially available units have been designed [4], these units are not capable of running at the high pressure (34MPa) required by the drill or have the necessary controllability within the motor for torque and speed. The motor design needed to provide high torque at low speeds and high power in a compact package. It was also desirable to limit the overall length (approximately 300mm) to allow the drill to turn in a tight radius. These requirements together with the low viscosity and the high pressure of the water, led to the use of a positive-displacement piston-based motor. To simplify the design, it was also desirable to use the same seal type, wear bands, valving, control electronics and software for the motor as that used for the hydraulic propulsion systems.

There are basically three types of piston powered hydraulic motor; swash plate, crankshaft and radial piston. However, due to the desire for a compact cylindrical machine, of the three designs, only the swash plate is suitable. Due to the poor lubrication properties of water, a significant redesign of the swash plate style motor was needed, in particular the shaft thrust bearing, rotating ports and the swash plate bearing. In a conventional swash plate oil powered motor, the oil is used to lubricate all of these surfaces to reduce wear. High pressure oil travels through the centre of the pistons and lubricates the piston, seals and base shoe which runs over the swash plate. Oil is also trapped by a hydrodynamic bearing on the top of the rotating cylinder block to lubricate this surface to provide porting.

To solve these issues a new type of motor was developed (Figure 3) which is different from conventional designs in a number of ways. Firstly, the lower half of the motor, which contains the shaft axial and thrust bearing, is immersed in oil. Water flow is restricted to the tops of the pistons only and is sealed with the same seal used for the body hydraulic rams. A second seal is provided below the first to prevent water that may pass the first from entering the oil filled bearing chamber. A vent to outside is provided between the two seals.

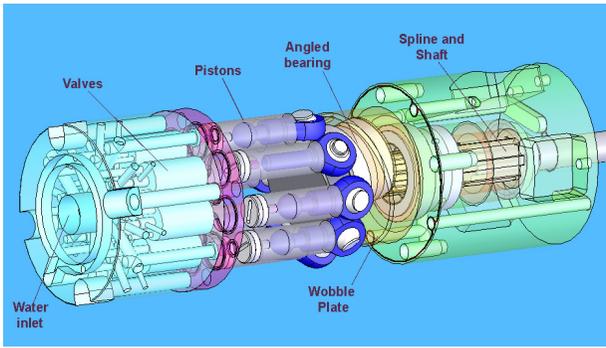


Fig. 3. Detailed motor assembly.

The second major problem is caused by the bearing surface between the cylinder block and port area. This area must rotate to provide the swash plate action and to open and close the ports for the inlet and exhausting of fluid. The third problem is the high pressure oil that normally lubricates the shoe. To solve both these problems, the cylinder block is fixed in place and the swash plate was attached directly to the shaft via an angled bearing. Instead of shoes sliding on the swash plate the load from the piston is transferred to the angled bearing via a combination of two spherical rod end bearings sliding on stubs and six double spherical linkages. Axial force from the piston acts on the outer race of the inclined bearing causes the inner race to rotate and hence drive the shaft.

3.1 Motor Sequencing Electronics and Operation

A consequence of opting for a fixed cylinder block is the provision of porting to the cylinders. Previously, the inlet and outlet ports were sequentially opened and closed as slotted ports were uncovered. This is beneficial as the porting area no longer needs to be lubricated, however, some form of porting system needs to be implemented. This was solved by supplying a solenoid valve to each individual piston and to sequence their operation electronically to simulate the normal operation of the motor. This means those pistons pushing on the down side of the internal swash plate bearing are being supplied pressure and those on the upper side are being vented to outside.

A typical valve control sequence is shown in Figure 4. The sequence shows that four of the valves, which are located on the down stroking side of the wobble plate, are open whilst the others are closed. While the valves are closed they are venting the contents of the other pistons to the outside. The rotary motion of the shaft pushes the pistons back to the top and forces the water out. The valves that are experiencing pressure are swapped from valve to valve to cause the linear piston to rotary shaft motion.

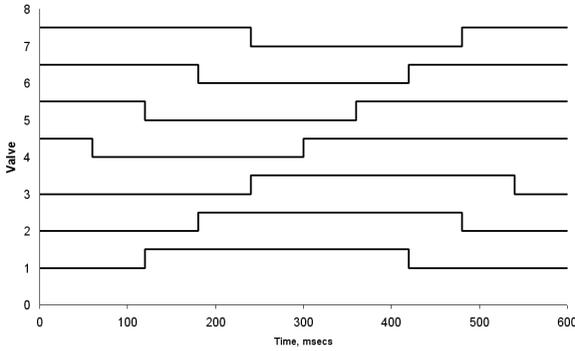


Fig. 4. Valve sequence for 100% motor torque.

The use of sequenced valves allows the operation of the motor in a number of unique ways that allow it to behave like a stepper motor. At present, the motor controller runs open loop with no feedback from the shaft angle. The additional features of the motor include the following.

Full Reversibility. By reversing the sequence of operation of the valves the motor can be reversed easily. This can be useful if the head jams.

Full Speed control. Changing the frequency of switching changes the operational speed of the motor. This can reduce stress on the head if hard material is encountered.

Stopping. Closing all valves stops the motor independently of the pump.

Braking. Activating all valves forces the motor into a braking mode.

Full torque at zero speed. All valves can be open for the period with speed controlled by the frequency of operation.

Full torque control. By activating only a single or multiple valves for various periods controls the torque of the motor.

3.2 Motor Torque Theory

An equation can be derived for the instantaneous torque at any position from the internal geometry of the motor, i.e.,

$$\tau = \sum_{i=1}^n (P_i * r * A * \sin \theta * \sin(\alpha - \zeta)), \tag{1}$$

where i is the number of the piston, P_i is the pressure of piston i , A is the cross sectional area of the piston, r is the distance from the piston to the centreline, θ is the angle of the wobble plate, α is the angular position of the first piston, and, ζ is the relative angle between the first piston and piston i . This equation is plotted in Figure 5 for a number of different sequencing algorithms. The plotted torque is a result of a summation of all the pistons acting on the shaft.

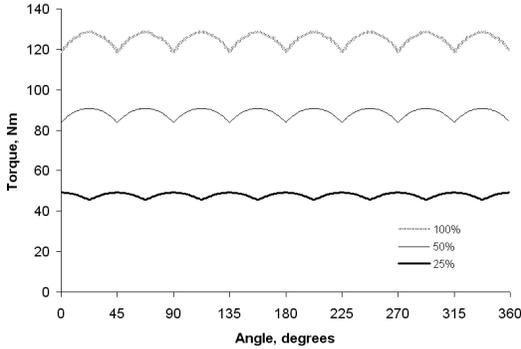


Fig. 5. Theoretical torque levels.

The maximum torque values and smoothest output results when the pistons are activated symmetrically around the midpoint of the downward stroke.

3.3 Cutting Head Performance

A prediction of motor torque, power and the penetration rate of the cutting head is required to determine the performance of the system based on the performance of the motor. Due to the complexities of the cutting head system and the lack of prior knowledge of type of rocks to be targeted, an empirical based formulation is employed [5]. Firstly, the energy required to remove a cubic centimetre of rock is given by

$$E_s = 0.9518 * S_c + 104.38, \quad (2)$$

where E_s is the Specific Energy per volume of rock (J/cm^3) and S_c is the compressive strength of the rock in MPa. For rotary drilling the torque, T (Nm) required to be supplied from the bit is given by

$$T = a * WOB * D/100, \quad (3)$$

where a is a dimensionless ratio for rotary drilling with 0.1 typical for tri-cone bits and 0.15 for polycrystalline composite (PDC) bits. WOB is the weight-on-bit or thrust (N) and D is the diameter of the bit in cm. The rate of penetration ROP (m/hr) of the bit is given by

$$ROP = 0.048 * a * WOB * RPM / (D * E_b), \quad (4)$$

where E_b is the work done by the bit ($= \text{Efficiency} * E_s$), and RPM is the speed in revolutions per minute.

To estimate performance a rock with a compressive strength of 220 MPa is assumed together with a cutter bit efficiency of 50% and a thrust force equal to the load capacity of the bit, namely 4500N. For a bit diameter of 104.8mm

and a speed of 400 rpm a tri-cone bit requires 47.14 Nm of torque and a PDC bit 70.72 Nm. The resultant rates of penetration are 1.31 and 1.97 m/hr for the tri-cone and PDC cutters, respectively.

3.4 Motor Experiments

The motor was set up with a no load condition on the test bench with a rotary potentiometer attached to the output shaft to measure rotation angle as a function of time. The results of these preliminary trials are shown in Figures 6 and 7.

Figure 6 shows the output from the motor at a number of different sequencing speeds with a setting of 100% torque. 100% torque is defined as the case where four valves are engaged simultaneously. In the figure, the top trace shows the output at 10 rpm. Here it can be seen how the motor quickly rotates to each position and then waits until the next valve change. In this case the behaviour is analogous to a stepper motor. As is the case for stepper motor applications, the motor behaviour will depend on the inertia of the total system. The second trace, at 50rpm, also shows the stepper like behaviour, although not as clearly as the 10rpm case. In the third trace, at 100rpm, the speed that the sequencing speed has started to catch up with the rotation of the shaft and the angular rotation is slightly smoother.

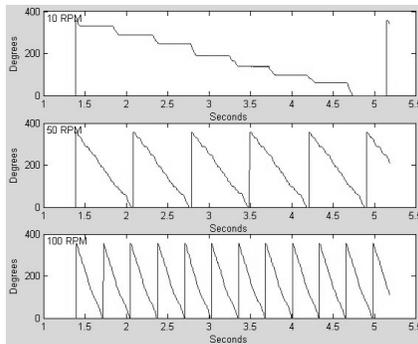


Fig. 6. Experimentally measured shaft angle at 10, 50 and 100rpm for 100% torque.

In Figure 7, a comparison is made of the operation of the unloaded motor at a number of different torque levels. The torque settings are defined as follows: 100% refers to four valves actuated, 75% three valves actuated, 50% two valves actuated simultaneously and 25% means one valve at a time. In these experiments, the motor sequencing speed is set to 100rpm. In the figure, the 100% torque case, represented by the solid line, shows the shaft accelerate to the next position and wait for the next sequence to switch. This is also evident for the 75% torque test, to a lesser extent, but is not as evident

for the lowest torque cases. Again, this behaviour will change with loading condition.

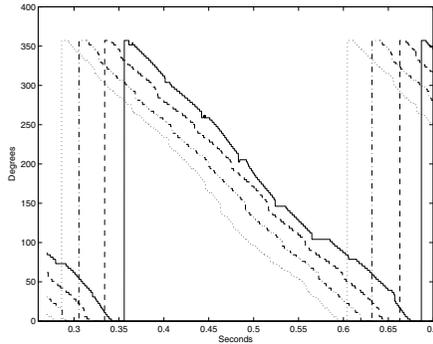


Fig. 7. Experimentally measured shaft angle at 25% (dotted), 50% (dotdash), 75% (dashed) and 100%(solid) torque settings for 100rpm shaft speed.

4 Electronics and Control

While PID control of water hydraulic systems has been reported [6], proportional control valves do not exist at the system pressure (34MPa) or in a compact enough package for use in the robotic drill. The drilling system software consists of embedded software and a graphical user interface software. The embedded software is responsible for tactical control, namely real-time control of the valve output sequences, and, the graphical user interface software is responsible for strategic control, such as start, stop, forward, backward and speed/torque control.

4.1 Steering Algorithm

The control of the drilling system has some similarities with the control of stepper motors. That is, the controller sends a sequence of digital signals which dictate the state of the motor and body valves. The axial ram controller uses position feedback, via linear potentiometers, to provide closed loop control. Due to unavailability of proportional control valves for low viscosity fluids, a rapid digital control is used. This is satisfactory due to the relatively slow motion of the steering due to its progress being limited by the rate of cutting of the rock. The feedback data is required to detect rams that may be leading the others due to irregularities in the rock which could bias the direction of the cutting head.

The control algorithm for straight-ahead-steering involves detecting the ram having the largest displacement and checking if this displacement exceeds

the average of the displacements by a prescribed threshold. If it does, than this ram's pressure is removed until its displacement is less than the average plus the threshold. To steer the drill, the desired pitch and yaw is used to provide a tilt of the cutting head. As the drill progresses along the cutting path, the selected tilt causes the drill to veer off in that direction. To implement this control, the straight-ahead-steering algorithm is used except that an offset is subtracted from each rams zero position which effectively misaligns the cutting head to the body. Given the tilt angles are less than 3 degrees the following offsets c_1 , c_2 and c_3 are calculated as

$$c_1 = r * \tan(\text{pitch}), \quad (5)$$

$$c_2 = (-r/2 * \tan(\text{pitch}) + \sqrt{2/3} * r * \tan(\text{yaw})), \quad (6)$$

$$c_3 = (-r/2 * \tan(\text{pitch}) - \sqrt{3/3} * r * \tan(\text{yaw})), \quad (7)$$

in which r is the distance from the drill axis to the ram axis. The pitch angle is positive down and yaw is positive to the left when looking down-hole. The offset c_1 corresponds to ram 1, which is directly above the drill axis. Ram 2 is 120° in the clockwise direction, when looking down hole, and ram 3 is 120° in the anti-clockwise direction.

4.2 Navigation

The section between the motor and the clamping section contains the embedded electronic systems to control the operation of the motor, rams and clamps. It also contains an inertial navigation system (INS) which uses 3 rate gyroscopes, 3 accelerometers and 3 magnetometers [7]. Data from the INS and from the linear displacement sensors (which measure the movement of the axial rams) is used to control the trajectory of the drill. All ram control, valve sequencing and steering is embedded within the drill, whilst supervisory control, position information and condition monitoring information is to be relayed to the surface. The surface computer is responsible for supervisory control, data logging and control of the surface infrastructure.

4.3 Embedded Software

The valves used for the body and motor were binary microhydraulic valves having a positive metal ball on seat seals. The valves had two states - they are either on or off, therefore, digital signal levels, i.e., either a '1' or a '0', are used to control the state of the valves. Consequently, the embedded software controller possesses body valve and motor valve state machines. The controller was developed using an ImageCraft ICC c compiler and implemented on an Atmel ATmega128 microcontroller. Briefly, a command parser serves to detect input commands and take appropriate action. The body or motor states are advanced whenever body state or motor state time out occurs.

4.4 Hardware Description

A stack of three printed circuit boards (PCBs) situated inside the u-shaped cavity within the body of the drilling system. One PCB accommodates the Atmel ATmega128 microcontroller while the other two each possess 8 solenoid driver circuits (8 for the body valves and 8 for the motor valves). The solenoid driver circuit consists of a microprocessor output port driving a MTD20N06HDL FET via a MOCD223-M opto-isolator. A 12CWQ04FN dual diode is used to diminish the back-EMF from the solenoid valve and preserve the FET.

5 Adaptive Cutting Mechanisms

The drill is designed to be flexible in its operation to allow it to adapt to a number of different rock types. This can allow the drill to progress through multiple rock types during a single drilling operation or, alternatively, to allow the drill to be used with various cutting heads for a number of different drilling tasks without the need to purchase a different drill for each task. This change in operation is achieved with common hardware but with changes to the software.

Three modes of cutting are envisaged with the robotic drill; rotary cutting with point attack or drag picks, percussive mode, and wobble mode.

The rotary cutting mode uses an innovative cutting head that contains cutting tips made from thermally stable diamond composite material. The cutting tips are relatively sharp compared to conventional bits and have a very high wear life. Experiments with these tips have shown that the cutting forces and energy consumption are significantly lower than conventional bits. In particular, the amount of thrust force was significantly reduced which is consistent with the use of friction to provide this thrust. Because of the electronic control of the motor, a control input can result in a rapid response from the motor. This control input is likely to be either that from the integrated torque measurement system or the linear displacement measurement system. If the rock type that the drilling system encounters suddenly changes, then the motor torque and rotation speed can quickly be modified to prevent breakage of the cutting head, shaft or motor components. This change to the speed torque characteristics also allows the drill to adapt to a more optimum regime to increase the rate of progress through the rock.

In the percussive cutting mode the three axial rams are used to reciprocate the cutting head back and forth to allow the cutting head to impact on the face of the rock. The rock is crushed under the cutting tips and flushed away by the cutting fluid. In this mode, the motor is controlled to allow the head to be indexed to the next position for the next impact. The drill control system allows complete control over the frequency and intensity of the

hammer impacts to allow optimisation of the cutting mode without changes to hardware.

The third regime of cutting is known as wobble mode. By sequencing the operation of the three axial rams, an off-centre load can be placed on the cutting head. This will increase the stress on individual bits allowing higher stresses to be placed upon the rock without any increase in the axial or torque loads to the drill. It is expected that the stress increase on the rock could be of the order of 10 times the normal cutting modes and is useful for the drilling of extremely hard rock.

6 Conclusions

The design and control of high power water hydraulic robotic systems presents unique challenges. This paper describes some of those challenges and the techniques used to overcome them to develop a unique system for an autonomous robotic drilling machine for underground mining.

References

1. M. Siuko, M. Pitkaaho, A. Raneda, J. Poutanen, J. Tammisto, J. Palmer, M. Vilenius (2003) Water hydraulic actuators for ITER maintenance devices, *Journal of Fusion Engineering and Design*, 69 pp141-145
2. G.H.Lim, P.S.K. Chua, Y.B.He (2003), Modern Water Hydraulics - the new energy transmission technology in fluid power, *Journal of Applied Energy*, 76, pp239-246
3. H. Tsukagoshi, S. Nozaki, A. Kitagawa 2000, Versatile Water Hydraulic Motor Driven by Tap Water, *Proceedings of the 2000 IEEE/RSI International Conference on Intelligent Robots and Systems*
4. <http://nessie.danfoss.com/Products/Motors.asp>, accessed 10 Jun2 2005.
5. Tim Harrison (2000) Very Deep Borehole: Deutags's opinion on boring, canister emplacement and retrievability, SKB Rapport R-00-35, ISSN 1402-3091
6. Kazuhisa Ito, Shigeru Ikeo (2002) PID Control Performance of a Water Hydraulic Servomotor System, *SICE 2002 Aug 5-7 Osaka*
7. Gregg Buskey, Jonathan Roberts, Peter Corke and Peter Ridley (2002) Sensing and Control for a Small-Size Helicopter, *International Symposium on Experimental Robotics (ISER), Sant' Angelo d'Ischia, Italy*

A Wearable GUI for Field Robots

Andreas Hedström, Henrik I. Christensen, and Carl Lundberg

Centre for Autonomous Systems (CAS),
Numerical Analysis and Computer Science (NADA),
Royal Institute of Technology (KTH),
S-10044 Stockholm, Sweden,
{ah,hic}@kth.se, Carl.Lundberg@fhs.mil.se,

Summary. In most search and rescue or reconnaissance missions involving field robots the requirements of the operator being mobile and alert to sudden changes in the near environment, are just as important as the ability to control the robot proficiently. This implies that the GUI platform should be light-weight and portable, and that the GUI itself is carefully designed for the task at hand. In this paper different platform solutions and design of a user-friendly GUI for a packbot will be discussed. Our current wearable system will be presented along with some results from initial field tests in urban search and rescue facilities.

Keywords: GUI, Field Robot, Wearable

1 Introduction

Deployment of robot systems for outdoor field applications poses a number of interesting challenges. The outdoor application of computer systems is in itself a challenge due to lack of screen contrast, limited availability of weather proof casings, etc. At the same time the mobile systems are often operated by inexperienced users, which calls for careful design of the interfaces. This involves both selection of a suitable hardware platform, and design of an associated interaction system with GUI etc.

In joint research between the Centre for Autonomous Systems, KTH and the Swedish Defense Materials Administration (FMV) a study is considering how rugged platforms such as a PackBot can be utilised for search and detect type missions by inexperienced users. As part of international peace keeping missions or as part of search for people after major disasters there is an interest to deploy PackBot systems (See Fig. 1). These systems are to be used for early intelligence gathering before people enter a building or a neighborhood.

A PackBot is by default delivered with an operator control unit (OCU) that is composed of a rugged laptop computer with an attached joystick for

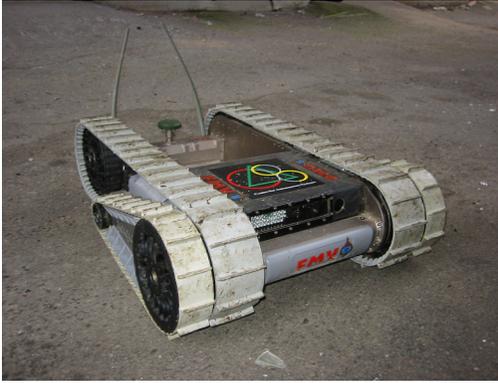


Fig. 1. The PackBot system used as part of this study

tele-operation. The system has an operation time of about 1 hour between recharges. The user interface is designed around a GUI that is divided into four sub-windows. The windows can visualize information such as:

- Video from fish-eye / IR camera (streamed video)
- Present speed
- Battery capacity/remaining power
- Temperature of key motors
- Track of GPS position
- Payload information
- Status of WLAN Connection

In most cases the information is presented as a small scalable icon and a text snippet that specifies the type of information visualized. The system poses a number of challenges

- It is heavy and not easy to carry
- The system is not packaged for rugged outdoor operation with weak connectors etc.
- The screen has limited contrast outdoor due to use of a regular LCD panel
- The system has a limited time of operation

Consequently there is a need to consider how a suitable interface could be designed to enable realistic use of the system, and how it could be designed to enable easier use by inexperienced users.

The prototypical user for a system is a young male with a high school degree. The users often have experience using a cell phone and some experience with computer games, but they cannot be considered computer experts.

Some of the requirements for a system include:

- High contrast of video screen during outdoor operation
- Easy control interface with intuitive control of the vehicle

- An uncluttered visual interface with access to key information
- A manageable learning curve that enable use after no or limited training
- Highly portable for easy transportation and suitable battery duration.

In the present paper a design of a wearable control unit for a field robot is described. The unit has been used for tele-operation of a PackBot in urban search-and-rescue missions involving operation both in factory buildings and in area coverage.

Design of user interfaces for this type of robots has been described in the literature earlier. Fong et al. [1] describe use of a PDA for basic teleoperation using an on-board keypad and video feedback to the user. The interface also includes text messages from the supervisory control. Perzanowski et. al. [2] at NRL have also designed a speech / PDA interface for control of iRobot ATRV robots. The emphasis is here on integration of speech with simple commands for teleoperation in the context of a semi-autonomous system. Leskinpala et al [3] describe the use of a PDA based system for simple tele-operation of a mobile platform, but the tests have only been performed indoor in relatively accommodating environments. Edwards [4] outline the design of a wearable computer system that has been deployed in Afganistan for teleoperation of the iRobot PackBot and the Forster-Miller system. Few details are available on the actual interface design. Mayol et al [5] have also described a wearable systems for localisation and mapping, in which small visual platforms are remote controlled for mapping. Few details are available on the interface design. The most relevant work here seems to be the work by Fong, and Perzanowski, though none of them have designed a truly wearable system for robot operation. One would have expected a large number of wearable systems for robot tele-operation, but our review of the literature have not shown much reported research.

Initially the various options for design of a portable user interface are discussed (Section 2). The details of the actual design are then presented in Section 3. A number of observations on the use of the system is provided in Section 4. Finally a summary and outlook for future work is presented in Section 5.

2 Selecting a Platform for Control of a Field Robot

As mentioned above a number of people have already reported use of a PDA for control of field robots in everyday use. In earlier work we have also reported use of a PDA for control of an iRobot ATRV [6]. The PDA has the advantage that it is a well known platform, it is lightweight and has good battery performance. Almost all of PDAs have poor outdoor performance in terms of screen contrast. In addition most newer PDA have a problem for control of platform motion. The built-in joystick does not offer adequate control of the platform, it is too difficult to determine if a side-way or forward/backward motion is requested.

An alternative to PDA control is the use of a tablet PC as for example provided by HP/Compaq. These platforms have a large screen and but the same contrast issues as a PDA. The platforms are not rugged and in addition there is a need to use a special purpose pen for the control, which makes it difficult to use and at the same time the platform is relatively large and difficult to carry.

Recently there has been significant progress on small rugged computers that can be carried in a large pocket. These platforms have standard interfaces for networking, screen, USB, etc. The platforms are available with standard processors and are relative power efficient, as reported in [4]. It is thus possible to design a standard wearable computer solution in which a computer is fitted with an 802.11b network and a pair of batteries for easy use. Using the standard VGA connector it is possible to attach either a pair of computer glasses or a small touch screen for mounting on the lower arm. The touch screen has a standard mouse interface. The glasses, with an integrated screen, has the advantage that the contrast problem is more manageable as the backlighting from the background enhances contrast, which is the reverse effect of normal LCD screens. For the control of a platform it is possible to use a wireless gamepad, which is a well-known modality for computer games. It typically offers two analog joysticks and a number of push buttons for selection of different mode variables. The gamepad design is similar to what is frequently used for tele-operation of cranes and mining equipment.

For our studies we have chosen a 650 MHz ULV Celeron computer with standard ports for PS2, USB, VGA, LAN and serial COM. To ensure rugged operation in an outdoor setting it was decided to use the built-in Compact Flash interface to install a small linux system on a standard CF card. These cards can typically tolerate shocks in excess of 100G which makes it a suitable solution for the present application. For communication a USB - WLAN interface is used and it is mounted on a shoulder pad for maximum operation range. For feedback to the user, two solutions were tested. A 7" LCD touch screen was tested (see Fig. 2), but the contrast is similar to that achieved with a PDA. It is an acceptable solution for indoor operation but not a good solution outdoors. An alternative is a head-up display (HUD). We have chosen the SV-6 "eye-monitor" system from Micro Optics. It offers a resolution of $640 \times 480 \times 18$ which provides adequate resolution when mounted in the accompanying glasses.

The overall hardware solution for a wearable system for PackBot control is shown in Fig. 3.

3 System Design

The PackBot comes with an on-board computer. The computer is used for local control of the PackBot in terms of speed, direction, and the flipper position. The control is purely in terms of speed. In addition the on-board



Fig. 2. Example LCD screen tested for lower arm mounting. The contrast is, however, a major challenge for outdoor usage. The screen is shown together with a corresponding iPAQ interface that can be used for portable control, but again with contrast issues

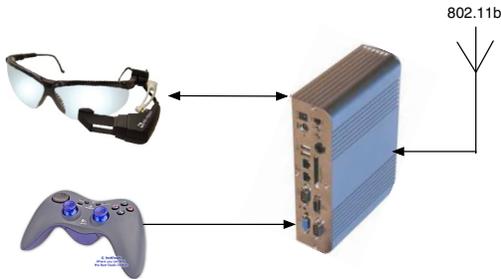


Fig. 3. The hardware setup for the control system

computer can stream video to the control unit. The PackBot has two on-board cameras. One is in the visual spectrum and has a fish-eye lens fitted. The other is an infra-red camera with a narrow field of view lens. The system also has a built-in IR illuminator that can be used for operation in complete darkness. The IR illuminator has binary control (on/off). All control and video is communicated using two 802.11b links. The video is multicasted for use by multiple clients. The computer system is not accessible to users and consequently it is used as a blackbox. All extra control functionality has to be achieved on the operator control unit.

To drive the system the wearable computer is setup with a small scale linux distribution that fits onto a CF card. Only the bare minimum is available. The DSL - Damn Small Linux distribution was used as a basis for our system. DSL is a 50Mb distribution designed to fit an X11 system on a small credit card sized cd. DSL was then modified to enable booting off a CF card.

The design of the software system is relatively simple. The overall system design is shown in Fig. 4

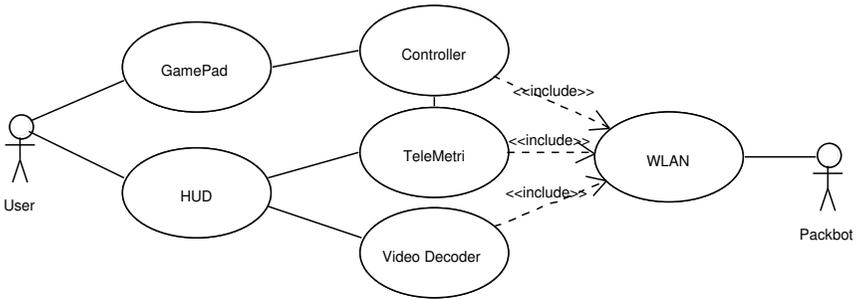


Fig. 4. The Software Components of the Wearable System

The controller handles the take-up of control of the PackBot, i.e. observer mode versus control mode. In control mode an active link to the system is maintained. The system is also responsible for handling telemetry feedback from the system. In the present setup the control is purely tele-operated, but other research is considering the addition of autonomy to the system. At present the telemetry that is feedback to the user / control process are velocity information, GPS coordinates (when available), battery status, temperature information, and orientation of the platform with respect to horizontal.

The user control of the platform is through the earlier mentioned gamepad. The gamepad is of the wireless variety, which provides a maximum of flexibility and at the same time it is for many people a natural interaction modality. The main interaction is through the two on-board joysticks. The left analog joystick is used for control of speed and direction of travel, which is the default behaviour of almost every video game. The right joystick controls the two on-board flippers. The flippers can only be moved in synchrony and thus only the forward/backward motion of the joystick is utilized. The gamepad has 11 on-board buttons. One is used for request of control of the platform and another is used to call up a help screen. The remaining buttons are mapped to frequently used actions such as toggle brake on/off, selection of video source (fisheye/IR camera), increase/decrease speed limit and finally also a GUI reset button.

The head-up display at all times streams the video data from the PackBot. The display is setup to have three modes of information: i) the bare video stream, ii) video plus velocity information, and iii) video, speed and telemetry information. The different types of video information is shown in Fig. 5 and Fig. 6.

The basic setup is so that one can toggle between the three types of information. For the beginner a minimum of information is typically the preferred



Fig. 5. The screen shown with video only information and with velocity information



Fig. 6. The screen shown with video, speed information and telemetry data

mode of operation. As the operator gains more experience the details can be added gradually.

The system has been implemented using the Simple DirectMedia Layer (SDL) which is a cross-platform library. SDL has been designed to provide low-level access to audio, keyboard, joystick, video and framebuffer. It is supported on a large number of platforms including Linux and as such provides for easy portability across PDA, Wearable and Tablet systems. The library is written in C with MMX optimized code, which provides for efficient execution and a minimum load on the CPU, which is important for battery operated systems.

The entire system is capable of streaming video to the user at 15Hz with a video resolution of 320×240 in real-time with a CPU load of 75 %. Actually it can show video of up to 25Hz (using more CPU) but this is only valid in the near proximity of the robot where the radio quality is good. As the radio quality drops, so does the frame-rate. For realistic scenarios, the frame-rate

varies between 5-15Hz which is enough to solve most tasks. It would be of interest to be able to stabilize the image, as there is significant shaking in the image during traversal of staircases etc., but this has not yet been achieved.

4 Field Use of the System

The wearable system has been tested in a number of pilot trials. The system mounted on a person is shown in Fig. 7



Fig. 7. The Wearable System in Operation

The system has been evaluated both for indoor operation in a factory setting, where the person is required to drive the PackBot into a building and up a staircase to search for a particular object, and in an urban setting where the system is deployed to enter an area with a number of buildings to determine if there are any suspicious objects. Here the system is required to drive up to a car and inspect the underside of the car for any unexpected objects.

For the operation indoor the main challenge is driving in a staircase. The shaky motion of the platform while traversing the staircase is a challenge for the operator.

In the outdoor setting the operation is natural and easily picked up by the operator.

For both cases it must be recognized that the goggles only feed video data to one eye of the operator. This poses a small challenge and in the beginning the disparity in information between the two eyes (the one with the screen and the eye with no feedback information) is a bit confusing. However, when entering an area there is an interest to be able to see the immediate surrounding with a minimum of obstruction of the field of view and at the same time shift to robot operation without any changes in the setup. This can easily be achieved with the present setup. After some training the users find that it is easy to drive the system and achieve efficient operation. The system has a battery time of about 3 hours, which is adequate for most interventions.

In comparison to the standard iRobot OCU a high degree of mobility is available, and the same information as is available through the OCU can be provided. The direct integration into the uniform is considered a major plus. At the same time the gamepad provides at least the same level of maneuverability as the OCU attached joystick. Initial comparative tests with the two systems demonstrate that a similar performance in terms of flexibility and speed is achieved for the wearable system and the OCU.

A wearable solution has been reported for the iRobot system in the paper by Edwards [4], unfortunately no information is available about the display of information, or the modes of operation. In addition it has been impossible to get access to such a system for comparative experiments.

5 Summary

In the present paper we have discussed the design of a wearable computer system for tele-operation of a PackBot system. The system was designed to address many of the challenges that exist with using a normal laptop for field applications, in particular in an outdoor setting. Through use of wearable goggles in combination with a gamepad it is possible to design an efficient interface that can be used by people after only a brief introduction. The design has been implemented on a linux based computer system that is worn in a small belt together with the required batteries. The system has been implemented and tested in pilot trials indoor and in urban settings with highly satisfactory results. As more degrees of autonomy are added to the system the overall design of the user interface will have to be adopted to accommodate such functions. However the present design is efficient, portable and easy to use. It offers an ideal work environment for control of a small field robot such as the PackBot.

Future work will include more thorough field evaluation of the system and indepth comparison to the standard OCU unit delivered with the PackBot. In addition various degrees of autonomy and post-processing of information such as image stabilisation is clearly of interest.

Acknowledgements

This research has been sponsored by the Swedish Defense Materials Administration (FMV). The support is gratefully acknowledged.

References

1. T. Fong, C. Thorpe, and C. Baur, "Advanced interfaces for vehicle teleoperation: Collaborative control, sensor fusion displays and remote driving tools," *Autonomous Robots*, vol. 11, pp. 77–86, July 2001.
2. D. Perzanowski, A. C. Schultz, W. Adams, E. March, and M. Bugajska, "Building a multimodal human-robot interface," *IEEE Intelligent Systems*, vol. 8, pp. 16–21, Jan/Feb 2001.
3. H. K. Keskinpala, J. A. Adams, and K. Kawamura, "Pda-based human-robotic interface," in *IEEE Conf on SMC*, (Washington, DC), April 2003.
4. D. Edwards, "Wearable computer musters for battlefield robot control," *COTS Journal*, pp. 20–22, April 2003.
5. W. Mayol, B. Tordoff, and D. Murray, "Wearable visual robots," in *Proc Int Symposium on Wearable Computing, 2000*, 2000.
6. H. Christensen, J. Folkesson, A. Hedström, and C. Lundberg, "Ugv technology for urban intervention," in *SPIE – Home Security*, (Orlando, FL), May 2004.

Design and Implementation of Machine Control Systems with Modern Software Development Tools

Matti Öhman¹ and Arto Visala²

¹ TKK Automation Technology Laboratory Otaniementie 17, 02150 Espoo
Finland matti.ohman@hut.fi

² TKK Automation Technology Laboratory Otaniementie 17, 02150 Espoo
Finland arto.visala@hut.fi

1 Introduction

The *Agrix project* studies machine control systems for agricultural implements. The research goal is to design and implement a prototype of an *open*, *generic* and *configurable* implement control system. Other research topics are user interface design, positioning and navigation, telematics and fault diagnostics, wireless communications and precision farming. In the project consortium, there are four research institutions and eight companies. Most of the funding comes from TEKES³. The project consists of three phases. In the first phase, the *Agrix fast prototype* was developed and tested in the summer 2003.



Fig. 1. Pneumatic drill, sprayer and no-tillage drill

The development of the *Agrix basic system* started in the autumn 2003 and was completed with field tests in summer 2004. Two combined seed and fertilizer drills and one sprayer were modified with new control units and additional sensors. The machines are shown in Figure 1. A commercial tractor and virtual terminal together with a home-grown task controller and GPS

³ National Technology Agency of Finland

adapter were used to complete the *ISO 11783* system. The control system software is developed with *RTI's Constellation* software development tool. In the third phase, the results from methods research will be integrated to the basic system thus completing the *Agrix advanced system* which will be tested in summer 2005.

Open control systems use standardized interfaces and communication protocols so that control units from different vendors can be used in the same system. The Agrix control system is based on ISO 11783, which is an emerging communication standard for agricultural vehicles with widespread support from the industry⁴. *Generic control systems* can be used to control several different implements by changing only the control software. The hardware, the operating system and the tool chain can be reused without modifications. This greatly reduces the design cost of a new system. *Configurable control systems* can be easily configured using a high-level, usually graphical tool, instead of writing it with a low-level programming language. Such high-level tools are widely used in industrial automation. Most embedded systems are still programmed with assembler or C languages because of the platform limitations. However, the computational and memory limitations are becoming less important with every generation of micro-controllers.

The control functions of traditional agricultural machines have been quite simple and low-level languages have been adequate for programming these systems. But the machines are getting bigger and more complex. Large machines require more automation to keep the operator strain at an acceptable level. Emerging production methods, such as precision farming, require positioning and feedback control. Automation technology can also be used to produce more accurate farm records. To reach their full potential, separate control systems need to be connected. Creating distributed real-time embedded control systems with low-level languages is slow, error-prone and prohibitively expensive, especially if the production series are small.

2 ISO 11783

In the late 1980's the development of communication networks between tractors and implements was started in Germany. The main parts of *DIN 9684* were completed by the end of 1991. At the same time similar efforts were made in the United States, targeting the *SAE J1939* standard. Both standards are based on CAN, but they are incompatible[8]. The development of the ISO 11783 standard was started in 1992 and it is still under development. ISO 11783 specifies the data network for control and communications on agricultural vehicles and it uses some parts of DIN 9684 and SAE J1939. The physical and data link layers are based on *CAN 2.0b* specification with extended identifiers[1].

⁴ ISO 11783 is sometimes referred as ISOBUS, which is actually the name of the implementation specification of the standard.

ISO 11783 defines two buses. Implement control units, auxiliary input devices, task controller, GPS receiver and virtual terminal are connected to the implement bus. Tractor's engine, transmission and hitch control units are connected to the tractor bus. The tractor bus is accessible from the implement bus only via the *tractor electronic control unit* (TECU) which acts as a bridge between the tractor bus and the implement bus. ISO 11783 specifies three tractor classes depending on the available features. *Class 1* tractor provides only basic measurements, *class 2* tractor has more advanced measurements and *class 3* tractor allows the implement to control some of its functions.

A *virtual terminal* (VT) is the generic HMI device defined in ISO 11783. It has a graphic display, soft keys and some means to enter data. Any control unit can transmit its user interface to the terminal. An *auxiliary input device* is an additional HMI device. It can be used to supplement the virtual terminal when a more specialized interface is required. With the help of the virtual terminal the operator can configure the auxiliary inputs to the desired functions.

One or several *implement electronic control units* (IECUs) are needed to control a single implement. Once the terminal has been initialized, the controller can update its user interface and receive operator feedback. Only one user interface can be active at a time but control units can update their user interfaces even if they are not active. When a user interface becomes active again, it will reflect the new changes. For the control units, this creates an illusion of an exclusive access to the virtual terminal.

A *task controller* is an optional part of an ISO 11783 system. The planned tasks are loaded into the task controller. It executes the tasks by sending *process data messages* to other electronic control units. The task controller can also collect data during field operations which can be used to produce accurate farm records. Planned tasks may range from field level operations to precision farming operations⁵. A *GPS receiver* provides position information to the ISO 11783 network. The message format is specified in the *NMEA 2000* standard[6] which is compatible with ISO 11783.

3 Methodologies and Tools

In software engineering, great efforts have been made to improve software quality and programmer productivity. New methodologies and tools have been created for designing increasingly complex systems. The new *IEC 61499* standard is intended for "distributed industrial-process measurement and control systems"[3]. The Object Management Group's *UML* standard dominates in most other application areas. UML promotes the best software engineering practices and provides a descriptive graphical language⁶. The use of UML in

⁵ In precision farming soil spatial and temporal variability is identified, analyzed and managed for optimal results.

⁶ Even if the graphical languages are not *silver bullets*[2], they can be very useful in specialized applications like control system design.

systems engineering is increasing[4]. UML approach was selected for the Agrix project simply because there are many commercial UML tools⁷ available but no mature tools that support IEC 61499. We selected RTI's Constellation software development system, as it supports UML and it is especially designed for building control systems.

4 Constellation Software Development System

RTI's Constellation provides a framework for building control systems and other periodically executable software components. Components for processing continuous signals are executed at the specified frequency. The control systems can also be reconfigured on the fly by activating and deactivating components. This kind of mode change is needed e.g. to transition from manual to automatic control. In addition, Constellation provides a framework for processing discrete events. UML-style state machines can be used for event processing. State transitions can be triggered by events or changes in continuous signals. Because time can be handled as a continuous signal, implementing time-dependent state transitions is very easy.

Constellation encourages component reuse. The components are loosely coupled by well defined interfaces. Because components are connected using interfaces, it is possible to compose fundamentally different control systems from the same set of components. Creating new interfaces and adding them to new components is easy. New components can be created by combining existing components, drawing state machines or by writing primitive components with *C++*. Because Constellation is based on *C++* programming language, it creates relatively efficient and fast code. Constellation runs in *Windows* and *Linux* operating systems. It can also create executables for *VxWorks* operating system and supports various processor architectures.

4.1 Component Types

Atomic Components (ATCs) provides one or several functions which other components can call. These functions can call other functions from other components, or read and write signal values. Atomic components are executed only when their functions are being called.

Data Flow Components (DFCs) have *OnExecute* and *OnStateUpdate* functions which are executed periodically by the run-time system. The run-time system sorts DFCs to satisfy their input–output dependencies and places them on an execution list[7]. The DFCs on the list are executed at each sample period. It is the programmer's responsibility to cut algebraic loops. Data-flow components can also be created directly from MATLAB/Simulink models.

⁷ E.g. I-Logix's Rhapsody, IBM's Rational Rose and Insoft's Prosa

Finite State Machine Components (FSMs) are defined graphically by drawing the states and the transitions between them. The state charts use standard UML notation and semantics with a few minor exceptions. The state transitions can be triggered by events or changes in signal values, which are evaluated periodically. *State Transition Components* (STCs) are executed as a response to a state transition in FSMs. Multiple STCs can be associated to a single state transition or used as entry or exit actions for states, levels and composite states.

Composite Object Group Components (COGs) are used to group other components and connectors. There are three kinds of connectors. Pins correspond to variables, bubbles correspond to functions and interfaces can contain pins, bubbles and other interfaces. In addition, Constellation provides library components for asynchronous message passing.

4.2 Configurability

Frameworks can provide different configuration strategies. The simplest one is *instantiation*, which means using existing components without any change. Generic components can be specialized by adding new functionality⁸. This kind of configurability is called *specialization*. An easy way to configure components is to specify parameters for parameterized components, which is called *parameterization*. However, perhaps the most powerful form of configurability comes from *object composition*, which means connecting components to create new functionality and new components.

Many parts of the Constellation application framework are used without change in all applications. For example, the run-time parser and execution engines for sampled-data and event-based systems are instantiated at program startup. There are also other run-time services that can be used when needed. Functionality can be added to Constellation applications by specialization. For example, Constellation creates specialized wrappers for Simulink code by inheriting a generalized Simulink wrapper. However, Constellation favors composition over inheritance. *Generalization* is the opposite process of factoring out the common features from a set of components. Constellation applications can be configured by parameterizing the components. For example, application's habitat⁹ component has parameters for adjusting the execution rates and priorities of sampled-data and event-processing threads. Object composition is the dominant technique of configuring Constellation applications. Object composition is defined dynamically at run-time through objects acquiring references to other objects. Composition requires objects to respect each other's interfaces[5]. In Constellation, the system's behavior depends on the relationships of the components rather than being defined in one component.

⁸ In object-oriented programming languages, subclasses can be inherited from super-classes.

⁹ Habitats are execution environments for other components.

The Constellation application framework is designed for building control systems. It emphasizes *design reuse* over *code reuse* although it has ready-to-use component libraries. Design reuse leads to an inversion of control between application and the framework[5]. When using a conventional function library, the programmer writes the main program that calls the functions he wants to reuse. But when using an application framework, the programmer writes the functions that are called by the framework which is reused. As a result, applications are faster to build, have similar structures and are easier to maintain.

5 Design

The design of the control system was done in a top-down manner. The high-level components and their interfaces were defined first and then attention was shifted to lower-level components and interfaces. Because no implementation was done at this time, interfaces could be easily changed as new deficiencies in the design were discovered.

5.1 Machine Modes

From the start it was obvious that the implements will have different operating modes and that there are some modes that are common to all implements. To keep things simple — for the designers as well as the operators — the number of modes was reduced to only three. The *free mode* turns off all automation. The operator can apply individual controls freely by pressing the soft keys. This is useful during maintenance when it is necessary to drive the machine to a specific position. The free mode is also used as a fall-back mode when something goes wrong in the other modes¹⁰. When entering the free mode, all ongoing controls are stopped for safety reasons. The free mode is also used for calibration functions so no dedicated calibration mode is required.

The *transport mode* is used for transporting the implement, as the name suggests. The implement is driven to its transport position and then the controls are locked. This allows the safe transportation of the machine. The *field mode* is the normal operation mode during the field operations. Automation is used as much as possible to minimize operator strain. The level of automation can be adjusted by selecting manual or automatic modes for individual functions e.g. marker mode or tramline mode¹¹. The field mode has two sub-modes. In the *work mode* the machine is “doing its thing” on the field and in *headlands mode* the machine can be quickly turned around. When changing between these two modes the control unit automatically controls the different functions in the proper sequence.

¹⁰ For example a sensor failure can render the field mode inoperable, in which case the manual mode can be used drive the machine to its transport position so that it can be moved for repairs.

¹¹ The free mode is not called manual mode to avoid ambiguity with these modes.

5.2 Basic Architecture

On the topmost level, the control system is divided into six components which will be discussed briefly. The *CAN component* is a wrapper for CAN device drivers. It provides a well defined interface to the rest of the system for sending and receiving CAN messages. It also hides all the details of a particular CAN driver. All it takes to adapt the system for different CAN hardware is to create a new wrapper for the CAN drivers.

The *ISOBUS component* contains data link layer, network layer and network management functionality. The ISOBUS component also transmits the object pool to the VT or loads it from the terminal's non-volatile memory, if the object pool is already stored on the VT. The ISOBUS component can be parameterized and reused without modification.

The *hardware component* is a hierarchical presentation of the implement. This component is highly machine dependent and cannot be reused as such. However, the components deeper in the hierarchical structure are more likely candidates for reuse. For example, if some machines have the same subsystems, they can be reused with little or no modification. Even if machines do not have similarities at the subsystem level, they can still use the same hardware components such as hydraulic cylinders and electric motors. At the device level, most of the components can be parameterized and reused. Although it is easier to reuse lower-level components, it is not as beneficial as reusing higher-level components.

The *mode component* manages the major modes of the control system (see section 5.1) by sending signals to the hardware component. This component is machine dependent as the implementation of the modes varies. However, an existing implementation can be used as a template and then modified to meet the new requirements.

The *I/O component* represents the machine specific I/O system. The I/O component uses the I/O driver to interact with the physical hardware. It provides an easy to use interface for the hardware component which hides the I/O system details. For example, input signals are scaled to engineering units, noisy signal are filtered and additional signals can be created by differentiating or integrating existing ones. For logging purposes, some signals can also be written to a file in a centralized manner.

The *I/O driver* is a wrapper for I/O device drivers and can be reused without modification. Unlike the CAN driver, the I/O driver cannot provide a standard interface. The exact I/O interface always depends on the actual I/O hardware. For example, the number of digital and analog inputs and outputs varies from device to device.

5.3 User Interface Design

The user interface designs are based on the existing commercial UIs and the experiences gained from the fast prototype. Later, the UIs were further refined

by conducting usability studies with real users. The actual ISO 11783 user interfaces were created with a home-grown Java tool designed for building object pools. The details of the UI design are not discussed in this paper.

6 Implementation

The implementation of the control system is done in a bottom-up manner. The low-level components are implemented and tested first and then the implementation effort is focused on higher-level components. Subsystems are tested as early as possible to find out possible design problems. For example, the CAN component was tested with a simple *driver* component. Driver components were also made for I/O and ISOBUS components so that they could be tested separately. Early found design problems are easier to fix, because the affected upper-level interfaces are not yet implemented.

As the design and implementation progress, new components are created and placed into component libraries or repositories as they are called in Constellation. Reusable *CAN*, *I/O* and *ISOBUS* components are placed in corresponding repositories. A separate repository is used to hold the generic low-level components that model physical machine parts. The Simulink controllers and fault diagnostic components are in their own repositories mainly because they were created by different researchers. Three machine specific repositories are used to hold the non-reusable high-level components, which are the “glue” that hold the control applications together.

6.1 Controllers

Some simple controller DFCs are implemented directly with *C++* but the more complicated controllers and filters are created with Simulink which is a widely used tool for modeling, designing, simulating and tuning controllers. Constellation is integrated well with Simulink which makes importing Simulink models easy. As an example, the sprayer pressure controller is shown in Figure 2.

6.2 Finite State Machines

Finite state machines are used for many things, such as describing the user interface logic, controlling machine modes, implementing marker and tramline automation, specifying alarm logic and defining calibration sequences. FSM components are also used extensively to implement protocols and procedures defined in ISO 11783¹². Creating finite state machines may not be easier or faster than writing conventional code, but it is less error-prone and the result is much more readable – it is almost self documenting! The mode control logic of the no-tillage drill is shown as an example in Figure 3.

¹² ISO 11783 defines many packet transport protocols for long messages and an initialization procedure for connecting to the virtual terminal.

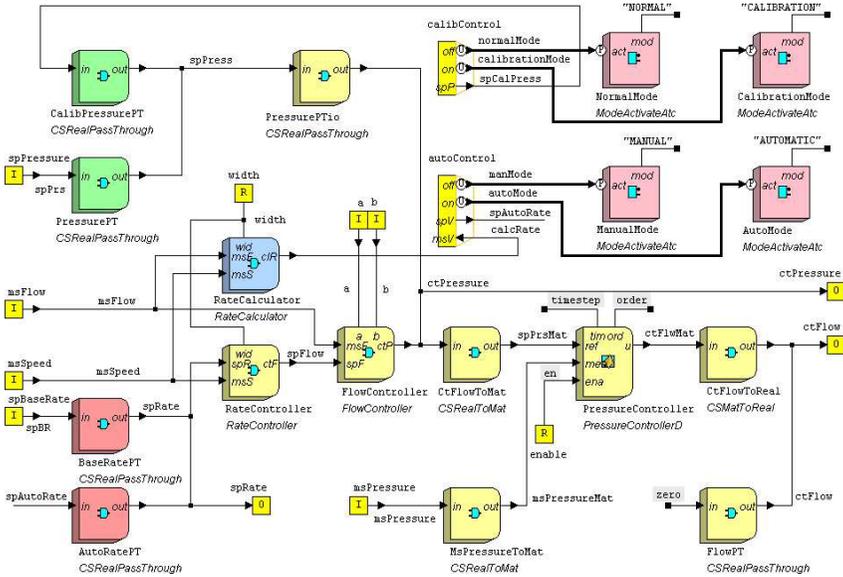


Fig. 2. Sprayer pressure controller COG

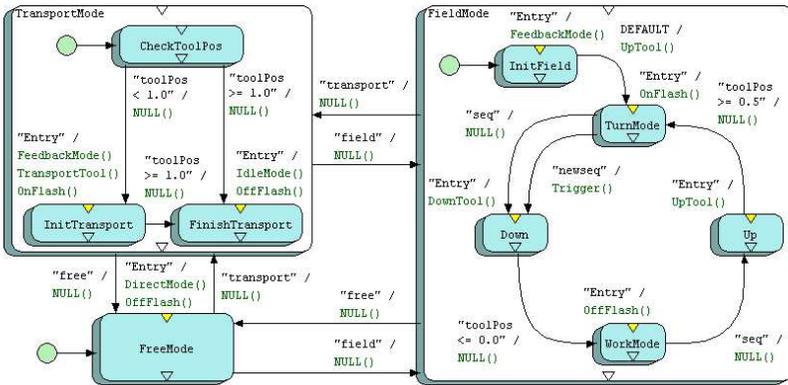


Fig. 3. No-tillage drill user interface logic FSM (STCs and interfaces are not shown)

7 Simulation and Debugging

Unit testing is simple with Constellation. Sometimes it is enough to create a test application, add the component and specify its initial values. In more complex scenarios, it may be necessary to create a *driver* and *stub* components which are used to exercise the component under test and to give suitable responses, respectively. The test application can then be compiled and run. At run time, the signals can be changed and monitored from application’s interactive shell. In addition to the shell, Constellation provides more advanced debugging facilities. *StethoScope* is an oscilloscope like tool for monitoring sig-

nals and *LiveLook* is used to animate state diagrams. These tools use TCP/IP to connect to the running program, so they can be used to monitor programs running on either the host or the target system in real-time.

8 Cases

The configurability of the control system was tested with three case implementations. A short summary of the implements and their control functions is given in table 1.

Table 1. Case implements and their control functions

| Pneumatic Drill | Sprayer | No-Tillage Drill |
|-----------------------|------------------------|------------------------|
| hitch position | left boom folding | machine elevation |
| coulter pressure | right boom folding | coulter pressure |
| coulter unit folding | boom elevation | coulter mode selection |
| front drag position | boom tilting | tramline device |
| front drag folding | spray application rate | fertilizer appl. rate |
| left marker | left marker | left marker |
| right marker | right marker | right marker |
| tramline device | section valves 1–5 | |
| seed application rate | | |
| fertilizer appl. rate | | |

8.1 Pneumatic Combined Seed and Fertilizer Drill

The pneumatic drill is the most complex of the three implements. The main difficulty with this machine is the lack of sensors. Some actuators are driven using timers and their state can be estimated after the controls have been applied the first time. However, the controller cannot know the initial state and the potentially dangerous actuators cannot be automatically driven to a known position at startup. This problem is solved by starting the implement in the free mode with all automation disabled and by notifying the operator about the up-coming automatic controls before changing modes.

Another difficulty with this machine is the coordination of the front drag and the coulter unit during headland sequences. This issue is solved by adding two submodes to the field mode. Because the mode component coordinates all mode changes in a centralized manner, the front drag and coulter unit components do not need to know anything about the headland sequences.

8.2 Sprayer

The sprayer has basically the same operation modes as the drills. Perhaps the biggest difference is that sprayer's application rate is not mechanically coupled to its speed. To keep the application rate constant the feeder controller has to continuously adjust the flow rate for the current speed. This is simple to implement but in some cases, e.g. washing the tank or mixing the batch, the operator may need to adjust the pressure and flow directly even if the machine is not moving. This issue is solved by adding a new mode to the pressure controller, which is active in the free mode.

Another difference to the other implements is that none of the hydraulic functions were included in the automation system. There is no need to automate the boom folding or boom elevation as the sprayer does not have similar headland sequences as the drills have. Instead, these functions are operated manually using tractor's standard hydraulic outputs, which simplified the system to some extent.

8.3 Combined Seed and Fertilizer No-Tillage Drill

The no-tillage drill is quite similar to the pneumatic drill. It is a simpler machine as there are no folding functions and the seed application rate can only be adjusted manually. What makes this machine interesting is that it does not have any electro-hydraulic valves. Instead, hydraulic power is requested from the tractor with standard ISO 11783 messages. This simplifies the machine and makes it more affordable but it requires an ISO 11783 class 3 tractor.

9 Conclusions

A software framework and component libraries for building ISO 11783 implement control systems were developed with RTI's Constellation software development system. Prototype control systems for three real agricultural implements, two drills and one sprayer, were implemented. The prototypes have worked reliably during field tests.

The initial design and implementation of the first and the most complicated drill control system took about four man-months, because the required components had to be created from scratch. The sprayer controller was implemented after the first drill. Even though the sprayer is a very different machine, it took only about two man-months to complete. CAN, I/O and ISOBUS components could be reused without modifications as well as many of the lower level components. The general architecture and the three operating modes could be adapted and refined from the first machine. The third control system was developed in a month, partly because of the similarities to the other drill and partly because the researchers had become more proficient with the Constellation tool.

Designing libraries of loosely coupled, reusable components is difficult. Nothing can replace the thorough understanding of the problem at hand, but Constellation can help avoiding common pitfalls by enforcing widely-accepted and proven software principles such as “*always program against an interface not an implementation*”. The principles are of no use if the language makes it too difficult to follow them consistently. While component reuse is important, frameworks allow reuse at application level. A well designed application framework solves many central design decisions in an elegant manner. The Constellation framework makes use of multiple *design patterns*, such as composite, observer and adapter patterns. The use of design patterns improves the general quality of the code. However, frameworks are notoriously difficult to design and creating one for a one-time solution is a waste of resources.

The prototype control systems evolve as new features, such as better controllers and fault diagnostics components, are added and the user interfaces are refined. Maintaining the control systems is easy due to their hierarchical structure and loosely coupled components. Hierarchical structure splits the diagrams into manageable pieces. Hierarchical components can be reused at different levels. Because of the loose coupling, modifications usually affect only a few components. Loose coupling also reduces the risk of unwanted side effects in the other parts of the program. Navigating within the application and plugging in new components is very intuitive with the graphical tool.

Although the three machine control systems are all different and no apples-to-apples comparison can be done, we feel that Constellation, or another similar tool, can be used to improve software quality and programmer productivity. However, these improvements stem from component reuse, which is exploited best only if multiple similar control systems are created. For creating a single control system, a more traditional approach might be more appropriate.

References

1. Bosch Robert GmbH (1991) CAN Specification, Version 2.0. Germany
2. Brooks F. Jr. (1987) No silver bullet: essence and accidents of software engineering. *Computer* v.20 n.4, p.10–19
3. Christensen J (2000) Basic Concepts of IEC 61499. <http://holobloc.com>. 24 October 2000
4. Douglass B (2004) Real-time UML, third edition. Addison-Wesley
5. Gamma E, Helm R, Johnson R, Vlisside J (1995) Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, Massachusetts
6. NMEA (2000) NMEA 2000: The network standard for interfacing marine electronics devices. National Marine Electronics Association
7. RTI (2003) Constellation User’s Manual. Real-Time Innovations Inc.
8. Stone M, McKee K, Formwalt C, Benneweis R (1999) An Electronic Communications Protocol for Agricultural Equipment. Agricultural Equipment Technology Conference, Louisville, Kentucky. 7-10 February 1999

Long-Term Activities for Autonomous Mobile Robot

– Autonomous Insertion of a Plug into Real Electric Outlet by a Mobile Manipulator –

Tomofumi Yamada¹, Keiji Nagatani², and Yutaka Tanaka¹

¹ The Graduate School of Natural Science and Technology, Okayama University,
3-1-1 Tsushima-naka Okayama, JAPAN,
{t.yamada, field}@usm.sys.okayama-u.ac.jp

² The Graduate School of Engineering, Tohoku University, Aramaki aza Aoba
6-6-01, Sendai 980-8579, JAPAN, keiji@ieee.org

Summary. Mobile robots used in the human-robot coexisting environment are required to perform continuous works without human maintenance. On the other hand, they need a rechargeable batteries that require charge, generally. Therefore, an autonomous battery-charging for mobile robots has a big advantage for performing continuous works. However, installation of exclusive use of battery-charging-stations requires much cost.

To improve this situation, we aim to realize an autonomous battery-charging motion for a mobile manipulator using conventional electrical outlets. In this motion, the robot is navigated to a front of an outlet, and a plug attached at the tip of the manipulator is controlled to insert into the outlet. To realize the motion, we implemented “distance transform method” for navigation, and a motion of plug insertion using force feedback control.

In this paper, we explain the above implementations, and we discuss advantages and limitations of such motions.

Keywords: Battery-charging, navigation, putting a plug into an outlet

1 Introduction

Mobile robots used in the human-robot coexisting environment are required to perform continuous works without human maintenance. Both (1) mounting batteries and (2) mounting power cables are ways to supply electric power to robots in common cases. In the former case, a free-maintenance time for robot is limited by capacity of its batteries. In the latter case, an activity area of robot is limited by length of cables. Therefore, autonomous battery-charging systems for mobile robots has a big advantage to perform

autonomous and continuous works. So, in this research, we aim to realize an autonomous battery-charging system for an autonomous mobile robot to enable long term activity of it.

There were several researches that aimed to realize long term activities of autonomous mobile robots. Yuta's work aimed to realize long-term activities for autonomous mobile robot including battery-charging motion [6]. Milo developed a docking station for recharging of autonomous robots [7]. Recently in commercial robots, there are some robots that have an autonomous recharging function, e.g. Roomba [8], AIBO [9]. However, in the above cases, exclusive battery charging stations are used. Usually, an installation of such stations requires cost.

In this research, our final goal is to realize an autonomous battery-charging motion for autonomous mobile manipulators using common electrical outlets in an actual environment. It requires motions of "navigation" and "putting a plug into an outlet". The robot is navigated to a front of an outlet, and a plug attached at the tip of the manipulator is controlled to insert into the outlet. To realize the motion, we implemented "distance transform method" for path planning of a base robot and a motion of plug insertion using force feedback control.

In this paper, we explain implementations of both navigation and plug insertion, and discuss advantages and limitations of such motions.

2 Problem Definition and Assumption

To realize long-term activities for autonomous mobile robots, there are many research topics (e.g. re-scheduling tasks, an architecture of operating system and so on). In this research, we focus on a basic function "autonomous battery charging", and we set our objective motions as "(1) autonomous navigation and (2) putting a plug into outlet". Assumptions of these motions are as follows.

- A type of target robot is a mobile manipulator.
- A plug is installed at the center of the manipulator's hand.
- A target environment is our laboratory.
- Location of an objective outlet is known.
- An initial position of the robot is known.
- No marks and guides are installed in the target environment.

These assumptions are proper because the target environment is a human-robot coexisting environment and the cost of altering an environment should be reduced as much as possible. Additionally, an usual mobile robot working in indoor environments can have a map.

A view and a layout of the target environment are shown in Fig.1 and Fig.2. This is one of common office environments, and some outlets exist in walls.



Fig. 1. Photograph of the environment

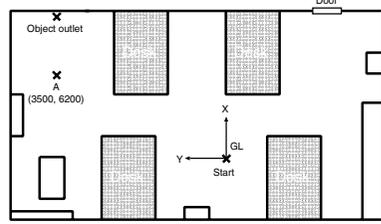


Fig. 2. Layout of the environment

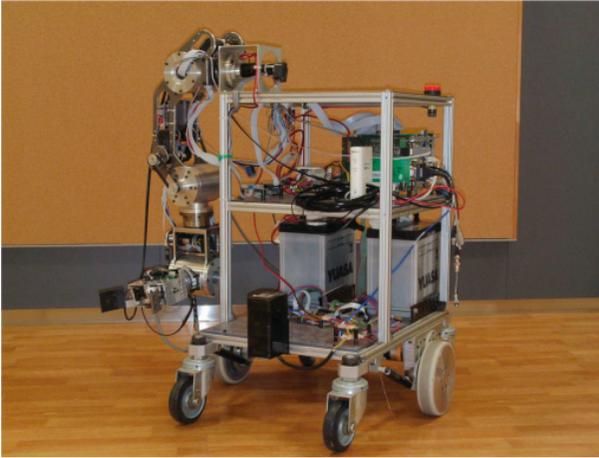


Fig. 3. Target robot (Autonomous mobile manipulator)

3 Target Robot

Target robot used in this research is an autonomous mobile manipulator. A view of the robot is shown in Fig.3.

1. Base robot

The size of the base robot is 56[cm] in width, 71[cm] in depth and 80[cm] in height. The weight is about 30[kg] which includes two batteries. Controllers and normal sensors are contained in the robot's body to be self-contained robot. It has a power-wheel-steering for locomotion.

2. Manipulator and hand

The 6 D.O.F. manipulator is mounted at the right of the base robot. It is a PUMA-configuration which was produced in our laboratory (made by aluminum). At the tip of the manipulator, a two finger hand is installed



Fig. 4. 2-D Range sensor

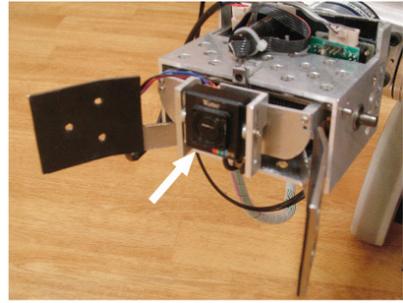


Fig. 5. Vision sensor and hand

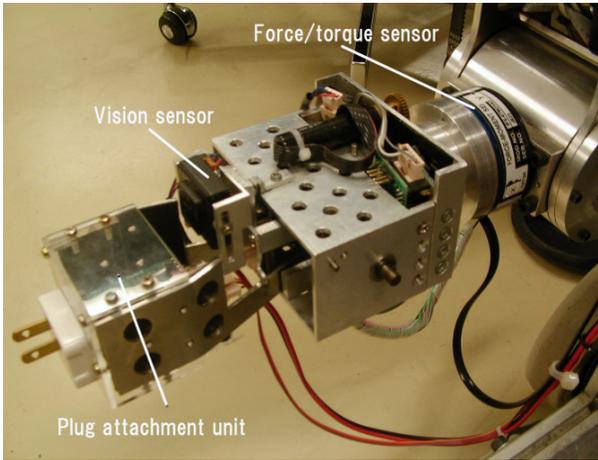


Fig. 6. Plug attachment unit and force/torque sensor

shown in Fig.5. A force/torque sensor is installed between the manipulator and the hand shown in Fig.6.

3. Plug attachment unit

A plug attachment unit for battery charge is installed at the center of the hand (Fig.6). It contains a tilt sensor ADXL202AE (ANALOG DEVICES corporation) to adjust the unit's posture. Currently, the plug is a dummy.

4. Force/torque sensor

The force/torque sensor is IFS-67M-25A 25-1 (Nitta corporation) which can detect 3 dimensional force and 3 dimensional torque. It is used for (1) force feedback control of manipulator and (2) detection of plug insertion in this research.

5. Vision sensor

A small CCD camera WAT-230 (Watec) is installed in the center of the hand as a vision sensor. It is used to recognize the relative position of the

objective outlet to adjust hand's position. The vision sensor is lifted up mechanically to keep its view area when the finger is closed. The function can be seen in Fig.5 and Fig.6.

6. Range sensor

Two dimensional range sensor PB9-11 (Hokuyo automatic corporation) is installed in front of the base robot (Fig.4). It scans 81 steps of half-circle area to measure length to objects using LED beam. It is used for acquiring environment information and position adjustment for the base robot.

4 Autonomous Navigation

The robot should move to the closest outlet when it requires battery charging. In our assumption, the robot can have a map of the target environment and information of outlet locations in advance. However, because of movable chairs and other obstacles, sensor based navigation is necessary. In this section, we introduce an implementation of sensor-based path planning using a range sensor.

4.1 Path Planning

The start position and the outlet position (goal position) are assumed to be known. On the other hand, we assume that a map of the environment is unknown because of movable chairs and other obstacles. Considering these assumptions, we apply distance transform method [3][4] for a sensor based path planning to navigate the robot to the goal position.

Usually, the method assumes that the environment is fully known. Instead, we use a range sensor to detect local environment information to plan local path. A concrete procedure of distance transform method is shown as follows.

Firstly, the local area (within sensing range) is divided into small grids to generate a grid map, and each grid has a binary information (free or occupancy). Basically, each grid's information is detected by range data measured by the two dimensional range sensor. Additionally, the obstacle area is expanded with consideration of the robot's size, and grids in the occluded area from the robot are registered as occupancy. Fig.7 shows an example of this procedure. In this implementation, the local area is set as $3[m]$ in length and $2[m]$ in depth, and each grid size is set as $10[cm]$ square.

Secondly, distance transform method is applied to the grid map from the goal position to generate the shortest path. If the goal position does not exist in the grid map, a sub-goal is set at the furthest position in free grids from the robot's position.

Thirdly, the robot navigates along the planned path to the goal (or sub-goal) based on odometry information. The above procedure is repeated until the robot arrives at the goal, as shown in Fig.8.

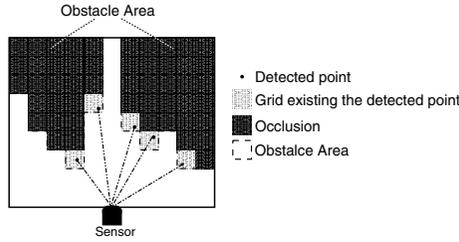


Fig. 7. Registration of grids' property

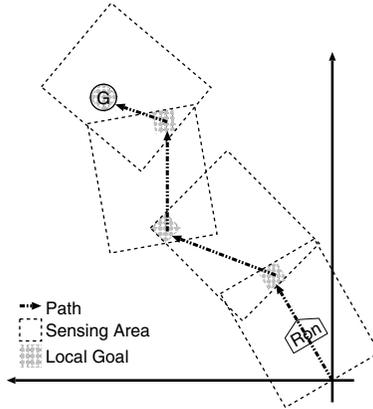


Fig. 8. An overview of repeating procedure

4.2 Adjustment of Position and Orientation

Theoretically, the robot reaches the goal position exactly if it follows the planned path. Practically in the real environment, it is impossible because a positioning error accumulates due to the error of the wheel diameter, the slippage between wheels and the ground and bumps of the ground. Therefore, an adjustment motion of the base robot's position is necessary for reduction of the error for plug insertion.

In this research, the robot adjusts its position and orientation by detecting distance to the wall and inclination of the wall (using two dimensional range sensor) when the robot arrives at the goal position. Hough transform method is used to abstract the wall's line from the range data (Fig.9).

4.3 Navigation Performance in Real Environment

We implemented the above method on the target robot, and executed the planning and navigation in the target environment. The total length of the path was about 5 [m], and there were some chairs and desks on the way to

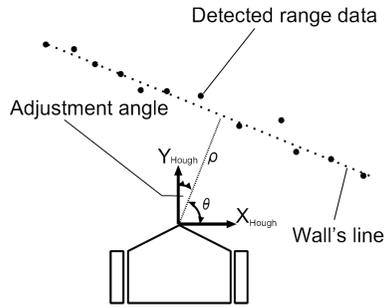


Fig. 9. Adjustment motion using Hough transform method

the goal position in this environment. By repeating the planning and navigation experiment in the same environment, the successful ratio to come to the vicinity of the outlet was about 60%.

4.4 Discussion in Failure Cases of the Navigation

- **Problem of obstacle detection**

In this research, obstacles are detected by the range sensor, and the sensor (installed at the height of 20[cm]) does not cover obstacles in high position. The lack of the sensing ability causes collision to such obstacles. To solve the problem, additional range sensors can be effective.

- **Problem of local goal selection**

The algorithm does not guarantee a completion of a path planning of the base robot, even if a path exists. One of the big reasons is that the range sensor is impossible to detect occluded area. We tried to avoid such situations in several heuristic methods, but the navigation sometimes failed because of a selection of useless local goal in some failure cases.

- **Problem of the error of odometry**

In this implementation, position adjustment of the base robot is performed when the robot arrives at its goal. However, the robot sometimes did not reach the goal position because of the accumulated error. To solve such problems, the robot should adjust its position using some landmarks while it navigates to the goal position.

5 Putting a Plug into an Outlet

After arrival of the robot in front of the target outlet, it puts a plug into the outlet using the manipulator's motion. In our assumption, the robot knows a position of the outlet, so the insertion can be completed by the pre-planned

motion theoretically. However, practically, the adjustment of hand's position and force feed-back control are necessary because of positioning error of the base robot. Therefore, the following element motions are required to realize an autonomous battery charging : (1) position recognition of an outlet, (2) adjustment of hand's position, (3) a motion of plug insertion with force-feedback control and (4) judging completion of the insertion. The element motions are introduced in the following sections.

5.1 An Outlet Recognition

To adjust the hand's position, the robot recognizes an outlet position using vision sensor which is installed at the hand. A conventional template matching method is used to recognize the outlet. The matching calculation is performed in IP5000 vision board (Hitachi Corp.) that uses normalized correlation technique.

Fig.10 shows the template image which is acquired in advance, and Fig.11 shows an example of matching result. Once the robot stops parallel to the wall correctly and the target outlet is located in the camera's image, it is almost 100 percent successive ratio for the matching procedure in our experimental setup. That is because the template image includes good texture for template matching.

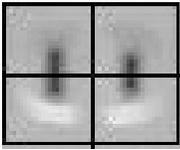


Fig. 10. The template image

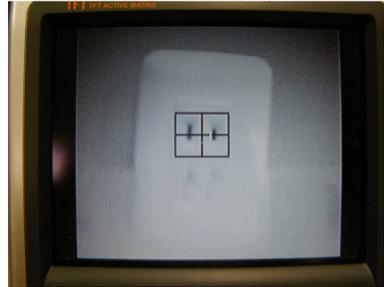


Fig. 11. An example of matching

5.2 Adjustment of Hand's Position

After the recognition of the outlet, pixel distance between the center of the image plane and the outlet position can be transferred into the actual length as a relative error of the hand's position. It is the unique value because the robot knows the distance from the hand to the outlet. Using the relative error, the robot adjusts the hand's position to align to the target outlet. However, we faced another problem as follows.

In the real motion, the error of the hand's pose still remain due to the backlash of the manipulator and a small tilt of the base robot. Sometimes this error is fatal and the robot can not perform the inserting motion with the error. Fig.12 and Fig.13 shows an example view of this problem.

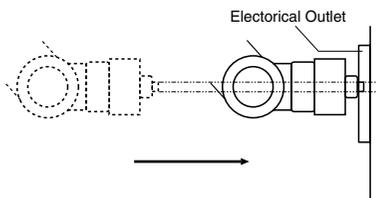


Fig. 12. Insertion without error

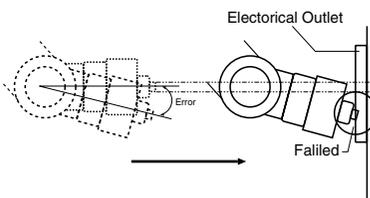


Fig. 13. Insertion with error

Usual approach to solve such problem is to improve accuracy of the hardware. However, such improvement has a limitation, and increases robot's weight that is not suitable for small indoor robot.

Instead of improvement of the hardware, our approach is to install tilt sensor at the plug attachment unit to keep hand's posture parallel to the ground.

5.3 Motion of Plug Insertion

We separated a plug insertion motion into 2 stages:

Stage 1: Approaching motion of the hand parallel to the ground.

Stage 2: Motion of putting the plug into the outlet.

In the stage 1, the hand is traced along the straight line to the center of the outlet without force-feedback control (Fig.14). When the plug contacts to the outlet, the contact force can be detected by the force/torque sensor, and the robot switches to the stage 2. In this stage, force feed-back control is applied to compensate large contacting force, shown in Fig.15. It is a very simple force-feedback control, the detected force values at the tip of the manipulator are converted into differential angles of joints (calculated by the inverse Jacobian matrix), and the angles are added to the reference angles of the joints.

Since there is a dent around the hole of conventional outlet to make plug insert easy for human, the motion of plug insertion can be performed to slide along the dent, using force-feedback control.

5.4 Judgement of Completion of the Insertion

In this research, we use force data measured by the force/torque sensor for a judgement of completion of the insertion. We implemented both a passive method and an active method for the judgement, shown as follows.

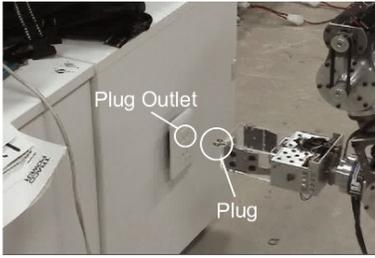


Fig. 14. Adjustment without compliance motion



Fig. 15. Insertion with compliance motion

Judgement using passive method

The passive method is to use a log of force data during the motion of plug insertion. Force data log of z axis (the same direction as perpendicular to the outlet) in a typical successful case is shown in Fig.16, and force data log of z axis in a failure case is shown in Fig.16. Both the data were logged when the hand is controlled along a straight line with a manual position adjustment.

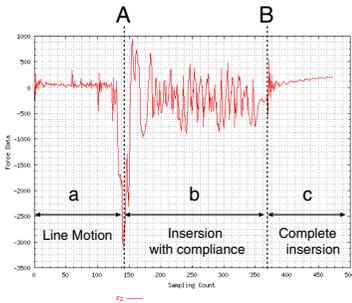


Fig. 16. Force data of successful case

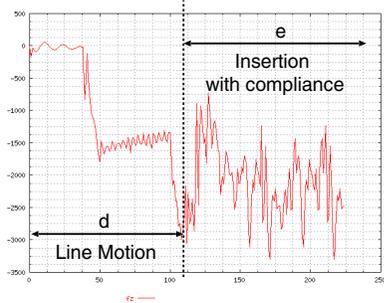


Fig. 17. Force data of failure case

Features of the above cases are shown in the following.

- [a :] In the stage 1 of successful case, the swing of the data is not so large.
- [b :] In the stage 2 of successful case, the data is oscillated due to force-feedback control.
- [c :] After completing insertion of successful case, the change of the force data increases linearly.
- [d :] In the stage 1 of failure case, detected force does not exceed the threshold even if the plug contacts to the outlet in this example.
- [e :] In the stage 2 of failure case, the force data is continuously increasing in average and the swing of the data does not converge.

The big difference feature is the data in the stage 2 ([b] and [e]). The log pattern in the successful case [b] is oscillated around the initial value, because the plug is inserted correctly. On the other hand, the log pattern in the failure case [e] is oscillated around the offset value, because the plug always push the outlet. Using this feature, the robot can judge completion of the insertion.

Judgement using active method

In our experiences, the passive judgement using force sensor sometimes gives a wrong answer. To confirm the plug insertion, we implemented the active method which uses additional motion of the hand.

If the plug is inserted correctly, the plug and the outlet are united. Therefore, when the robot tries to rotate the hand, the large torque can be detected at the wrist of the hand because the plug can not move. It is performed when the robot finishes inserting motion.

It is very simple and powerful method for judgement of completion of the insertion.

5.5 Plug Insertion Performance in Real Environment

We implemented the above method on the target robot, and executed the plug insertion motion in the target environment. The initial position of the base robot was set by eye measurement. By repeating the motion, the successful ratio to insert the plug was about 40%. Note that we counted failure case when the robot detected a failure of plug insertion.

5.6 Discussion in Failure Cases of the Plug Insertion

- **Problem of the recognition of the outlet**

When the robot recognized the outlet, sometimes the template matching method was not succeeded in. We guess that the brightness in the environment was changed. (The normalized correlation technique did not work well in this case.) To solve the problem, installing a flashlight at the hand is effective to keep brightness constantly.

- **Problem of a lack of the stiffness at the manipulator base**

When the manipulator is stretched (or the manipulator is overhanged from the base), a physical deformation of the base of the manipulator occurs due to the lack of the stiffness at the base part. This is a peculiar problem of our hardware, and it affects to successful ratio in large. A part of the problem can be adjusted by the tilt sensor shown in the section 5.2, but we will improve the part of the manipulator base.

6 Conclusion

In this research, we assume our research objective as “autonomous battery charging using common outlets” to realize the long activity for a mobile robot. We separated the motion into two sub-motions, “autonomous navigation” and “motion of plug insertion”, and reported each implementation and performance using a real robot. Currently, the two motions are performed separately, and the integrated motion is very low level in the successful ratio. One of the reasons is that the positioning error of the base robot in navigation disturbs the start of motion of plug insertion.

Although the successful ratio is currently not enough practically, we guess that the ratio can be increased very much by fixing the problems shown in this paper. Therefore, our future work is to search and to fix problems by using a real robot in a real environment. Finally we aim to realize long term activity for mobile robot.

References

1. Jean-Calude Latombe(1997) Robot Motion Planning. Kluwer Academic Publishers, Dordrecht
2. Keiji Nagatani, Shin'ichi Yuta (1998) Autonomous Mobile Robot Navigation Including Door Opening Behavior (System Integration of Mobile Manipulator to Adapt Real Environment). In: Field & Service Robotics Springer-Verlag, pp. 195-202
3. Alexander Zelinsky Shin'ichi Yuta (1993) A Unified Approach to Planning, Sensing and Navigation for Mobile Robots. In: Preprints of the Third International Symposium on Experimental Robotics
4. Alexander Zelinsky(1994) Using Path Transforms to Guide the Search for Find-path in 2D. In: International Journal of Robotics Research, Vol.13, No.4, pp. 315-325
5. TH.Meitinger, F.pfeiffer(1995) The Spatial Peg-in-Hole Problem. In: IEEE/RSJ Cong. on Intelligent Robots & Systems, VolIII, pp. 54-59
6. Shin'ichi Yuta, Yasushi Hada(1998) Long term activity of the autonomous robot -Proposal of a bench-mark problem for the autonomy-. In: Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1871-1878
7. Milo C.Silverman, Dan Nies, Boyoon Jung and Gaurav S.Sukhatme(2002) Staying Alive: A Docking Station for Autonomous Robot Recharging. In:IEEE International Conference on robotics and Automation, pp. 1050-1055
8. <http://www.irobot.com/> (Roomba)
9. <http://www.jp.aibo.com/> (AIBO)

Synthesized Scene Recollection for Robot Teleoperation

Naoji Shiroma¹, Hirokazu Nagai², Maki Sugimoto², Masahiko Inami² and Fumitoshi Matsuno^{1,2}

¹ International Rescue System Institute, Minami-Watarida 1-2, Kawasaki, Kanagawa 210-0855, Japan naoji@hi.mce.uec.ac.jp

² University of Electro-Communications, Chofugaoka 1-5-1, Chofu, Tokyo 182-8585, Japan {hnagai, sugimoto, inami, matsuno}@hi.mce.uec.ac.jp

Summary. In this paper we propose an innovative robot remote control method, a synthesized scene recollection method, which provides the operator with a bird's-eye view image of the robot in an environment which is generated by using position and orientation information of the robot, stored image history data captured by a camera mounted on the robot, and the model of the robot. This method helps the operator to easily recognize the situation of the robot even in unknown surroundings and enables the remote operation ability of a robot to be improved. This method is mainly based on two technologies, robot positioning and image synthesis. In this paper we use scan matching of laser rangefinder's scan data for robot positioning and realized self-contained implementation of the proposed method in 2D horizontal plane.

Keywords: Teleoperation, positioning, image synthesis, scene recollection

1 Introduction

In the teleoperation of a mobile robot in a remote site, the controllability of the robot will increase greatly if the operator can easily recognize and understand the situation of the robot in the remote site and its unknown surroundings.

Many studies on the control methods of the teleoperation of mobile robots have been investigated and proposed up till now. System structure in most of the previous studies uses a system where there is a mounted camera on the robot and the operation is usually performed from a remote site using captured images by the mounted camera. If you have ever experienced operating a mobile robot using such system structure, you would agree that it is difficult to operate a robot by only using a direct camera image and controllability of the robot is very different from operating a robot close to you. This is mainly because it is hard to understand the situation of the robot and its surroundings based on only the information of captured images unless you

are well trained in robot operation, possesses good sense of space perception and are good at imagining the robot itself in the unknown environments.

Obtaining 3D environmental data of the unknown surroundings and constructing it into a 3D model of the surroundings [1], adding extra mechanisms on a robot from where the mounted camera can capture bird's-eye view like images of the robot [2] [3], and using vision support of other robots [4] [5] are some of the ways to overcome the teleoperation difficulties. Even though these methods have some disadvantages such as long process time is required to construct a 3D model of the unknown surroundings and cannot handle dynamically changing environments, and increase cost, size, weight, complexity [6] and the number of robots. Another difficulty in the mobile robot teleoperation is the communication efficiency. Sometimes in bad communication conditions it is hard to send captured images as the data size of them is usually large.

We have proposed an innovative teleoperation method, a synthesized scene recollection method, which increases the controllability of a robot by using stored images captured by the camera mounted on the mobile robot as spatial-temporal information [7]. This method can deal with the above mentioned difficulties and disadvantages in the teleoperation of mobile robots. Plainly speaking, this is the teleoperation method which uses a bird's-eye view of a robot in unknown surroundings, is synthesized from spatial-temporal information of formerly captured images.

This teleoperation method is developed as a rescue robot technology. Since it is still difficult to develop autonomous robots to function well in real environments with current robot technology, the system structure such as a human operator remotely controlling a robot is one of the realistic solutions which works well in real disaster sites during rescue robot operation [2] [3] [8] – [10]. Although this method is implemented as part of rescue robot technologies, it can also be applied to any moving device such as medical surgery support which uses an optical fiber scope.

Although high mobility of a mobile robot is indispensable for rescue robots in actual disaster sites, it is hard to make full use of the mobility of the robot by ordinary teleoperation methods since the situation of the robot and its surroundings are vaguely known. The proposed method can overcome this problem and can make full use of the locomotion ability of the robot and it can also increase its mobility.

In teleoperation of a manipulator with system time delay, there are works which use the predictive display and/or force feedback information based on environmental model constructed in the computer system to deal with the time delay [11] [12]. Our method does not handle system time delay but can switch mode of data transmission according to contents of data such as use low transmission rate for images which are large in data size and high transmission rate for robot position information which are small in data size. This mode switching will contribute improving the robustness of data transmission in low bandwidth communication.

This teleoperation method also introduces several other benefits such as: real time synthesis of bird's-eye view images because of image-based method and no model construction needed and can handle dynamic environment which changes in low frequency. In addition, it will also reduce blind spots, prevent the operator from getting camera motion sickness, and so on.

In the previous implementation of the method we used an external camera to measure the position of the robot. To realize self-contained system which is suitable for a realistic application such as a rescue activity use we installed a laser rangefinder and used scan matching of the laser rangefinder's scan data for robot positioning in 2D horizontal plane.

In this paper we will explain this synthesized scene recollection method, a novel teleoperation method for a mobile robot using real image data records and self-contained implementation of the proposed method using positioning by a installed laser rangefinder. With this method we can obtain the bird's-eye view image of a robot in a scene even with only a single camera.

2 Synthesis of the Bird's-Eye View Images to Improve Remote Controllability

In our work, synthesis of the bird's-eye view images, which improve remote controllability, are conducted using the following technologies:

- Estimation of the position and orientation of the robot
- Image synthesis technique for bird's-eye view images using estimated position and orientation information of the robot and spatial-temporal information which are formerly captured real image data records

That is, we need to know the position and orientation of the robot and its stored real image data records which include formerly captured images associated with the position and orientation of the mounted camera where the image was captured. Overview of the bird's-eye view synthesis is represented in Fig. 1. The upper left, center and right pictures of Fig. 1 are images currently captured by the camera, current position and orientation information of the robot, and the selected bird's-eye view like image of the robot from real image data records respectively. The bird's-eye view of the robot in its unknown surroundings shown in the bottom picture of Fig. 1 is the synthesized image using above information and a CG model of the robot which is created in advance.

An operator remotely controls the robot using the composite bird's-eye view images which are synthesized according to the process presented in Fig. 1 using real image data records captured by the camera mounted on the robot and position and orientation information of the robot measured by the sensors. The operator can easily understand the situation of the robot and its unknown surroundings in the teleoperation using these composite images and the remote controllability will increase.

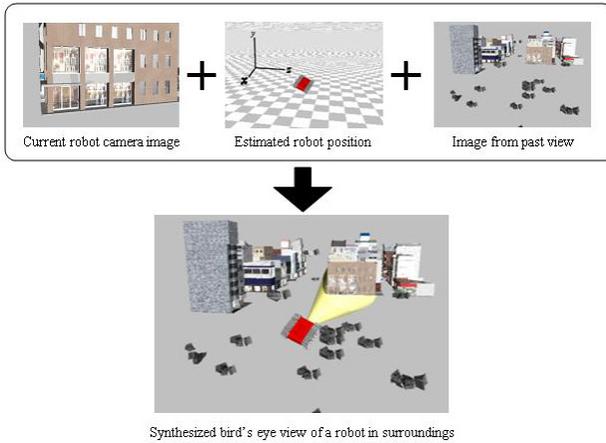


Fig. 1. Overview of the bird's-eye view synthesis.

The algorithm for synthesizing the bird's-eye view images is as follows:

Algorithm

1. Obtain position and orientation information of the robot during operation.
2. Store images associated with position and orientation information of the mounted camera when they are captured to the buffer while the robot is moving.
3. Select an appropriate image from the stored real image data records according to the current position and orientation information of the robot and make the position and orientation of the selected image as the viewing position of the bird's-eye view image.
4. Render the model of the robot according to the current position and orientation information of the robot and the selected viewing position.
5. Superimpose the model of the robot viewed from the selected viewing position onto the selected image from the stored real image data records (generation of the bird's-eye view image).
6. Repeat this procedure continuously.

Overview of this system is shown in Fig. 2. Images captured by the mounted camera are stored in the buffer as bitmap images along with the associated position and orientation information of the camera when they are captured. When the current position and orientation information of the robot is obtained from the sensors, the most appropriate image to view the robot in the present time is selected from the stored real image data records according to this information of the robot. Then the model of the robot which is viewed from the selected image position is superimposed onto the selected image. The selection of the most appropriate viewing position is according to the position

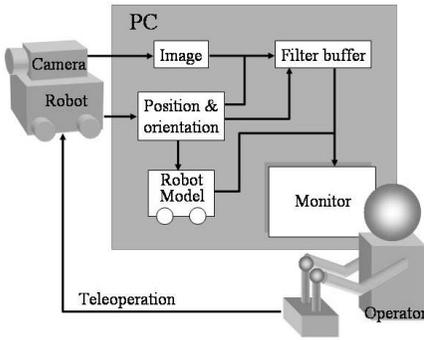


Fig. 2. System overview.

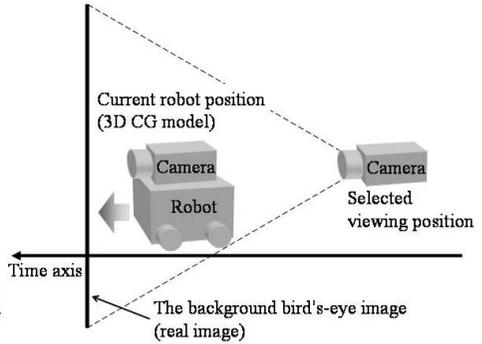


Fig. 3. Pseudo real-time view.

and orientation information of the mounted camera which is stored with the captured images in the buffer. As shown in Fig. 3, the selected image is used as the background image of the bird's-eye view image. This background image is not real-time one but it is a pseudo real-time image. Because of this system configuration it can handle dynamically changing environments in a pseudo real-time manner. Also this system does not require the construction of a 3D environmental model since this is an image-based method, and it does not take much time to synthesize the bird's-eye view image.

3 Robot Positioning Using Scan Matching

Although we have used an external camera to measure the position of the robot in the previous implementation of the method, this system configuration is not suitable for a realistic application such as a rescue activity use since we can not place external cameras in advance in unknown surroundings. For realistic implementation we need to realize self-contained system including robot positioning. In this paper we installed a laser rangefinder as a positioning sensor on the robot and used scan matching of the laser rangefinder's scan data for robot positioning in 2D horizontal plane to realize self-contained positioning and system.

3.1 Scan Matching

In scan matching two scan data from a laser rangefinder (LRF): a reference scan, R_n , and an input scan, S_n are used to determine the relative rotation, dR , and the relative translation, dt , of the LRF position. This relative rotation and translation are the same as the ones of the robot position. The ICP (Iterative Closest Point) algorithm [13] [14] which is based on the least square registration is a well known algorithm for local scan matching. In this paper

we use the ICP algorithm for scan matching and to determine the relative rotation and translation of the robot position.

The algorithm used in this paper for scan matching in 2D horizontal plane is as follows:

Scan Matching Algorithm

1. *Determine closest point pairs*

Find a closest point $r_i \in \mathbb{R}^2$ in the points of the reference scan data, R_n , which corresponds to each point $s_i \in \mathbb{R}^2$ for all the points in the input scan data, S_n .

2. *Suppress bad closest point pairs*

Ignore closest point pairs (s_i, r_i) of the input and reference scan data whose point distances $|s_i - r_i|$ are larger than the specified threshold distance δ .

3. *Subtract centroids of the scans from the scan data*

Calculate centroids of each scan, s_c and r_c . Subtract corresponding centroid from all the closest point pair (s_i, r_i) .

$$s_c = \frac{1}{N} \sum s_i \quad r_c = \frac{1}{N} \sum r_i \quad (1)$$

$$s'_i = s_i - s_c \quad r'_i = r_i - r_c \quad (2)$$

Here, N is the number of the closest point pairs.

4. *Calculate the correlation matrix*

Correlation matrix H can be obtained as follows:

$$H = \sum r'_i s'^T_i = \begin{bmatrix} h_{xx} & h_{xy} \\ h_{yx} & h_{yy} \end{bmatrix} \quad (3)$$

5. *Calculate the small relative rotation and translation*

The small relative rotation, dR , and translation, dt , can be obtained using the SVD (Singular Value Decomposition) of the correlation matrix H as follows:

$$H = UDV^T \quad (4)$$

$$dR = VU^T \quad (5)$$

$$dt = r_c - dR s_c \quad (6)$$

Here, the matrix D is a diagonal matrix whose diagonal elements are singular values of the matrix H . The matrices U and V are orthonormal matrices which contain right and left singular vectors as their column vectors that correspond to the diagonal elements (i.e. singular values of the matrix H) of the matrix D and placed in corresponding order, respectively.

6. *Move the input scan by (dR, dt)*

Move the input scan data by the obtained relative rotation and translation (dR, dt) .

7. *Repeat this procedure continuously*

3.2 Robot Positioning

The total rotation and translation of the robot can be obtained by accumulating the small relative rotation, dR , and translation, dt , at each time step. We calculated the robot position using scan matching as following algorithm:

Robot Positioning Algorithm

- 0) Take the first scan at the initial position of the robot and register the scan data as the reference scan. Following scans are used as input scans unless the specified conditions are met.
 1. When the next scan is obtained, use the scan as the input scan and do scan matching with the registered reference scan. The relative robot motion (relative rotation and translation) will be obtained.
 2. Calculate the current robot position by adding the obtained relative robot motion to the position of the robot where the reference scan was registered.
 3. Update reference scan when the robot translates the specified distance or rotates the specified angle.
 4. Go back to 1) and repeat this procedure continuously.

The specified distance and angle in step 3) can be determined experimentally according to an environment. Since the sensor error of scan matching is accumulated every time when a reference scan is updated, we update a reference scan not so often but after certain translation distance and rotation angle in which the scan matching works well. The scan matching and other processes can be performed in real-time. That is the robot positioning is performed in real-time.

It should be noted that since we have the robot position information and the scan data from that robot position, we can generate a 2D horizontal map by combining these information and stitching each scan data according to the robot position information in some accuracy.

3.3 Robot Positioning Experiment

Experimental setup

We have developed a four-wheeled rescue robot platform called FUMA for environment information gathering at a disaster site which is shown in Fig. 4 [15]. The core design principle of FUMA is to achieve fast mobility efficiency with a simple mechanical structure. A 1-DOF arm is installed at the rear end of the robot to provide a high viewing position and a center of gravity balancing device when climbing over larger obstacles. It is generally understood that wheeled robots, without special mechanisms, endure many difficulties when climbing over objects that are higher than its wheel radius. Nevertheless, incorporating this simple structure arm, FUMA is capable of climbing over obstacles that are much larger than the radius of its wheels.



Fig. 4. A four-wheeled rescue robot FUMA with LRF. FUMA has five cameras and one on the top of the arm is used for the experiment. The yellow box on the body of the FUMA is the laser rangefinder, RS4-4 which is used as a positioning sensor.

The RS4-4 (Leuze) is used as a laser rangefinder and it is mounted on the FUMA as shown in Fig. 4. The RS4-4 can scan in the range of 190 degrees in angle and 50m in distance in front of it. Its resolution is 0.36 degrees in angle and 5mm in distance. It uses the 905nm infrared laser and its scanning rate is 40msec/scan.

Positioning experiment

We have conducted a positioning experiment using the RS4-4 mounted on FUMA. Each parameter is set as follows:

- The number of scanning point at one scan: 133 (every 1.44 degrees)
- Distance threshold between closest point pairs: 500mm
- Scan matching sampling rate: 100msec
- Update condition of a reference scan: motion difference of 200mm in distance and/or 5 degrees in angle

These parameters are obtained experimentally.

In the experiment the robot moved along the L shape path drawn by the red line from the start point at the bottom right corner to the goal point at the top left corner as depicted in Fig. 5. Fig. 5 is a schematic figure of the floor where the experiment was conducted.

The positioning experimental result is shown in Fig. 6. The pink dots denote the position of the robot and the blue dots the point on the objects around the path which form the map around the path where the robot has traveled. As shown in Fig. 6 the position of the robot can be reasonably obtained by scan matching using the LRF.

Even though the positioning by scan matching using the LRF has better positioning accuracy than the positioning by the odometry using wheel rotation encoders, errors at each scan matching accumulate as the robot moves

We have realized self-contained implementation of the proposed method using the positioning method described in the previous section. One of the examples using FOV evaluated image teleoperation method in real-time is shown in Fig. 7. This is the scene of the FUMA after traveling total 25m translation and 420 degrees rotation. The images in Fig. 7 are taken at the same time. (a) is the image of the FUMA taken from the external camera which shows the reference of the scene. (b) is the image from the camera mounted on the FUMA. (c) is the synthesized image of FUMA in the environment by the proposed method. As shown in Fig. 7 it can be said that the synthesized image is well representing the situation of the robot in the environment even though it is hard to understand the robot situation in the environment by the image (b) which is taken from the camera mounted on the FUMA.

The other example using FOV evaluated image teleoperation method in real-time is shown in Fig. 8. The viewing positions are selected automatically according to how close the model of the robot can be seen in the image area. The snapshots of the experiment are represented from the left to the right and from the upper row to the bottom one as time goes by. It can be said that the situation of the robot in the surroundings of the remote site can be understood with ease and this helps the operator to control the robot.

The combination use of the synthesized bird's-eye view and the direct camera image will provide a spatial view of the environment, thus helps the operator to conceptualize the relative locations between the robot, objects and the surrounding environment. This will be effective in actual robot teleoperation. In the current implementation, when an obstacle exists in-between the robot and the selected viewing position, instead of occlusion of the robot by the obstacle, the robot could be seen through the obstacle. In this case, the operator would know the depth order of the objects and thus able to correctly understand the scene. In other words, the operator could see the robot through obstacles that could not be seen through physically, and thus enhances the controllability of the robot teleoperation.

5 Conclusion

In this paper we have presented a novel teleoperation method for a mobile robot, the synthesized scene recollection method using real image data records and the self-contained implementation of the method with the positioning by scan matching of a laser rangefinder's scan data. Some examples using implemented self-contained system which works in real-time are shown.

The proposed teleoperation method enables an operator to easily understand and recognize the situation of the robot in its surroundings which leads to the improvement of the remote controllability. This teleoperation method will also introduce several other benefits such as: robustness in low band width communication, real time synthesis bird's-eye view images because of image-based method and no model construction. In addition, it will also reduce blind



(a) External camera image of FUMA.



(b) Image from the mounted camera.



(c) Proposed synthesized image.

Fig. 7. One scene of FUMA at the experiment.**Fig. 8.** Example of FOV evaluated image teleoperation. The snapshots of the experiment are represented from the left to the right and from the upper row to the bottom one as time goes by.

spots, prevent the operator from getting camera motion sickness, and so on. This method not only can be applied to mobile robot teleoperation but also can be applied to any other applications for a moving object such as tele-surgery with an endoscope. It can be said that this self-contained implementation of the proposed method made the progress for the realistic application of the method such as a rescue activity use.

In our future work we will extend our implementation of the method to the 3D environment with some 3D robot positioning method.

Acknowledgment

This work was done as part of the Special Project for Earthquake Disaster Mitigation in Urban Areas, supported by the Ministry of Education, Science, Sport and Culture of Japan.

References

1. M. Lhuillier and L. Quan, Quasi-dense reconstruction from image sequence, *Proc. of European Conference on Computer vision*, pp.125–139, 2002.
2. T. Kamegawa, T. Yamasaki, H. Igarashi and F. Matsuno, Development of The Snake-like Rescue Robot "KOHGA", *International conference on Robotics and Automation*, pp.5081–5086, 2004.
3. N. Shiroma, N. Sato, *et al.*, Study on Effective Camera Images for Mobile Robot Teleoperation, *13th IEEE International Workshop on Robot and Human Interactive communication*, Kurashiki, Japan, 2005.
4. R. Murphy, E. Rogers, Cooperative Assistance for Remote Robot Supervision, *Presence*, **5**(2), pp.224–240 1996.
5. N. Sato, N. Shiroma, *et al.*, Cooperative Task Execution by a Multiple Robot Team and Its Operators in Search and Rescue Operations, *International Conference on Intelligent Robots and Systems*, 2004.
6. S. Tachi, Real-time Remote Robotics - Toward Networked Telexistence, *IEEE Computer Graphics and Applications*, **18**(6), pp.6–9, 1998.
7. N. Shiroma, *et al.*, A Novel Teleoperation Method for a Mobile Robot Using Real Image Data Records, *Int. Conf. on Robotics and Biomimetics*, 2004.
8. J. Casper, Human-Robot Interactions during the Robot-Assisted Urban Search and Rescue Response at the World Trade Center, *MS Thesis, Computer Science and Engineering, USF*, April 2002.
9. R. Murphy, J. Casper, M. Micire and J. Hyams, Assessment of the NIST Standard Test Bed for Urban Search and Rescue, *Proc. of the 2000 Performance Metrics for Intelligent Systems Workshop*, 2000.
10. S. Tadokoro, H. Kitano, *et al.*, The RoboCup-Rescue Project: A Robotic Approach to the Disaster Mitigation Problem, *Proc. of the 2000 IEEE International conference on Robotics & Automation*, pp.4090–4095, 2000.
11. T. Kotoku, A predictive display with force feedback and its application to remote manipulation system with transmission time delay, *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.239–246, Raleigh, USA, 1992.
12. T. Matsumaru, S. Kawabata, *et al.*, Task-based data exchange for remote operation system through a communication network, *Proc. of IEEE Inter. Conf. on Robotics and Automation*, pp.557–564, Detroit, USA, 1999.
13. P. J. Besl and N. D. McKay, A Method for Registration of 3-D Shapes, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol.14, No.2, pp.239–256, 1992.
14. Z. Zhang, Iterative Point Matching for Registration of Free-Form Curves and Surfaces, *Int. Journal of Computer Vision*, Vol.13, Issue 2, pp.119–152, 1994.
15. Y. Chiu, N. Shiroma, *et al.*, FUMA: Environment Information Gathering Wheeled Rescue Robot with One-DOF Arm, *International Workshop on Safety, Security and Rescue Robotics*, 2005.

Development of a Networked Robotic System for Disaster Mitigation — Test Bed Experiments for Remote Operation over Rough Terrain and High Resolution 3D Geometry Acquisition —

Kazuya Yoshida¹, Keiji Nagatani¹, Kiyoshi Kiyokawa², Yasushi Yagi²,
Tadashi Adachi³, Hiroaki Saitoh³, Hiroyuki Tanaka⁴, and Hiroyuki Ohno⁵

¹ Tohoku University

² Osaka University

³ IHI Aerospace Co., Ltd.

⁴ Eizoh Co., Ltd.

⁵ National Institute of Information and Communications Technology

Summary. In this paper, a newly initiated project of networked robotic system for disaster mitigation is introduced. In this project, multiple robots are coordinately operated through ad-hoc wireless communication network, including satellite-based IP communication link, for surveillance tasks at a disaster site. The robot system consists of a large-scale outdoor robot to serve as a carrier of small robots and a fleet of small robots to be deployed at a specific spot such as an inside of a building complex. A combination of a laser range scanner and an omni-directional camera is used to acquire high resolution 3D geometry data and rendering images. Those data and images are displayed using *Mixed Reality* (MR) technology at a remote site to provide an overall picture for operation managers with high fidelity. This paper presents our initial experiments using a robot test bed with an emphasis on remote operation over rough terrain and for acquisition of high resolution 3D geometry data and telepresence using MR technology.

Keywords: Disaster Mitigation, Surveillance Robots, Wireless and Satellite-Based IP Communication, Ad-hoc Networking, Telepresence and Teleoperation, Omni-directional Camera, Laser Range Scanner, Mixed Reality

1 Introduction

Development of robotic systems for search and rescue operations receives increasing attention and national priority after the Hanshin-Awaji earthquake in 1995, Japan [1] and the World Trade Center incident in 2001, U. S. A. [2] In case of such natural or man-made disasters, it is necessary to grasp a whole

picture of the extent and degree of the damages and victims as quick as possible. But when the extent and degree becomes grater it becomes more and more difficult to do immediate surveillance and rescue operations, because the access of the human teams becomes difficult and the communication networks go disorder due to physical damages on the ground facilities and the rush of access from a general public. To the robotics community, the development of remotely operated robots for immediate surveillance and possible rescue operations is strongly expected to mitigate the disaster by saving the lives of victims and avoiding a secondary disaster on the human rescue teams.

Since 2003, a group of present authors have been working on a newly initiated project of networked robotic system for disaster mitigation, under the support from the Japanese Ministry of Internal Affairs and Communications (MIC). The project aims at the development of a robotic system for surveillance of a remote disaster site. However the focus is not limited to the development of a single robot, but covers more *Information Technology* oriented subjects and integration of those robotics and information technologies. Three key issues of our project are summarized as follows:

1. Development of a network configuration and congestion control technologies to secure the emergency communication by making maximum use of Internet and wireless ad-hoc networks in case of wide-area disasters.
2. Development of a robotic system that can be deployed in the disaster site and teleoperatively or autonomously do surveillance tasks by cooperating among multiple robotic agents.
3. Development of a *Mixed Reality* technology to effectively display the high resolution 3D geometry data and images of the disaster site, acquired by the robotic agents, to the operation managers at a remote site with high fidelity.

Finally, the project looks at a possibility to demonstrate the integrated technology by using a satellite-based IP communication link that will be provided by ETS-VIII, a Japanese Engineering Test Satellite for advanced telecommunication technologies. Currently, satellite-based communication has a disadvantage of lower transmission bandwidth, however it has a grater advantage that the satellites are not damaged by the disasters on the ground.

This paper presents our initial experiments using a robot test bed with an emphasis on remote operation over rough terrain for acquisition of high resolution 3D geometry data and telepresence using MR technology.

2 System Concept and Mission Scenario

We develop a robot system consists of a large-scale outdoor robot (hereafter termed as a “parent robot”) to serve as a carrier of a fleet of small robots (hereafter termed as “children robots”) to be deployed at a specific spot such as an inside of a building complex. Fig. 1 depicts an artist’s impression of



Fig. 1. Artist's impression of the parent robot

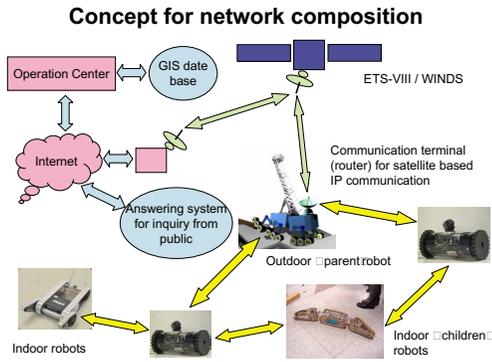


Fig. 2. A concept of the networking for the proposed robotic surveillance system

the parent robot. The parent robot should provide rough terrain mobility to approach a collapsed building, then using a ladder lift up children robots and deploy them on the higher floors of the building. After the deployment of the children robots, the parent robot could serve as a router for the wireless network of children robots and a bridge to the satellite-based communication link. Some of the children will go deep inside of the complex where the wireless signals from the parent cannot reach. In such a case a communication link should be established by relaying through a chain of children robots. Fig. 2 depicts our concept of IP based operation network that connects multiple surveillance robots at the depth of a disaster site and an operation center that could be located far from the site, via local area wireless transmission, satellite-based wireless transmission, and the Internet in the undamaged area.

The mission of the robot system is to acquire 3D geometry data and photo images of the site. For this purpose a combination of a laser range scanner and an omni-directional camera will be mounted on each robot. The geometry data and images are displayed using *Mixed Reality* technology at the operation center to provide an overall picture of the disaster site for operation managers with high fidelity. Also, other sensors for detecting victims such as infrared and CO₂ sensors should be mounted on the robots.

3 Mobile Robot Test Bed

For the initial development and tests of the technologies, a four-wheel mobile test bed was designed and developed as depicted in Fig. 3. The test bed weighs about 30 kg including an on-board computer, electronics and batteries. Each wheel has an independent motor drive (with Maxson 22W DC motors) and a steering control. On each side the front and rear wheels are connected by a mechanical link that looks like a leg, and the left and right links are

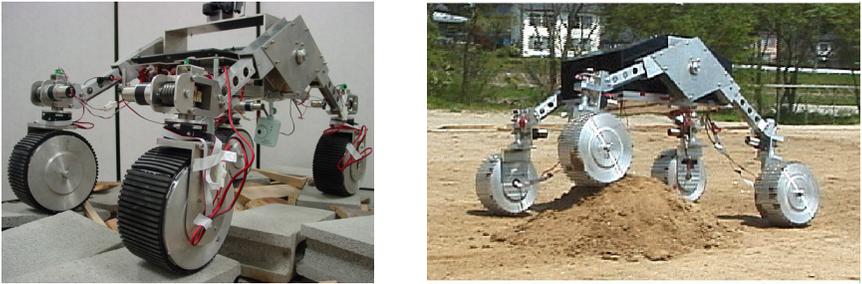


Fig. 3. A four-wheel mobile test bed for initial experiments

differentially connected at the central main body. This differential suspension system is called “rocker” suspension [3] and shows highly adaptive capability in traveling over rough terrain.

Fig. 4 depicts a block diagram of the on-board control system. As for the motor controllers and power drivers, we used common and commercially available products as much as possible. Particularly, for the interface with an on-board computer which is a standard laptop PC, we use USB. Through the USB hub, we can add more motors and sensors onto the system easily.

Wireless ethernet connection (IEEE802.11b), the modem of which is built in the on-board laptop computer, is used for remote operation of the robot. The traveling velocity and steering angle commands are given by a remote operator using a joystick and transmitted to the robot, then the local feedback control is performed by USB interfaced motor controllers (iMCs01, iXs Research Corp, with Hitachi H8 micro processor at 20MHz clock) on the robot to follow the given commands. The views of navigation cameras (with VGA

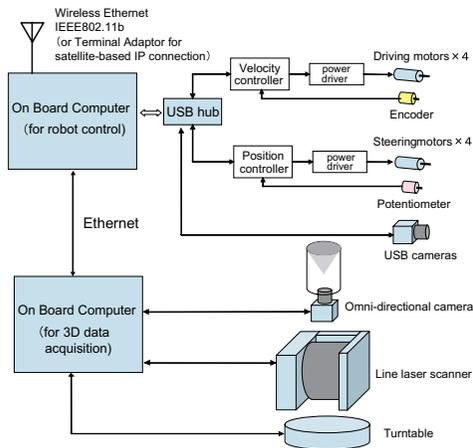


Fig. 4. Block diagram of the on-board control system

quality, Motion JPG format at 6fps) are transmitted back to the operator for hazard detection.

For the acquisition of geometry information and picture images, a combined sensor system with a line laser scanner and an omni-directional camera is mounted on the central body of the robot. The line laser scanner turns stepwise by the turntable controller. The details of the telepresence technology are elaborated in the following section.

For the hazard detection during the rover locomotion, a stream of visual images around the front wheels is strongly necessary. But the quality of the images is not necessarily so high. On the other hand, for the construction of a map around the environment, the resolution of 3D measurements should be as high as possible. However, we do not need to transmit a high-quality video stream, but three dimensional mesh data and still images taken at selective locations. This strategy eases the requirement for the transmission bandwidth. Eventually, in our test bed experiments, the overall bandwidth of the wireless connection between the robot and an operator console is just within 11Mbps, including the video stream from the navigation (hazard) cameras.

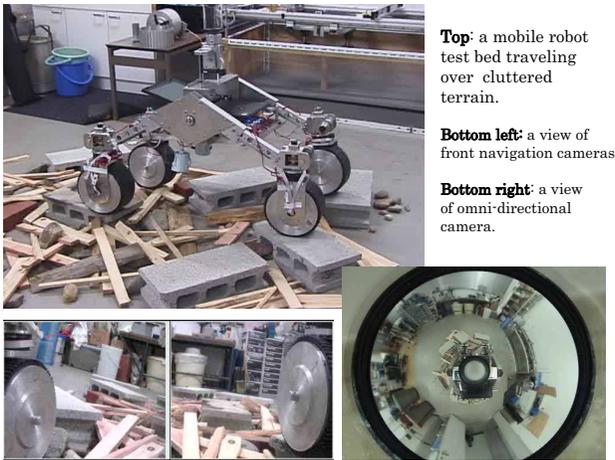


Fig. 5. A snapshot of the experiments of teleoperation

The experiments of teleoperation were carried out successfully with the above apparatus. Fig. 5 depicts a snapshot of the indoor experiment traveling over a cluttered floor.

Note that the robot test bed used in the experiments here does not represent the parent robot or children robots in terms of the size, mechanical design, or specific mobility performance. The parent robot in our mission scenario should be much bigger and tougher for outdoor operation, and the children robot can be much smaller for the investigation in a narrow space.

But in terms of wireless teleoperation and data acquisition, core technologies are common in any mobile robots with different scales or designs.

4 Telepresence Using Mixed Reality Technologies

4.1 High Resolution 3D Geometry Acquisition of a Remote Environment

As a preparation of reproducing a remote environment using Mixed Reality technologies, our high resolution 3D geometry acquisition system of a real scene is described in this subsection. Instead of using a pair of stereo cameras which has a disadvantage of inaccurate depth measurement for distant targets, an acquisition method using a laser range scanner is employed with a co-axis omni-directional camera on a turntable. Although a number of laser scanning systems have already been commercialized, our method is advantageous over them in terms of drastic cost reduction and measurement flexibility.



Fig. 6. A line laser scanner (SICK, LMS291) mounted on a turntable

At a measurement site, the depth information of the real scene is measured by a line laser range scanner (SICK, LMS291) on a turntable (Chuo Precision Industrial, QT-CM2 and ARS-136-HP) and sent to an operation center together with omni-directional images. At the operation center, a 3D polygon model (mesh model) is reconstructed from the depth information (point cloud) and images (texture data) received, and projected on an immersive display (Matsushita Electric Works, CyberDome, approx. 140 degrees of horizontal viewing angle). The line laser range scanner measures depth information along a line. The 3D reconstruction is realized by rotating the measurement line and by integrating a series of depth information. Fig. 6 shows the line laser range scanner on the turntable. Fig. 7 and Fig. 8 show examples of a measurement site and a reconstructed 3D polygon model, respectively. In this case, 901 lines were measured in about two minutes by rotating the turntable



Fig. 7. A scene of a remote site

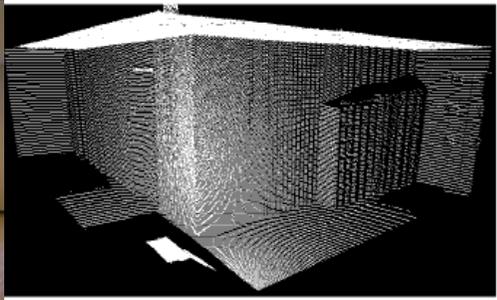


Fig. 8. A reconstructed 3D polygon model of the remote site

with an interval of 0.1 degree, each having 361 measurement points scanned by the line laser range scanner with an interval of 0.5 degree.

In order to acquire the texture information of a real scene, if a symmetric omni-directional camera is used, a single shot is enough to obtain the image around 360 degrees. But if using a non-omni camera or an asymmetric omni-directional camera whose images has the highest resolution in the directions of 0 and 180 degrees [4], it is effective to rotate the camera on the same turntable of the laser scanner.

4.2 Telepresence Technique Combining Model-Based and Image-Based Approaches

The requirement for the surveillance of a disaster site is twofold. One is image based surveillance in which visual features such as smoke and fire or some characteristic colors are important. However, this kind of surveillance does not provide fine 3D geometry information. The other is the acquisition of relatively high resolution of 3D geometry data of the environment. This kind of information is particularly useful for the localization and navigation of the robot. In case exploring an unknown environment, map building of the environment is a priority task. The former can be termed “image-based” approach and the latter “model-based.” A comparison of these two approaches is summarized in Table 1.

Table 1. A comparison of telepresence techniques

| | image-based | model-based | | proposed method |
|------------------------|-------------|------------------|--------------------|------------------------|
| type of data | images | fine 3D geometry | coarse 3D geometry | images and 3D geometry |
| photorealistic | high | high | low | high |
| real-time construction | possible | difficult | possible | possible |
| movement of viewpoint | difficut | possible | possible | possible |

In this research, we develop a telepresence technique that satisfies the requirements of both image-based and model-based approaches. Based on the assumption that the environment does not change so quickly, we construct the geometry model and make telepresence with color (texture) information obtained from visual images. Our technique is advantageous over both model-based and image-based telepresence approaches.

In the proposed technique, the color (texture) information of a patched area obtained from a still-camera image is allocated onto the corresponding 3D polygon model. Since the calculation of a huge amount of intersecting points by a CPU is too time-consuming to perform in real-time, a multi-pass rendering algorithm has been newly developed. The algorithm achieves a two-step projection for texture mapping by high-speed GPUs [5]. Fig. 9 depicts a texture-mapped version of the wire-framed model that was shown as Fig. 8.



Fig. 9. A texture-mapped representation of the 3D geometry model

By using this technique, a telepresence system has been realized, in which the latest image is updated in real-time while fine texture is gradually mapped on the entire model according to the camera movement in the environment. In the telepresence display, a human operator can interactively choose an arbitrary viewpoint of data representation. An indoor high-precision 3D tracker (3rdTech, HiBall-3100) was employed for sensing the position and orientation of human operator's head, so that the display can synchronize with the operator's view point and line of sight.

Fig. 10 depicts a snap shot of the developed MR-based telepresence system. The scenes around the disaster site are displayed (right) from an arbitrary viewpoint with an arbitrary view angle according to the head motion of the operator (left). Fig. 11 depicts a concept of an MR-based decision making room where multiple operation managers can share the whole picture of the

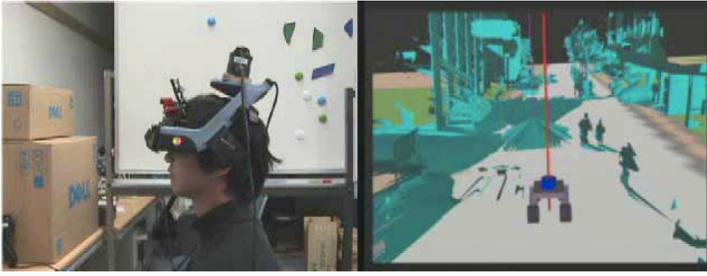


Fig. 10. A snap shot of the developed MR-based telepresence display (right) that is interactive with the operator's head motion (left)

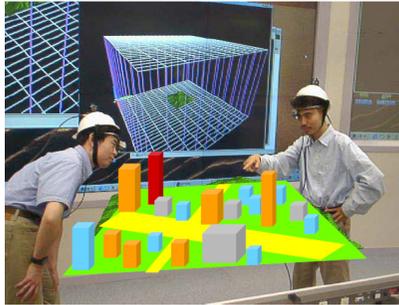


Fig. 11. A concept of an MR-based decision making room for operation managers

disaster cite with their individual view points through the individual head-mount displays.

For the incremental 3D geometry map construction along with the rover's motion, odometry information was used in the initial indoor experiments with the robot test bed, but odometry becomes unreliable when traveling over rough terrain. Alternative methods for the estimation of camera positions and orientations in practical situations need to be developed. SLAM (Simultaneous Localization and Mapping) technologies, around which there are an increasing number of papers and tutorials recently [6], are strong candidate for this purpose. We are currently looking into the practical implementation aspect of the SLAM algorithms in terms of less computational complexity and robustness in a real world.

5 Remote Navigation of the Robot

By integrating the developed technologies of teleoperation and telepresence, a "Stop and Go" type navigation is possible for practical surveillance tasks.

1. Before starting, the 3D geometry data should be obtained around the robot. A couple of minutes later, texture-mapped mixed-reality (MR) images

of the environment will be displayed to a remote robot operator. Then the operator decides where to go.

2. During the robot motion, geometry measurement is not necessary, but the images of the navigation (hazard) camera should be transmitted at maximum available bit rate (but VGA quality at 6fps is enough). The images from the omni-directional camera are also useful, but the transmission frequency does not need as high as the navigation camera.
3. The MR display should show the model-based images with the moving viewpoint according to the robot movement. Simultaneously the real-time images obtained by the navigation camera should be superimposed for the operator to recognize immediate hazard.
4. The robot can continuously travel until the boundary of the model constructed by the recent previous measurement. The effective range of the 3D geometry acquisition is about 10 m in our test bed. Then, stop and obtain the 3D range data after every 5-10 m of locomotion for the incremental map building of the environment.
Go back to step 1 and repeat the navigation.

6 Conclusions

In this paper, a newly initiated project of networked robotic system for disaster mitigation is introduced. The key concepts of the project are (1) utilization of the Internet and ad-hoc wireless networks for emergency communication, (2) coordination of multiple robots for outdoor and indoor surveillance tasks, and (3) the mixed reality representation of the disaster environment to the operation managers by combining the image-based and model-based techniques. Finally, the project looks at a possibility to demonstrate the integrated technology by using a satellite-based IP communication link, which has an advantage that the satellites are not damaged by the disasters on the ground. In this paper, the focus was made on the development of key technologies for the topics (2) and (3).

A robot test bed was developed as a general and common research platform that has a standard laptop PC with wireless ethernet communication interface as an on-board controller. Operation of the robot is relatively simple. Just give the traveling velocity and steering angle commands by a joystick. For immediate hazard detection, navigation cameras are mounted on the robot gazing around the wheels. The camera images are transmitted within the bandwidth of 11 Mbps. In addition, the robot carries a laser range scanner and an omni-directional camera mounted on a turntable, in order to acquire high resolution 3D geometry data and rendering images of the environment around the robot. Those data and images are displayed using *Mixed Reality* technology at a remote site to provide an overall picture for operation managers with high fidelity.

A key technology for telepresence display was developed in combining the advantages of image-based “reality” and model-based “substance.” In the proposed technique, the color texture information of the camera images is allocated onto the corresponding 3D polygon model in real-time. This technique allows us interactive display of the scenes from arbitrary viewpoints. A feasible operation scheme for the teleoperation of a remote robot with the assistance of the MR based telepresence was developed and tested by our mobile robot test bed.

References

1. “Annual Report on the Development of Advanced Robots and Information Systems for Disaster Response” C2003 (in Japanese).
2. Angela Davids, “Urban Search and Rescue Robots: From Tragedy to Technology”, IEEE Intelligent Systems, pp.81–83, March/April 2002.
3. http://marsrovers.jpl.nasa.gov/mission/spacecraft_rover_wheels.html
4. Kazuaki Kondo, Yasushi Yagi, Masahiko Yachida, “Asymmetric Omnidirectional Vision for Robot Navigation,” Meeting on Image Recognition and Understanding, July 2004. (in Japanese)
5. Tomoaki Adachi, Takefumi Ogawa, Kiyoshi Kiyokawa, Haruo Takemura, “A Telepresence System by Using Live Video Projection of Wearable Camera onto a 3D Scene Model”, to be presented at International Conference on Computational Intelligence and Multimedia Applications, Las Vegas, August 2005.
6. For example: Summerschool on “Simultaneous Localisation and Mapping,” <http://www.cas.kth.se/SLAM/>, Stockholm, Sweden, August 2002.

Towards Intelligent Miniature Flying Robots

Samir Bouabdallah¹ and Roland Siegwart²

¹ Autonomous Systems Lab, EPFL samir.bouabdallah@epfl.ch

² Autonomous Systems Lab, EPFL roland.siegwart@epfl.ch

Summary. This paper presents a practical method for small-scale VTOL³ design. It helps for elements selection and dimensioning. We apply the latter to design a fully autonomous quadrotor with numerous innovations in design methodology, steering and propulsion achieving 100% thrust margin for 30 min autonomy. The robot is capable of rotational and translational motion estimation. Finally, we derive a non-linear dynamics simulation model, perform a simulation with a PD test controller and test successfully the robot in a real flight. We are confident that "OS4" is a significant progress towards intelligent miniature quadrotors.

Keywords: VTOL design, quadrotor, quadrotor modelling

1 Introduction

Research activities in rolling robots represent the lion's part in mobile robotics field. In the case of complex or cluttered environments the miniature flying robots emphasis all their advantages. The potential capabilities of these systems and the challenges behind are attracting the scientific community [1], [2], [3], [4]. Surveillance, search and rescue in hazardous cluttered environments are the most important applications. Thus, vertical, stationary and slow flight capabilities seem to be unavoidable making the rotorcraft dynamic behavior a significant pro. In cluttered environments the electrical propulsion, the compactness, the hard safety and control requirements, the abandon of GPS are not only a choices, they are imposed. Most of the early developments suffer from a lack of intelligence, sensory capability and short autonomy except for the larger machines. In this paper we present the new design of a fully autonomous quadrotor helicopter named "OS4", equipped with a set of sensors, controllers, actuators and energy storage devices enabling various scientific experiments. This robot was built following a design methodology adapted for miniature VTOL systems.

³ Vertical Take-Off and Landing

2 Design

The interdependency of all the components during the design phase makes the choice of each one strongly conditioned by the choice of all the others and vice-versa.

2.1 Design Methodology

The open-loop simulation analysis [5] have shown clearly the strong dynamic instability of a quadrotor. However, one can improve the stability by simply acting on several system parameters. For instance, spreading the mass in each of the four propulsion groups⁴ (PG) increases the diagonal elements of the inertia matrix. Moreover, building the quadrotor in a regular cross configuration simplifies the control law formulation [6]. One can also optimize the vertical distance between the CoG and the propellers center in order to increase the damping (CoG below propellers), or slow the natural frequency [7] (CoG above propellers). On the other hand, augmenting the horizontal distance (CoG-propellers) increases the inertia. Taking a decision concerning all these design variables requires to follow an appropriate methodology. This paper proposes a practical method to handle the design problematic of a small scale rotorcraft by combining the theoretical knowledge of the system and a minimum of optimization results analysis. This method is by far less complex than a traditional MDO⁵.

The General Method

The starting point of the design process is to define an approximate target size and weight of the system, dictated generally by the final application. This gives a good idea about the propeller size to use. Using an analytical model of a propeller with for instance blade element theory or by an experimental characterization of a given propeller [8] one can estimate the thrust and drag coefficients which permits the verification of the thrust requirements. For the special case of the quadrotor a rule of thumb fixes an optimum thrust to weight ratio to 2:1⁶. This was observed during several simulations and experienced with the limited actuators of the first "OS4" prototype [8]. The propeller's information helps to build a selected actuators data bank which are likely to meet the power requirements. Then, a rough estimation of the airframe and avionics masses is necessary (see Fig. 6) to have a first estimation of the total mass without battery. The latter is found by an iterative algorithm as schematized in Fig. 1.

⁴ Propeller+Gearbox+Motor

⁵ Multidisciplinary Design Optimization

⁶ 1.4:1 for a miniature coaxial and 4:1 for small scale aerobatic helicopters

The Iterative Algorithm

The process starts by picking-up an actuator from the data bank, estimating its performances with the propeller's model, computing the system total mass, power consumption, propulsion group cost and quality factors in the equilibrium and maximum thrust points. Moreover, the autonomy and a special index (autonomy/mean power) characterize the overall system quality. This is done for an incremental battery mass variable, for every actuator in the data bank as schematized in Fig. 1.

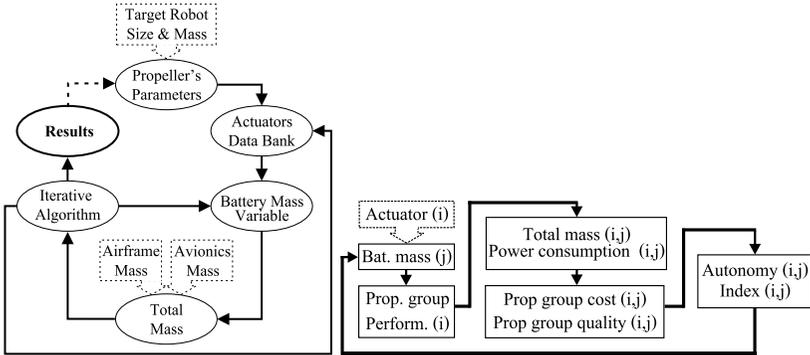


Fig. 1. Left: The design method flowchart. Right: The iterative algorithm flowchart.

2.2 "OS4" Quadrotor Design

The "OS4" quadrotor developed during this project represents a design example following the method described in Subsec. 2.1. The targeted system is about 500 g in mass and 800 mm in span.

The Propulsion Group Design

The "OS4" requirements lead to a 300 mm diameter propeller. The main design variables of a PG are listed in Table 1, and used in the models in Table 2. Finally, the choice of the PG components was based on the iterative algorithm classification with an average cost factor of $C = 0.13 W/g$ and a quality factor of about $Q = 5 g/W$. This was for a given Lithium-polymer battery mass of $m_{bat} = 230 g$, (11V, 3.3Ah) and an autonomy estimation of 30 minutes. The choice of 2 blades propeller topology rather than more is mainly due to loss of motor efficiency and large rotor inertias with a heavier propeller. The latter is made out of carbon and was adapted to our specifications. The electrical motors torque in these application being limited, the gearbox seem to be mandatory and beneficial for such VTOL to preserve good motor efficiency.

Table 1. Propulsion group design variables

| propeller | | OS4 | unit | gearbox | | OS4 | unit | motor | | OS4 | unit |
|--------------|----------|---------|-------------------|-------------|-------------|--------|-------------------|---------------|----------|-------|-------------------|
| efficiency | η_p | 62-81 | % | efficiency | η_{gb} | 96 | % | efficiency | η_m | 50-60 | % |
| mass | m_p | 5.2 | g | mass | m_{gb} | 7 | g | mass | m_m | 12 | g |
| thrust coef. | b | 3.13e-5 | N s ² | max. torque | | 0.15 | Nm | max. power | P_{el} | 35 | W |
| drag coef. | d | 7.5e-7 | Nm s ² | max. speed | | 1000 | rad/s | internal res. | R | 0.6 | Ω |
| inertia | J_r | 6e-5 | kg.m ² | inertia | J_{gb} | 1.3e-6 | kg.m ² | inertia | J_m | 4e-7 | kg m ² |
| speed | Ω | 199-279 | rad/s | red. ratio | r | 4:1 | | torque cst. | k | 5.2 | mN m/A |

Table 2. Propulsion group component’s models. \mathbf{T}_w and \mathbf{BW} (max. control frequency) are respectively the thrust/weight ratio and the PG bandwidth (see tab:PGDesignVariables for symbols definitions)

| component | model |
|------------|---|
| Propeller | $\mathbf{b}, \mathbf{d} \times \Omega^2 = \mathbf{T}, \mathbf{D}$ |
| Gearbox | $\mathbf{P}_{in} \times \eta_{gb} = \mathbf{P}_{out}$ |
| DC motor | $-\frac{k^2}{R}\omega - \mathbf{D} + \frac{k}{R}\mathbf{u} = \mathbf{J} \frac{d\omega}{dt}$ |
| PG cost | $\mathbf{P}_{el}/(\mathbf{T} - \mathbf{m}_{pg}) = \mathbf{C}$ |
| PG quality | $\mathbf{T}_w \times \mathbf{BW}/\Omega \times \mathbf{C} = \mathbf{Q}$ |

This is linked to the fact that we prefer to use large and low speed propellers. The high power/weight ratio of the selected (12 g, 35W) BLDC motor justifies this choice even with the control electronics included. A 6 g MCU based I²C controller was specially designed for the sensorless outrunner LRK195.03 motor as shown in Fig. 2. Obviously, BLDC motors offer high life-time and less electromagnetic noise. The ready to plug PG weights 40 g and lifts more than 260 g.

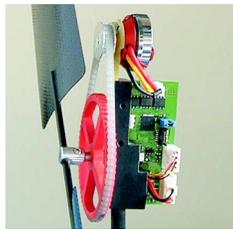


Fig. 2. The "OS4" propulsion group.

The Avionics

The limited payload imposes some restrictions on the sensors. For yaw angle and linear displacements measuring on "OS4" we use a lightweight vision based sensor. Fig. 3 represents the block diagram of the "OS4" avionics.

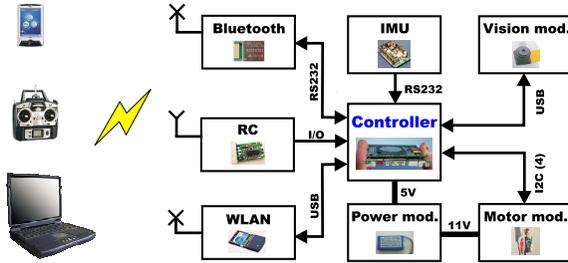


Fig. 3. OS4 block diagram.

The Inertial Measurements Unit

The "OS4" quadrotor uses the MT9-B, a 15 g (OEM) commercially available IMU to get absolute roll and pitch angles and their corresponding angular velocities at up to 512 Hz. The IMU is installed horizontally at 45 deg from the carbon rods. In this configuration the robot flies forward following the IMU *x* axis. This original quadrotor steering makes it possible to reduce the lift dissymmetry effect as showed in Fig. 4.

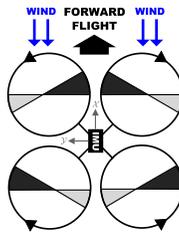


Fig. 4. Reducing the lift dissymmetry effect. Black region:High lift, Grey region:Low lift.

The Vision Module

The GPS signal weakness and precision in cluttered environments makes it difficult to use. On the other hand, the surrounding metallic structures strongly disturb the IMU magnetic based yaw estimation. Thus, it was necessary to

develop a lightweight visual positioning module, assuming a flat floor with chessboard structure. The system uses a 0.6 g micro-camera (OV7648) to extract and track the chessboard corners and the roll and pitch information to correct the motion estimation. It is presently possible to provide the relative altitude, the yaw angle, the linear horizontal displacements and their corresponding time derivatives at up to 15 Hz. The precision is of the order of the tenth of degree for the yaw, millimeter for the altitude and centimeter for the horizontal displacements. Obviously, the error grows with the displacement speed while the sensor is valid for roll and pitch angles of ± 20 deg. Considering chessboard squares of 40 mm side, the altitude measurement range is 0.5 m to 3 m. It was thus necessary to add a laser diode and to extract its spot position in the image estimating the altitude for the take-off and landing procedures. The actual module is a preliminary approach. The final goal is to achieve a visual odometry without modifying the environment.

The Controller

Embedding the controller for our application is definitely advisable as it avoids all the delays and the discontinuities in wireless connections. A miniature computer module (CM), based on Geode 1200 processor running at 266 Mhz with 128 Mo of RAM and as much of flash memory was developed. The computer module is x86 compatible and offers all standard PC interfaces in addition to an I²C bus port. The whole computer is 44g in mass, 56 mm by 71 mm in size (see Fig. 5) and runs a Debian based minimalist Linux distribution.



Fig. 5. The x-board based, 40 g and 56x71 mm computer module.

The Communication Modules

The controller described in the paragraph above includes an MCU for Bluetooth chip interfacing with the computer module. The same MCU is used to decode the PPM⁷ signal picked-up from a 1.6 g, 5 channels commercially available RC receiver. This makes it possible to change the number of channels as convenient and control the robot using a standard remote control. Finally, a wireless LAN USB adapter was added. On the ground side, a standard GCS⁸

⁷ Pulse Position Modulation

⁸ Ground Control Software

for all our flying robots is developed. Presently, it permits UAV environment visualization, waypoints and flight plans management as well as data logging and controller parameters tuning.

The Design Results

The robot as a whole represents the result of the design methodology and fits the requirements. One can see mass and power distributions from Fig. 6. The total mass is about 520g where the battery takes almost the one-half and the actuators only the one-third thanks to BLDC technology. All the actuators take obviously the lion's part, 60 W of 66 W the total power consumption. However, the latter depends on flight conditions and represents a weighted average value between the equilibrium (40 W) and the worst possible inclination state (120 W) without losing altitude. Fig. 7 shows the real robot.

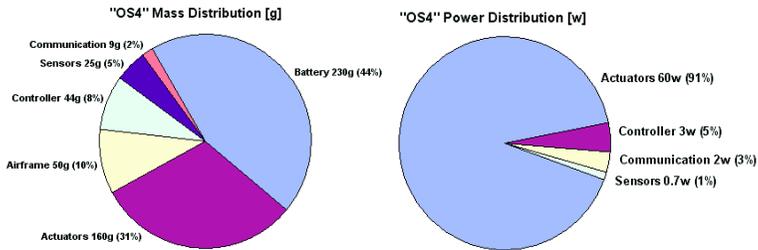


Fig. 6. Mass and power distributions in "OS4" robot.



Fig. 7. The "OS4" quadrotor.

3 Modelling

Modelling a helicopter is a quite complex task and one has to make some simplifying. In this case, the airframe is rigid, all the propellers are in the same horizontal plan and the quadrotor structure is symmetric. Obviously, only the dominant effects are modelled. The dynamics of a rigid body under external forces applied to the center of mass and expressed in the body fixed frame as shown in [9] are in Newton-Euler formalism:

$$\begin{bmatrix} mI_{3 \times 3} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mV \\ \omega \times I\omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix} \tag{1}$$

Where $I \in \mathfrak{R}^{(3 \times 3)}$ the inertia matrix, V the body linear speed vector and ω the body angular speed. Let's consider U_1, U_2, U_3, U_4 as the system inputs and Ω as a disturbance:

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = b(-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_3 = b(-\Omega_1^2 + \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\ U_4 = d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\ \Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{cases} \tag{2}$$

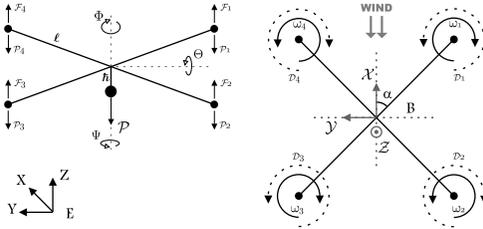


Fig. 8. The "OS4" coordinate system.

3.1 Moments Acting on a Quadrotor

Actuators Action

Several combinations of propellers actions are possible for rolling or pitching a quadrotor. Following the coordinate system on Fig. 8, one can write:

$$\tau_\alpha = \begin{pmatrix} l \cos \alpha U_2 \\ l \cos \alpha U_3 \\ U_4 \end{pmatrix} \tag{3}$$

The first two elements of (3) include the $\Delta T = \Sigma T_i$ and the third one the $\Delta D = \Sigma D_i$ aerodynamic effect listed in Table 2.

Rotors Gyroscopic Effect

One of the most important sources of instability in a quadrotor. One can attenuate it by reducing the propellers rotational speed or inertia. The dumping also increases by lowering the CoG. Otherwise, one can constrain the control to keep it compensated between each pair of propellers.

$$\tau_p = \begin{pmatrix} J_r \dot{\theta} \Omega \\ J_r \dot{\phi} \Omega \\ 0 \end{pmatrix} \quad (4)$$

Rotors Inertial Counter Torque

These terms result from the reaction torque produced by a change in rotational speed [10].

$$\tau_i = \begin{pmatrix} 0 \\ 0 \\ J_r \dot{\Omega} \end{pmatrix} \quad (5)$$

Horizontal Motion Friction

The friction force on the propellers resulting from horizontal linear motion induces moments on the helicopter body. The $F_{x,y}$ forces depend on V and Ω_i and must be estimated.

$$\tau_f = \begin{pmatrix} F_x h \\ F_y h \\ 0 \end{pmatrix} \quad (6)$$

The moments due to propeller lift dissymmetry are neglected thanks to "OS4" construction (see, Fig. 4). From (1) – (6) one can rewrite the quadrotor rotational dynamics:

$$\begin{bmatrix} I_{xx} \ddot{\phi} \\ I_{yy} \ddot{\theta} \\ I_{zz} \ddot{\psi} \end{bmatrix} = \omega \times I \omega + \tau_p + \tau_a + \tau_i - \tau_f \quad (7)$$

3.2 Forces Acting on a Quadrotor

Actuators Action

The quadrotor is an underactuated system hence it's horizontal motion is mainly due to the orientation of the total thrust vector (using the rotation matrix).

$$F_a = \begin{pmatrix} \cos \phi \sin \theta \cos \psi U_1 + \sin \phi \sin \psi U_1 \\ \cos \phi \sin \theta \sin \psi U_1 - \sin \phi \cos \psi U_1 \\ -mg + \cos \phi \cos \theta U_1 \end{pmatrix} \quad (8)$$

Horizontal Motion Friction

The friction force on vehicle's body during horizontal motion is:

$$F_f = -C_{x,y,z} V^2 \quad (9)$$

From (1), (2), (8) and (9) one can rewrite the quadrotor translational dynamics:

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \end{bmatrix} = \omega \times mV + F_a - F_f \quad (10)$$

3.3 "OS4" Model Parameters

Table 3 lists most of "OS4" model parameters. The inertia matrix is supposed diagonal thanks to the symmetric construction. The CAD software gives the exact inertia values. The remaining aerodynamic parameters will be identified in near future.

Table 3. "OS4" Main Model Parameters.

| parameter | value | unit | |
|------------------------|----------|----------|-------------------|
| thrust coef. | b | 3.13e-5 | N s ² |
| drag coef. | d | 7.5e-7 | Nm s ² |
| inertial moment on x | I_{xx} | 6.228e-3 | kg m ² |
| inertial moment on y | I_{yy} | 6.225e-3 | kg m ² |
| inertial moment on z | I_{zz} | 1.121e-2 | kg m ² |
| arm length | l | 0.232 | m |
| CoG to rot. plane | h | 2.56e-2 | m |
| robot mass | m | 0.52 | kg |
| propeller inertia | J_r | 6e-5 | kg m ² |

4 Simulation

Several simulations were performed under Matlab using the model parameters listed in Table 3 with a simple PD controller (Roll and Pitch: Kp=1, Td=0.6. Yaw: Kp=0.4, Td=0.3). The task was to stabilize the helicopter attitude to ($\phi = \theta = \psi = 0$), from ($\phi = \theta = \psi = \pi/4$) initial conditions. The simulated performance was satisfactory as showed in Fig. 9.

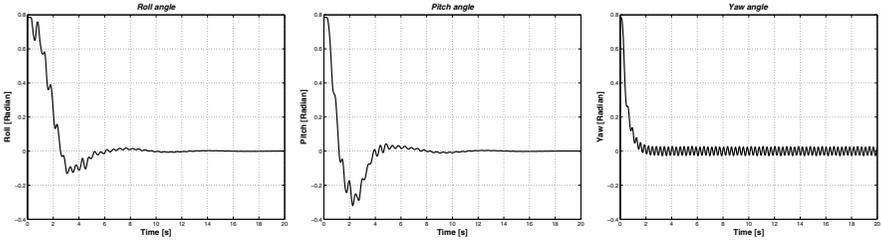


Fig. 9. Simulation: The PD controller has to stabilize the attitude.

5 Experiment

We tested successfully a real flight experiment using only the IMU sensor for attitude control (Roll and Pitch: $K_p=0.8$, $T_d=0.3$. Yaw: $K_p=0.08$, $T_d=0.03$). The robot exhibits the predicted thrust. However, the motor module bandwidth seem to be slow, this is partly responsible for the oscillations in Fig. 10. A new version of the motor module is under development. The experimental results are considered satisfactory as they practically validate part of the system in real operation.

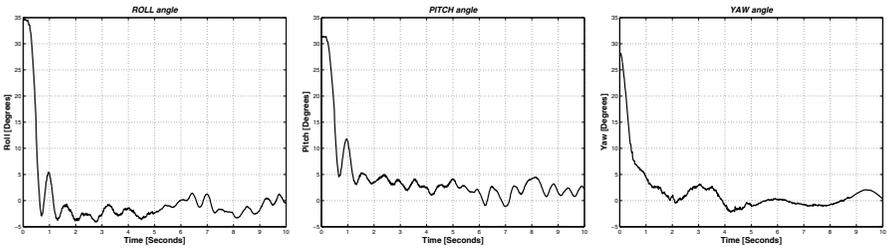


Fig. 10. Experiment: The first test flight with a PD controller. The stabilization is satisfactory.

6 Conclusion

This paper presented a practical method for miniature rotorcraft design. It was the only tool used to get the satisfied design requirements and achieve the excellent 100% thrust margin for 30 min autonomy. Our quadrotor embeds all the necessary avionics and energy devices for a fully autonomous flight. We derived the nonlinear dynamic model with accurate parameters, performed a simulation and successfully realized a test flight. The future goal is the

implementation of the control strategies developed for the first prototype at the beginning of the "OS4" project. Most parts of this development are for indoor as well as outdoor environments with minor adaptations. The numerous innovations and design results presented in this paper reinforce our conviction in the emergence of miniature intelligent flying platforms.

Acknowledgement

The authors would like to thank André Noth for fruitful discussions about flying robots, André Guignard for the mechanical parts realization, Peter Bruehlmeier for PCB design and all the students who worked or are working on the project.

References

1. Floreano D, Zufferey J.C. and Nicoud J.D. (2005) Artificial Life Winter-Spring 2005:121–138
2. Pounds P, Mahony R, Gresham J, Corke P, Roberts J (2004) Towards Dynamically-Favourable Quad-Rotor Aerial Robots. Australasian Conference on Robotics and Automation, Canberra, Australia
3. Ruffier F, Franceschini N (2004) Visually Guided Micro-Aerial Vehicle: Automatic Take Off, Terrain Following, Landing and Wind Reaction. IEEE International Conference on Robotics and Automation, New Orleans, USA
4. Kroo I, Prinz F, Shantz M, Kunz P, Fay G, Cheng S, Fabian T, Partridge C (2000) The Mesicopter: A Miniature Rotorcraft Concept Phase II Interim Report. Stanford University, USA
5. Bouabdallah S, Siegwart R (2005) Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor. IEEE International Conference on Robotics and Automation, Barcelona, Spain
6. Bouabdallah S, Murrieri P, Siegwart R (2004) Design and Control of an Indoor Micro Quadrotor. IEEE International Conference on Robotics and Automation, New Orleans, USA
7. Prouty R.W (2002) Helicopter Performance, Stability, and Control. Krieger Publishing Company
8. Bouabdallah S, Murrieri P, and Siegwart R (2003) Autonomous Robots Journal Mars 2005
9. Sastry S (1994) A mathematical introduction to robotic manipulation. Boca Raton, FL
10. Müllhaupt P (1999) Analysis and control of underactuated mechanical nonminimum-phase systems. PhD Thesis, EPLF, Switzerland
11. Olfati-Saber R (2001) Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles. PhD Thesis, MIT, USA

Design of an Ultra-lightweight Autonomous Solar Airplane for Continuous Flight

André Noth¹, Walter Engel, and Roland Siegwart²

¹ Autonomous Systems Lab, EPFL andre.noth@epfl.ch

² Autonomous Systems Lab, EPFL roland.siegwart@epfl.ch

Summary. The Autonomous Systems Lab of EPFL³ is developing, within the framework of an ESA program, an ultra-lightweight solar autonomous model airplane called Sky-Sailor with embedded navigation and control systems. The main goal of this project is to jointly undertake research on navigation, control of the plane and also work on the design of the structure, the energy generation system. The airplane will be capable of continuous flight over days and nights, which makes it suitable for a wide range of applications.

Keywords: Autonomous UAV, solar powered airplane, sustainable flight

1 Introduction

Development of unmanned aerial vehicle (UAV) has attracted the attention of several agencies and university laboratories over the past decade, due to their great potential in military and civilian applications.

There are a dozen commercial autopilots (Micropilot, Procerus, etc.) which combine tiny dimensions, low weight and quite efficient navigation capabilities. Despite all this, they usually use limited CPU power which restricts the control of the airplane to classic control methods like separated PID loops and doesn't allow the onboard execution of more complex algorithms, for example, those of image processing.

On the other side, there is a lot of research in Universities in various fields, such as SLAM⁴, hardware design, control, navigation, trajectory planning, etc. But whether they are done on VTOL⁵ systems or fixed-wing model airplanes, the embedded system is often over-dimensioned, compared to the airplane itself, in order to have high computational capabilities and efficient sensors.

³ Ecole Polytechnique Federale de Lausanne

⁴ Simultaneous Localization and Mapping

⁵ Vertical Take-Off and Landing

Consequently, the UAV becomes very heavy, needs high electrical power and the flight endurance reduces dramatically. Thus, endurance being one of the most important parameters for the targeted applications, the development and the application are not in correlation.

In this paper, we present the airplane developed for the project Sky-Sailor whose aim is to build a solar autonomous motor glider by taking care of all aspects, not only the autopilot system but as well the mechanical structure, the solar generator, the energy storage, etc. It differs from other similar projects like Helios or Centurion by its low weight and low cost. The final airplane only weighs 2.5 kg and according to the AUVS-international is part of the *High Altitude Long Endurance* UAV category [3].

2 Airplane Overview

2.1 Mechanical Structure

The approach we chose for the design of the airplane was to combine the knowledge of aerodynamics engineers and the experience of lightweight model airplane designers. The starting point for this design was the model airplane of Walter Engel that holds the world record for flight duration of over 15 hours with 1 kg of battery. Sky-Sailor version 1 is basically a motor-glider with a structural weight of only 0.6 kg for a wingspan of 3.2 m and a wing surface of 0.776 m^2 (Fig. 1). The resulting total weight including motors, propeller, solar cells, batteries and controller is around 2.5 kg.

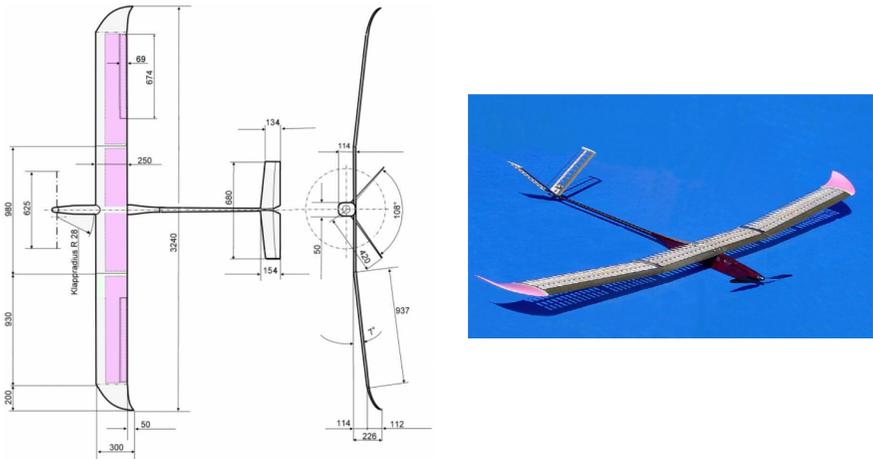


Fig. 1. Mechanical structure of Sky-Sailor

2.2 Solar generator, Battery and Propulsion System

As explained in the introduction, one major challenge is the power management that has to ensure continuous flight over days and nights.

A total of 216 silicon solar cells, divided in three modules, cover an area of around 0.512 m^2 . In terms of efficiency, the better choice would have led us to GaAs Triple Junction cells with efficiencies of 27-28 %, but taking into account the impact of the weight on the required power for levelled flight, the better choice is RWE-32 silicon cells with 16.9 % efficiency. Furthermore, the flexibility of those thin cells is also an advantage for their integration on the wing.

The cells are encapsulated using a mechanically favorable symmetrical laminate combined with a fiber glass reinforced plastic coating. This encapsulation is non-reflective. Thus, we obtain a flexible arrangement easily integrable on the plane and connectable to the power circuit. At maximum sun conditions, the available power is 28 W for each module, which makes a total of 84 W.

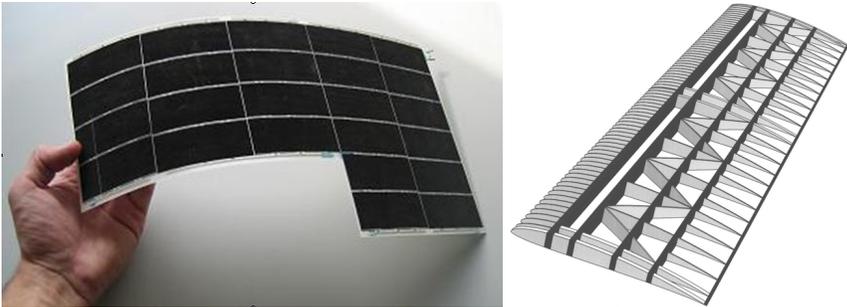


Fig. 2. Flexible solar module that can be directly integrated on the wing.

In order to get the highest amount of energy from the solar modules, a MPPT⁶ is used to charge the battery. This device is basically a high efficiency DC/DC converter with variable and adjustable gain. One of its additional function is to monitor the current and the voltage of each solar module and make those information available for the central processor through I²C.

The energy is stored in a lithium-ion polymer battery that has a nominal voltage of 28,8 V and a capacity of 7200 mAh. The propulsion group is composed of a Maxon DC motor, a gearbox and a carbon fiber propeller. The required electrical power for levelled flight of Sky-Sailor is around 16 W.

⁶ Maximum Power Point Tracker

3 Navigation and Control System

In order to reach the goal of the project, the autopilot design phase followed those principles:

- select components not only according to criteria of precision and resolution, but as well of weight and power consumption to be suitable for the targeted application.
- use as much as possible digital output and calibrated sensor to reduce the development time and avoid additional need of A/D converter, interface microcontroller, etc.
- interface the sensors so that the central processor doesn't have to wait on them but can access directly and rapidly to the information on request. This applies for example to the GPS.

3.1 Computer and Interfaces

Sky-Sailor will fly autonomously using an onboard autopilot, only high level orders being given from the ground. The system is mainly based on a single board computer, the X-board <861> which is a compact embedded PC design for low power consumption.



Fig. 3. X-board <861> single board computer from Kontron

It includes a Geode SC1200 Processor, up to 128 Mbyte of DRAM and up to 128 Flash storage media on board. Despite the compact size of a business card, it offers a lot of interfaces: integrated Graphics, Ethernet, USB, RS 232, I²C, audio... The OS running on it is a reduced Linux distribution, based on Debian, that only contains the necessary features.

3.2 Sensors

In Fig. 4, one can see the power generator system and the autopilot, with all sensors and their interfaces to the X-board.

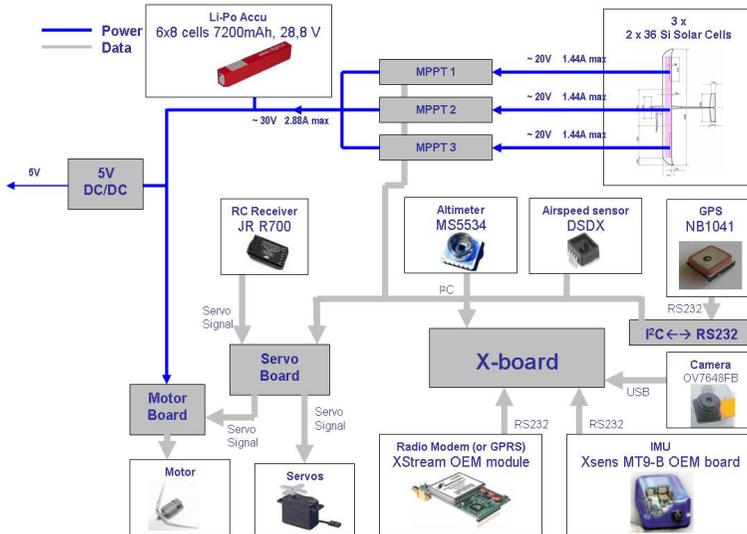


Fig. 4. Schematic view of the power and control parts of Sky-Sailor

Attitude

The attitude and angle rate of the airplane are given by the MT9-B IMU⁷ at a frequency of up to 512 Hz. Such a low-cost sensor is perfectly sufficient to perform inertial navigation compared to heavier one [6]. It contains accelerometers, magnetometers, gyroscopes and communicates through serial port (RS232) with the X-board on which data fusion is executed. In the future version of this device, the sensor fusion will be done by a DSP chip, reducing the computational cost on the central processor of the autopilot.

VGA camera

One direction of the project is to achieve autonomous navigation based on vision, using SLAM techniques as shown in [1] [2]. One or more lightweight VGA cameras will give 680 x 480 images of the landscape and allow localization and mapping of the terrain. Efforts will be done in this direction in the following month. Cameras are connected to the central processor via USB.

Absolute x-y position and altitude

The absolute position is given by an ultra low power GPS sensor with patch antenna from Nemerix. This sensor consumes only 61 mW for a weight of 12.36 gr. In terms of position accuracy, 95 % / 99.7 % of the time, the estimated position lies within 2.7933 m / 4.2028 m respectively of the actual position.

⁷ Inertial Measurement Unit

A future version will accept WAAS/EGNOS correction for more precise measurements. The data are sent on a serial port at a fixed rate of 1 Hz to a microcontroller that decodes the NMEA protocol, stores the value internally and sends them on demand to the main processor via I²C.

The same microcontroller interfaces the altitude pressure sensor MS5534. Pressure and temperature values, as well as four calibration factors allow the computation of the altitude with a resolution of 1 m. The relation between pressure and altitude being variant with the atmospheric condition, the microcontroller will achieve data fusion, using the GPS altitude as an absolute value to correct the drift of the MS5534.

Airspeed

The airspeed sensor DSDX is a differential pressure sensor, with digital I²C readout and temperature compensated. It is connected to a Pitot tube fixed at the attack border of the wing.

3.3 Ground Control Station

The control of the airplane is executed onboard but there is a link to a ground control station through a serial radio modem that allows a baudrate of 9600 bps. The goal is to:

- download and upload airplane and control parameters, but as well the flight plan, before the takeoff,
- get a visual feedback of the state of the airplane once airborne, modify flight plans on-the-fly,
- retrieve and record the telemetry for flight analysis, system identification, etc.

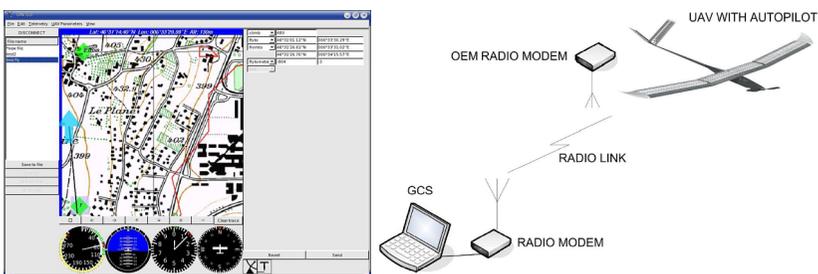


Fig. 5. Ground control station and its graphical user interface

The GUI⁸ was developed with QT graphical libraries under Linux (Fig.5). It is composed of three main layers which ensures modularity:

⁸ Graphical User Interface

- the graphical interface, that allows a visual overview of the state of the airplane and its position on a 3D map of the terrain.
- a second layer which processes data and control the GUI
- a communication module that receives and sends the data in packets to the airplane through the serial port connected to the radio modem.

Control of the airplane from the ground

As shown in Fig. 4, the commands given to the servos can come from the autopilot or a human pilot on the ground using an RC transmitter. The "servo board" decodes the PPM⁹ from the RC source and get the value given by the autopilot through the I²C bus. Based on one additional channel on the RC remote, it switches from one source to the other. It is also possible, for control tuning purpose, to mix sources and, for example, allow the autopilot to command only the elevator while the other actuators are commanded manually.

3.4 Autopilot Design Results

The final design leads us to a navigation and control system with a total mass of 140 g for a consumption of around 4 W. One can see that 6/8 of the power is used by the X-board and 1/8 for the transmission, the rest being used by the sensors.

Table 1. Autopilot power and mass distribution

| Part | Weight [g] | Power consumption [W] |
|-----------------------|------------|-----------------------|
| X-board | 22 | 3.00 |
| Mother Board | 22 | - |
| IMU | 14.5 | 0.21 |
| VGA Camera | 0.55 | 0.1 |
| GPS | 12.4 | 0.061 |
| Altitude sensor board | 2 | 0.03 |
| Airspeed sensor board | 3 | 0.03 |
| Radio-modem | 24 | 0.5 |
| Antenna | 19.6 | - |
| Cables, connectors | 20 | - |
| Total | 140 g | 3.93 W |

Globally, the autopilot represents 5% of the total mass of the airplane and uses 20% of the power.

⁹ Pulse Period Modulation

4 Simulation of the Solar Flight

For the validation of a long endurance solar flight, a simulation was realized under Matlab Simulink. Fig.6 represents the schematic of the model that includes first the irradiance model based on [12] and depending on the geographic position, time and solar panels orientation. We then take into account the surface of solar cells, their electrical efficiency and the efficiency of the connection configuration. For the MPPT, the electrical and algorithm efficiencies are taken into account. The power consumption is the addition of the autopilot power and the power needed for flight, which was measured in the case of levelled flight and climbing phase. Depending on the irradiance conditions and the consumption, the battery is charged or discharged, taking into account the efficiency of the energy transfer.

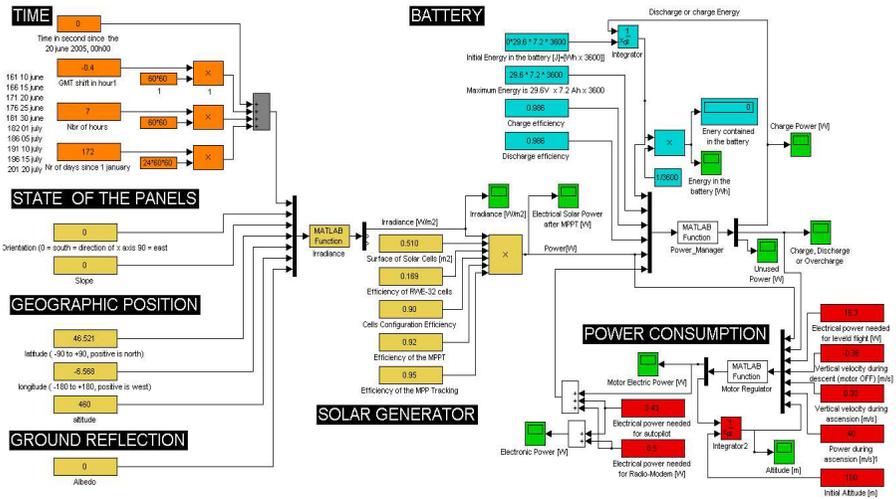


Fig. 6. Schematic of the simulation model under Matlab Simulink

4.1 Study of Various Scenarios

The simulation environment allows to test different flight strategies in order to accomplish a long endurance flight and analyze the benefit of a climbing phase or the influence of the other parameters on the feasibility of a multi-days flight. We will present here two scenarios.

In the first simulation, Sky-Sailor starts a flight at EPFL location on the 21th of June with an empty battery, keeping always the same altitude. The two graphs below show the evolution of the power distribution during 48 hours.

With good sun conditions, the battery is fully charged at 13h30. At this moment, the MPPT measures that the battery voltage reaches the maximum

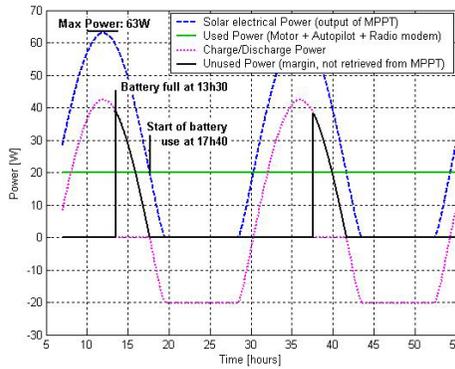


Fig. 7. Power distribution on Sky-Sailor during levelled flight

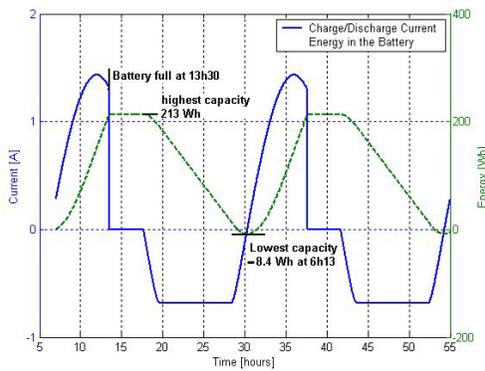


Fig. 8. Battery charge/discharge current and energy during levelled flight

voltage of 33.7 [V] and adapts the maximum power point to avoid overcharge. In this phase, the total amount of energy that is not used but that could be retrieved from the solar panels reaches 92.5 [Wh]. During the night, the battery supplies the all airplane but at 5h10 it is totally discharged.

Another strategy is to better use the energy after the battery charge by increasing altitude. Fig. 9, 10 and 11 show the same scenario presented before but with a climbing phase until 2000 [m].

Basically, Sky-Sailor uses the additional energy to gain altitude at 0.3 [m/s] using an electrical power of 40 [W]. Having reached 2000 [m], it stays at this altitude until the energy is not sufficient anymore for levelled flight. At this point, the motor is turned off and the descent starts. Finally, at the most critical point at 6h13 in the next morning, the battery still has a capacity of 4.7 [Wh] and the charging process starts again. Globally, the unused energy during the day is 61.5 [Wh].

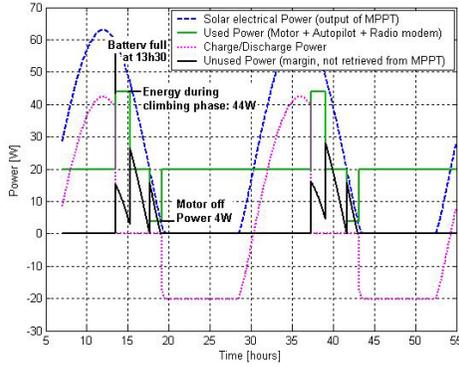


Fig. 9. Power distribution on Sky-Sailor during flight with climbing phase

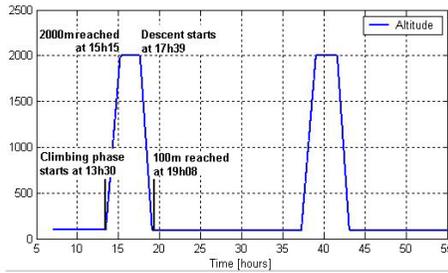


Fig. 10. Altitude during flight with climbing phase

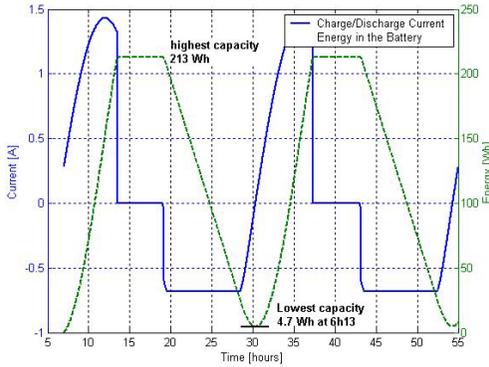


Fig. 11. Battery charge/discharge current and energy during flight with climbing phase

5 Status of the Project and Future Work

The mechanical structure of the airplane is actually ready, it has been successfully tested and validated in terms of power and stability. The solar generator, composed of the solar modules and the MPPT, is in the integration phase on the wing.

Concerning the autopilot, the different parts of the system are being assembled and all functionalities will be tested during the first half of this year. In the summer, we should have achieved many flights and experiments to clearly evaluate the capabilities of our UAV.

6 Potential Applications

Small and high endurance UAVs find uses in a lot of varied fields, civilian or military. The civil applications, leaving side the military ones, could include coast or border surveillance, atmospherical and weather research and prediction, environmental, forestry, agricultural, and oceanic monitoring, imaging for the media and real-estate industries, and a lot of others. The target market for the following years is extremely important [11].

The great advantages of Sky-Sailor compared to other solutions would be without any doubt its capability to remain airborne for a very long period, its low cost and the simplicity with which it can be used and deployed, without any ground infrastructure for the launch sequence.

As an example, in the hypothetical case of forest fire risks during a warm period, a dozen Sky-Sailor, easily launched with the hand, could efficiently monitor an extended surface, looking for fire starts. A fast report would allow a rapid intervention and thus reduce the cost of such disaster, in terms of human and material losses.

Sky-Sailor would be as well a very interesting platform for academic research, in aerodynamics or control.

7 Conclusion

In this paper, the design of an ultra-lightweight UAV was presented, including details about its mechanical structure, the solar generator and the autopilot system. The approach adopted doesn't aim only at building an efficient autopilot, but also keeps in mind its future application. This is done by designing and selecting all the parts to obtain a lightweight and low-power airplane. We plan to perform the first experiments with the autonomous airplane during the first half of this year and a long endurance flight this summer.

Acknowledgement

The authors would like to thank all the people who contributed to the definition study, Samir Bouabdallah for fruitful discussions and advices on flying robots, Walter Engel for the realization of the mechanical structure and all the students who worked or are working on this project.

References

1. Davison A J (2003) Real-time simultaneous localization and mapping with a single camera, IEEE Int. Conf. on Computer Vision, ICCV-2003, pp. 1403-1410, Nice (France), October 2003
2. Lacroix S, Kung I K (2004) High resolution 3D terrain mapping with low altitude imagery, 8th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA'2004), Noordwijk (Pays-Bas), 2-4 Novembre 2004
3. Eisenbeiss H (2004) A mini unmanned aerial vehicle (UAV): system overview and image acquisition, International Workshop on "Processing and visualization using high-resolution imagery" 18-20 November 2004, Pitsanulok, Thailand
4. Kim J.-H, Sukkarieh S (2002) Flight Test Results of GPS/INS Navigation Loop for an Autonomous Unmanned Aerial Vehicle (UAV), The 15th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION) 24-27 September, 2002, Potland, OR, USA
5. Kim J.-H, Wishart S, Sukkarieh S (2003) Real-time Navigation, Guidance and Control of a UAV using Low-cost Sensors. In International Conference of Field and Service Robotics (FSR03), Japan, July 2003.
6. Brown A K, Lu Y (2004) Performance Test Results of an Integrated GPS/MEMS Inertial Navigation Package, Proceedings of ION GNSS 2004, Long Beach, CA, Sept. 2004
7. Atkins E M et al. (1998) Solus: An Autonomous Aircraft for Flight Control and Trajectory Planning Research, Proceedings of the American Control Conference, Pennsylvania, June 1998
8. Johnson E N et al. (2004) UAV Flight Test Programs at Georgia Tech, Proceedings of the AIAA Unmanned Unlimited Technical Conference, Workshop, and Exhibit, 2004.
9. Granlund G (2000) Witas: An intelligent autonomous aircraft using active vision. In Proceedings of the UAV 2000 International Technical Conference and Exhibition, Paris, France, June 2000. Euro UVS
10. DeGarmo M, Nelson G M (2004) Prospective Unmanned aerial vehicle operations in the future national airspace system, AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum, 20 - 23 Sept 2004, Chicago
11. Wong K.C, Bil C (1998) UAVs over Australia - Market And Capabilities, Paper No. 4, Proceedings of the 13th Bristol International Conference on RPVs/UAVs, Bristol, UK, 1998
12. Duffie J A, Beckman W A (1991) Solar Engineering of Thermal Processes, Second Edition. New York: Wiley-Interscience.

Control and Guidance for a Tail-Sitter Unmanned Air Vehicle

R. Hugh Stone

School of Aerospace, Mechanical and Mechatronic Engineering,
Building, J07, University of Sydney, NSW, Australia 2006
hstone@aeromech.usyd.edu.au

Summary. This paper details the control and guidance architecture for the T-Wing tail-sitter unmanned air vehicle, (UAV). The vehicle uses a mixture of classical and LQR controllers for its numerous low-level and guidance control loops. Different controllers are used for the vertical, horizontal and transition flight modes, glued together with supervisory mode-switching logic. This allows the vehicle to achieve autonomous waypoint navigation throughout its flight-envelope. The control design for the T-Wing is complicated by the large differences in vehicle dynamics between vertical and horizontal flight; the difficulty of accurately predicting the low-speed vehicle aerodynamics; and the basic instability of the vertical flight mode. This paper considers the control design problem for the T-Wing in light of these factors. In particular it focuses on the integration of all the different types and levels of controllers in a full flight-vehicle control system.

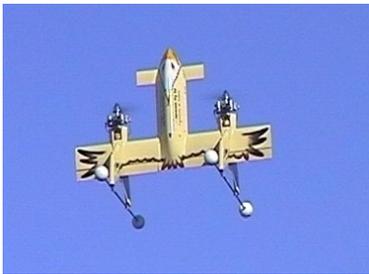
Keywords: UAVs, Guidance, Control, VTOL, Tail-sitter

1 Introduction

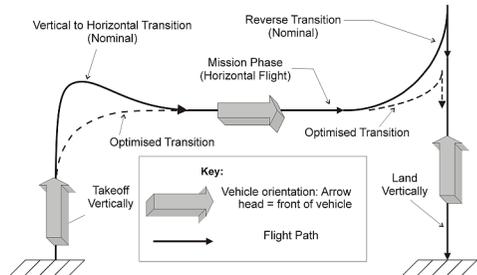
The T-Wing is a VTOL UAV that aims to combine the greater efficiency of wing-born flight with the operational flexibility offered by VTOL configurations such as the helicopter. The T-Wing is a tail-sitter twin-engined vehicle and is controlled during vertical flight via propeller-wash over its wing and fin-mounted control surfaces. In this respect it is similar to the early manned tail-sitter vehicles of the 1950s, the Lockheed XF-V1 and the Convair XF-Y1 [1, 2]. This allows the T-Wing to be substantially less complicated than other “convertiplane” configurations such as the tilt-wing, tilt-rotor or tilt-body. In overall configuration the T-Wing is most similar to the Boeing Heliwing of the early 1990s [3]. This was also a twin-engined vehicle but unlike the T-Wing used helicopter cyclic and collective pitch controls during vertical flight. A picture of the T-Wing vehicle during fully autonomous vertical mode flight is shown in 1(a), while a diagram of its typical flight profile is given in 1(b).

The fact that the T-Wing operates across a much wider range of speeds and attitudes than a conventional aircraft complicates the design of the flight control system for this vehicle. It is required to operate in vertical flight when the vehicle aerodynamics are dominated by the propeller slipstream effects and the dynamic modes are significantly unstable, as well as in forward flight when the vehicle behaves like a conventional aircraft. Due to the widely different behaviour of the basic plant between these two fundamental flight modes (vertical and horizontal) it is necessary to have a range of controllers to cover operation at these fundamental modes as well as the transitions between them. Although individually the controllers are relatively simple, the overall control system is quite complex and involves considerable switching logic to determine which controllers to use at any given time as well as how to smoothly transition between these controllers.

This paper will discuss the overall control system architecture of the T-Wing vehicle. Section 2 will outline the basic mathematical model of the vehicle, while Section 3 will discuss the on-board sensors and filters. Sections 4 and 5 will introduce some aspects of the control and guidance of the vehicle for vertical and horizontal flight phases respectively, while Section 6 will deal with the transition modes. Section 7 considers the flight-state logic and controller transition issues, before looking at the actual implementation of the control system and some flight-test results in Section 8.



(a) Autonomous hover flight



(b) Flight profile

Fig. 1. T-Wing Vehicle in full autonomous hover flight and Flight Profile. Styrofoam balls attached to fins are for tip-over protection during initial vertical flight testing

2 Basic Vehicle Model

For the purposes of simulation and control design, The T-Wing vehicle is modeled as a standard 6-DOF non-linear rigid-body aircraft model. The modeling is done in standard body-axes centered at the aircraft center of gravity where x points forward, through the aircraft nose; y is directed to the starboard, (right); and z is directed through the belly of the aircraft.

The standard body-axis equations of motion are given below using the notation of Stevens and Lewis [4]. In order these equations are the force (1), moment (2), kinematic [quaternion form] (3), and navigation (4) equations.

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \begin{bmatrix} +RV - QW + g_x + \frac{F_x}{m} \\ -RU + PW + g_y + \frac{F_y}{m} \\ +QU - PV + g_z + \frac{F_z}{m} \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \bar{L} \\ M \\ N \end{bmatrix} - \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix} \mathbf{J} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} 0 & P & Q & R \\ -P & 0 & -R & Q \\ -Q & R & 0 & -P \\ -R & -Q & P & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} \dot{p}_N \\ \dot{p}_E \\ -\dot{h} \end{bmatrix} = \mathbf{B}^{-1} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \quad (4)$$

In the above equations (U, V, W) are the body axis velocity states; (P, Q, R) are the body axis rates; (q_0, q_1, q_2, q_3) are the quaternion representation of the vehicle attitude; and (p_N, p_E, h) are the North, East and height positions. These variables form the basic vehicle state vector. The aerodynamic force and moment vectors are $\mathbf{F} = (F_x, F_y, F_z)$ and $\mathbf{M} = (L, M, N)$ respectively. \mathbf{J} is the standard inertia matrix for the vehicle, m is the mass, and \mathbf{B} is the transformation matrix that takes vectors from the North-East-Down (NED) frame to the body axis frame. The gravitational acceleration in the body axis frame is (g_x, g_y, g_z) The \mathbf{B} -matrix can either be obtained directly from the quaternion parameters or equivalently from other Euler angle representations.

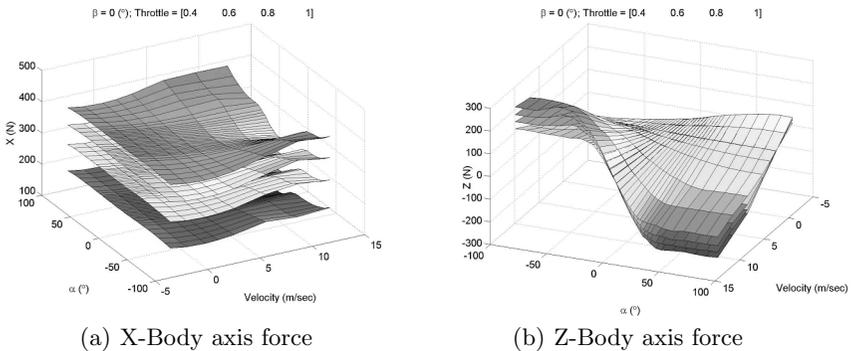


Fig. 2. X and Z Body forces plotted verses velocity (m/s) and angle of attack ($^\circ$)

In these equations non-linearities arise in both their basic structure as well as in the force and moment vectors \mathbf{F} and \mathbf{M} . This can be appreciated by considering the X and Z body axis forces plotted versus velocity and angle of attack as shown in Figure 2.

In Figure 2(a) the four stacked surfaces represent X -force values at different throttle settings. This particular graph is taken from an estimated aerodynamic database for the vehicle [5].

Unlike conventional aircraft where the non-linearity in the force and moment data is largely confined to the parabolic variation of these terms with speed, the non-linearities for the T-Wing are more complicated due to the changing relative importance of the propeller generated forces in comparison to those due to the free-stream dynamic pressure. This change occurs as the vehicle goes from low-speed vertical flight (propeller and propeller slipstream forces dominate) to high-speed horizontal flight (free-stream dynamic pressure dominate).

2.1 Attitude Representations

Three distinct attitude representations are used for the simulation and flight control of the T-Wing vehicle. These consist of a quaternion representation and two Euler angle representations.

The quaternion representation is used exclusively in the flight simulation of the vehicle, as well as in the guidance controller for the transition maneuvers between horizontal and vertical flight. The quaternion representation has the advantage of being unique (to within a choice of sign) and of having no areas of degeneracy of its solution.

For horizontal mode flight the standard ordered Euler angle rotations of yaw (ψ), pitch (θ) and roll (ϕ) about the vehicle z , y and x body-axes respectively are used to describe the vehicle's attitude.

Due to the degeneracy of the standard Euler angles for vertical flight attitudes, a second set of Euler angles has been defined for vertical mode flight. Starting from a vertical attitude with the vehicle belly (z -axis) facing North, these consist of a vertical roll (ϕ_v) [opposite sense to ψ] about the x axis; a vertical pitch (θ_v) about the y -axis; and finally a vertical yaw (ψ_v) about the z axis.

The advantage of coupling this new system in vertical flight with a standard aircraft system for horizontal flight is that the senses of pitch, roll and yaw are consistent between the different representations. Transformations between these three representations can be easily calculated [6].

It is also possible to express the kinematic equations of motion in terms of either the vertical or horizontal Euler angles. This is useful for control-design and is in-line with standard aircraft control practice. The rationale for using Euler angles for control is threefold. Firstly, their use allows the state variables (including attitude states) and controls to be separated into distinct longitudinal and lateral partitions. Secondly, within these partitions,

the approximate assignment of control surfaces to attitude states is invariant with Euler Angle attitudes. Lastly, it is much easier to relate meaningful control objectives to Euler angles than to the quaternion parameters.

3 Sensors and Flight Control Hardware

The vehicle has a fairly simple suite of on-board sensors to enable it to estimate its current state. A schematic of this system is shown in Figure 3. The primary components of the flight control system are as follows:

- A 400MHz Celeron flight computer in a PC-104 form-factor. This interfaces with the IMU and GPS units via standard serial communication.
 - A Honeywell HG1700AG17 Ring-Laser Gyro inertial measurement unit (IMU) comprising 3 accelerometers and 3 rate gyros, with 10° /hour drift.
 - A Novatel ProPak-G2 Plus 5Hz GPS receiver. This receives standard RTCM differential corrections from the ground station and also includes a dedicated Kalman filter to allow interfacing with the Honeywell IMU to give filtered position, velocity, attitude (PVA) estimates.
 - Analog voltage signals are used for a static pressure sensor to enable measurement of pressure altitude and rate of climb as well as for simple voltage dividers to monitor battery voltages. These are sampled via an AD card.
 - Control of the standard RC hobby servos is via PWM signals generated by a PC104 card.
 - The data-link to the ground is via a spread spectrum radio-modem.
- All significant state data is transmitted and recorded at 80Hz.

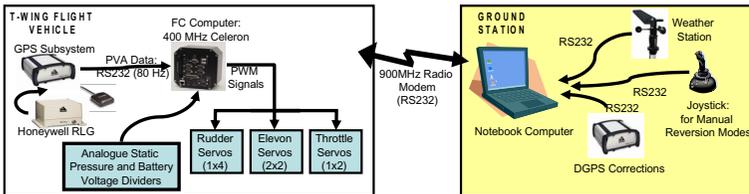


Fig. 3. Complete T-Wing UAV System

4 Vertical Flight Control and Guidance

During vertical flight the T-Wing uses a set of gain-scheduled LQR controllers to control translational velocities in the body axis y and z directions using the elevons and rudders. These are combined with a vertical roll-rate controller (for heading control – or in other words which direction the belly is pointing in) and a simple vertical velocity throttle controller. The vertical flight mode is where the vehicle is most unstable and also where the mode of operation is most novel in comparison to that of a conventional aircraft.

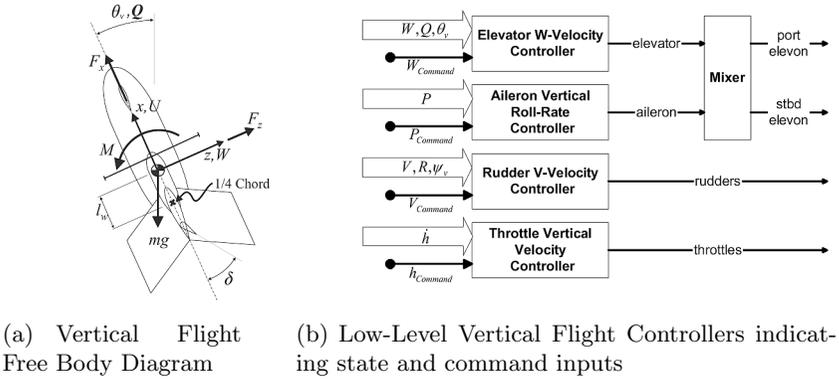


Fig. 4. Vertical Flight Free Body Diagram and Overall Control Structure

In considering the design of a vertical attitude translational W -velocity controller use will be made of Figure 4(a), which gives a free-body diagram of the vehicle, without lateral states. For low-speed vertical-mode flight the perturbation system, linearised about the hover trim state ($U \simeq 0$) in standard $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ state-space form is as follows:

$$\begin{bmatrix} \dot{W} \\ \dot{Q} \\ \dot{\theta}_v \end{bmatrix} = \begin{bmatrix} \frac{F_z W}{m} & \frac{F_z Q}{m} & -g_0 \\ \frac{M_W}{I_{yy}} & \frac{M_Q}{I_{yy}} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} W \\ Q \\ \theta_v \end{bmatrix} + \begin{bmatrix} \frac{F_z \delta_e}{m} \\ \frac{M_{\delta_e}}{I_{yy}} \\ 0 \end{bmatrix} \delta_e \quad (5)$$

In the above equation standard mechanics of flight notation has been used to represent the significant force and moment derivatives, (eg $M_W = \partial M / \partial W$ etc.). From this equation the trim elevon deflection and the trim vertical pitch angles can easily be determined [?].

This reduced set of equations can also be used to develop LQR controllers for the vehicle. During flight the W -velocity component, the pitch rate, Q , and the vertical pitch angle, θ_v , are all available for feedback control of the elevator via the onboard inertial and GPS sensors. The LQR design process gives rise to a vector of control gains $\mathbf{K} = [K_Q \ K_\theta \ K_W]$, which can be used directly in a W -translational velocity controller (rather than regulator) by replacing zero regulation on the W -state with zero regulation of a W -command error. This type of body-axis velocity control, though slightly unusual for an air-vehicle, has been found to work well. In practice, the gains are scheduled with vertical velocity. The design of a sideways V -velocity controller is similar.

The suite of vertical flight controllers is completed by an aileron roll-rate controller (yaw rate when viewed as a helicopter) and a throttle vertical velocity controller. These are simple SISO proportional controllers. The overall structure of the low-level vertical flight controllers is given in Figure 4(b).

Guidance during vertical flight is implemented via simple proportional guidance based on the current errors between the vehicle position and the next

waypoint. Waypoint definitions consist of North, East, Height (NEH) position coordinates and a pointing angle, (which is the angle that the vehicle's belly faces). The current errors in horizontal position are converted into errors in the local vertical, local horizontal (LVLH) reference frame of the vehicle, and are then used (with judicious saturation limits) to generate appropriate W and V velocity commands for the elevon and rudder controllers respectively. Height errors are similarly used to generate vertical velocity commands to the throttle controller, while pointing errors are used to supply vertical roll-rate commands to the aileron control-circuit.

5 Horizontal Flight Control and Guidance

For horizontal flight the vehicle uses pitch and yaw rate controllers for its elevator and rudder control surfaces as well as a roll-rate controller for the ailerons. Speed control is via throttle. All these controllers are designed using classical SISO root-locus techniques. The form of the pitch and yaw rate compensators are as given in equation (5) for the pitch-rate controller.

$$\delta_e = \frac{K(s + z_1)(s + z_1)}{s^2} (Q_{command} - Q)$$

As the pitch and yaw-rate controllers are gain-scheduled with velocity, this requires that there never be any significant GPS outages, (> 20 seconds). This is because the GPS velocity estimates are required to prevent inertial drift.

During horizontal flight, waypoints are represented as 3 position coordinates and a target speed at which to fly through the waypoint. As for the case for vertical flight, guidance is performed in the LVLH frame. In horizontal flight, heading is not independently controlled. Instead the vehicle compares its current position with the next waypoint and the off-axis error is converted into a desired sideways acceleration (in the LVLH frame) which in turn corresponds to a bank-angle command to turn the vehicle towards the waypoint. This is similar to well-known missile "proportional navigation guidance" coupled with bank-to-turn algorithms [?]. While performing a banked turn, yaw rate and pitch rate are commanded to match that of a coordinated turn. The climb-rate error is converted into an additional pitch-rate command increment and added to the turn pitch rate to allow the vehicle to transition between different altitudes of operation.

6 Transition Mode Guidance

For the transitions between vertical and horizontal flight the same low-level rate controllers are used as for horizontal flight. The main differences are to be found in the guidance used. For both transition maneuvers the guidance algorithm is based on determining a "quaternion velocity" that takes the vehicle from its current attitude to a target attitude as detailed below.

- A target attitude for the transition is selected.
- The target attitude is then converted to a quaternion and compared with the current vehicle attitude expressed in similar form.
- The quaternion difference between the current and desired attitude is then interpreted as a quaternion velocity vector over a nominal time increment and this is used to generate commanded rates for the vehicle controllers, (with suitable saturation limits applied) as given in (6).

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = K \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} \Delta q_0/\Delta t \\ \Delta q_1/\Delta t \\ \Delta q_2/\Delta t \\ \Delta q_3/\Delta t \end{bmatrix} \quad (6)$$

In the above equation, (P, Q, R) are the commanded rates, K is a gain, (q_0, q_1, q_2, q_3) is the current quaternion attitude and $(\Delta q_0, \Delta q_1, \Delta q_2, \Delta q_3)$ is the quaternion difference between the target and current attitudes. Lastly, Δt is a nominal transition time interval that converts the quaternion difference into a quaternion velocity vector.

During the transition manoeuvres, throttle can either be specified open-loop via a pitch-angle schedule or controlled with a velocity controller to match a pre-specified velocity schedule with pitch-angle.

7 Flight State Logic

The low-level and guidance controllers described so far allow the vehicle to operate autonomously throughout its complete flight envelope under non failure conditions given suitable logic to govern transitions between these different flight modes. However, the situation is complicated because a significant number of extra controllers and guidance modes are required for a full vehicle control system. These other controllers implement functionality to allow for sensor and system failures as well as for various levels of manual intervention in the vehicle control and guidance.

The main system failures that the control system must address are loss of GPS signal and loss of the communication link. In the case of loss of GPS signal the vehicle can only navigate for a relatively short period of time using its (low-accuracy) inertial sensors. This means that any controllers that depend on velocity information (such as the vertical flight velocity controllers) must be replaced with other controllers and guidance action must be taken to attempt a safe landing of the vehicle. In the case of sustained communication failure, the vehicle needs to abort its mission and return home. As well as handling these vehicle failure modes, it is also necessary to allow fully autonomous operation to be overridden with different levels of manual guidance and control for testing purposes.

The addition of manual and failure state modes of operation requires careful thought in terms of allowing or forcing mode transitions. For instance a

transition back to autonomous operation must be forced if the vehicle is in a manual mode at the time of a communication failure. Failures of GPS sensors or communications during any of the four normal flight modes (vertical, horizontal, and the two transition modes) must also trigger forced transitions to degraded modes of operation that involve the use of different sets of flight controllers or different guidance algorithms. This requires complex supervisory logic to sit above the guidance and control functionality.

To handle this problem in a consistent and rigorous manner, use has been made of the Mathworks Stateflow Toolbox that integrates with the rest of the SIMULINK simulation and rapid-prototyping environment in which all the flight control design and development is conducted. This toolbox allows graphical representation of all the flight mode logic as a finite-state machine. An example of a typical state transition diagram is shown in Figure 5.

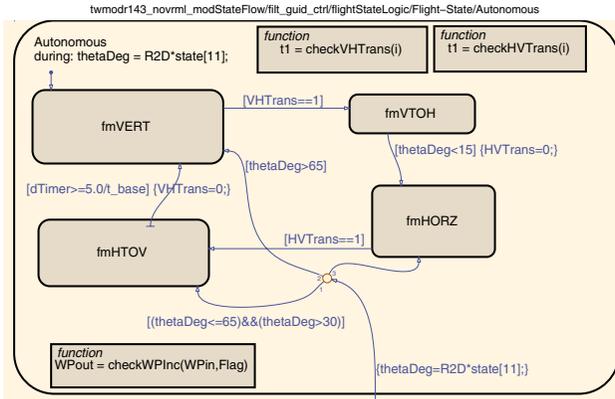


Fig. 5. Stateflow diagram for standard autonomous mode

Figure 5 looks inside the autonomous mode block of the overall Stateflow diagram at the transitions between the different autonomous flight conditions such as vertical and horizontal flight. Further drilling down into the individual autonomous modes would reveal extra flight states for various failure conditions. Going up a level would reveal transitions between autonomous and manual modes of operation.

The advantage of using Stateflow is that it formalizes the description of the state transitions in a readily understandable graphical representation for the whole control system. This obviates the need for manual coding of complex decision trees that is prone to coding errors and logical inconsistencies.

7.1 Controller Transitions

The fact that the T-Wing does not have one set of consistent controllers that operate throughout its entire flight envelope requires attention be paid to transitions between different controllers during control mode changes. The

most critical transition occurs at the re-entry to vertical flight before landing when the transition rate controllers are replaced with vertical flight translational velocity controllers. This is because the vehicle may still have significant residual translational velocity even after it has reached a vertical attitude and suddenly switching to velocity based controllers may cause significant control transients. To handle this the following procedure is used.

- On re-entering the vertical mode the control surface deflections and state-variables are used to back-calculate translational velocity guidance commands corresponding to the current vehicle state based on the known velocity mode control algorithms.
- These initial guidance commands are ramped down to zero over a set time period to bring the vehicle to a stable hover.
- The vehicle then starts to accept normal guidance commands to navigate to its landing location.

To see this consider the simplest form of the W -velocity controller.

$$\delta_e = K_W \cdot (W_{command} - W) - K_Q \cdot Q - K_{\theta_v} \cdot \theta_v \quad (7)$$

Denoting the states at re-entry to this controller as $[W_0, Q_0, \theta_{v0}]$, and the scheduled gains at re-entry as $[K_{W_0}, K_{Q_0}, K_{\theta_{v0}}]$ with δ_{e_0} , being the last value of the elevon deflection, the required value for the $W_{command}$ to prevent control transients is:

$$W_{comm.required} = \frac{\delta_{e_0} + K_{Q_0} \cdot Q_0 + K_{\theta_{v0}} \cdot \theta_{v0}}{K_{W_0}} + W_0 \quad (8)$$

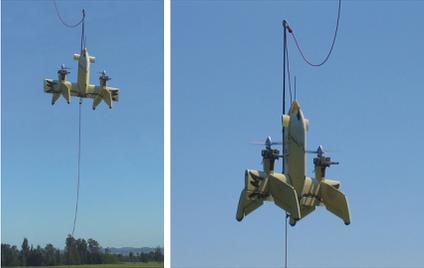
Using this value on entry to the W -controller and then ramping the command down smoothly to zero over say 5 seconds brings the vehicle to a stable hover, after which normal vertical mode guidance can take-over.

Control transition problems are not found in going from vertical flight to the horizontal-to-vertical transition mode because the translational velocity controllers automatically keep angular rates close to zero. Hence the entry to the transition rate controllers is well-behaved. As both sets of transition controllers use the same low-level rate controllers as are used for horizontal flight, there are also no problems in changing at either end of the horizontal flight mode.

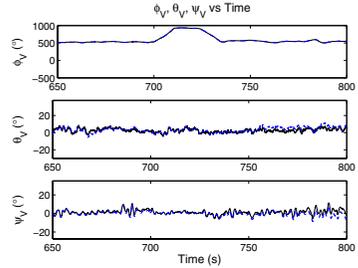
8 Implementation Details and Results

The complete control system for the T-Wing vehicle is developed and simulated within the SIMULINK simulation environment. To convert the control system to a hard real time flight controller, use is made of the Mathworks Realtime Workshop and xPCTarget toolboxes. These allow for the automatic generation of hard real-time code from the SIMULINK model as well as interfacing to external sensors and actuators through xPCTarget driver blocks.

The whole process of converting the simulated flight controllers into real-time controllers on-board the vehicle simply requires cutting and pasting the flight-control block from the simulation model into the flight control model and performing a real-time build.



(a) Demonstrator undertaking tethered hover tests March 2005. Both top and bottom tethers are slack



(b) Vertical Euler Angles for Tethered Flight Test March 2005. Vertical Roll Angle (top graph) shows positive and negative pirouettes.

Fig. 6. T-Wing Vehicle Flight Test Results

The T-Wing has flown numerous flights in different configurations including:

- Full manual hover flights (demonstrating the significant inherent instability of the vehicle in this condition);
- Tethered hover flights in which the pilot provides velocity guidance commands to the translational velocity controllers (Figure 6(a) and 6(b));
- Tethered 5-DOF autonomous flights in which operation is fully autonomous except for vertical velocity commands provided by the pilot;
- Velocity mode free flights (velocity guidance commands from pilot); and
- Fully autonomous vertical mode flight. A picture of the vehicle in this mode is shown in Figure 1(a).

9 Conclusion and Future Work

This paper has considered the overall flight control system for the T-Wing VTOL UAV, including low-level and mid-level guidance controllers for all the non failure-state flight modes that the vehicle must operate in. Although the individual controllers are relatively simple the overall control structure requires a complex layer of supervisory control logic to handle controller and flight-mode transitions. This has been implemented using Stateflow. So far the T-Wing vehicle has been successfully flown in a variety of vertical flight modes including fully autonomous vertical flight. It is anticipated that flight throughout the rest of the flight envelope will occur before the end of 2005.

In addition to this, a new model predictive controller will be flown on the vehicle starting in July/August 2005. This is based on successive linearisations of an onboard non-linear model of the vehicle dynamics and solving for the optimum control response over a finite time-horizon [?]. Besides promising better control performance it will also allow significant simplification in the overall control structure. This is because much of the guidance and low-level control functionality, which is currently contained within ~ 20 discrete subsystems, glued together with complex supervisory logic, will be replaced with a single unified predictive controller.

Acknowledgments

This work has been supported by Sonacom Pty Ltd through direct funding and technical collaboration as well as by ARC SPIRT Grant C89906759. Recent funding to demonstrate non-linear model predictive control has been through the US Air-Force Office of Scientific Research via Research Contract FA5209-04-P-0563.

References

1. L. Bridgman. *Jane's All the World's Aircraft*. Jane's All the World's Aircraft Publishing Co., London, 1955.
2. L. Bridgman. *Jane's All the World's Aircraft*. Jane's All the World's Aircraft Publishing Co., London, 1956.
3. K. Munson. *Jane's Unmanned Aerial Vehicles and Targets*. Jane's Information Group, Sentinel House, 163 Brighton Rd, Coulsdon, Surrey CR5 2NH, UK, 1998.
4. B.L. Stevens and F.L. Lewis. *Aircraft Control and Simulation*. John Wiley and Sons, New York, 1992.
5. R. H. Stone. Aerodynamic modeling and simulation of a wing-in-slipstream tail-sitter uav. In *Biennial AIAA International Powered Lift Conference*, Williamsburg, Virginia, USA, Nov 2002. The American Institute of Aeronautics and Astronautics.
6. R.H. Stone. *Configuration Design of a Canard Configured Vertical Takeoff and Landing Tail-Sitter Unmanned Air Vehicle Using Multidisciplinary Optimisation*. Phd thesis, Department of Aeronautical Engineering, University of Sydney, NSW, Australia, 1999.
7. R. H. Stone. The t-wing tail-sitter research uav. In *Biennial AIAA International Powered Lift Conference*, Williamsburg, Virginia, USA, Nov 2002. The American Institute of Aeronautics and Astronautics.
8. J.H. Blakelock. *Automatic Control of Aircraft and Missiles*. John Wiley and Sons, New York, 1991.
9. P. Anderson and R. H. Stone. Predictive control design for an unmanned aerial vehicle. In *11th Australian International Aerospace Congress*, Melbourne, Australia, March 2005. Australian International Aerospace Congress Organizing Committee.

The Development of a Real-Time Modular Architecture for the Control of UAV Teams

David T. Cole, Salah Sukkarieh, Ali Haydar Göktoğan, Hugh Stone, and Rhys Hardwick-Jones

ARC Centre of Excellence for Autonomous Systems
Australian Centre for Field Robotics
The Rose St. Building, J04
The University of Sydney, NSW, Australia 2006
{d.cole, salah, agoktogan, r.jones}@cas.edu.au,
hstone@aeromech.usyd.edu.au

Summary. This paper presents a modular architecture for the implementation and evaluation of control strategies for a team of autonomous Unmanned Aerial Vehicles (UAVs). The architecture is designed to be a test bed for control algorithms ranging from low level vehicle control to higher level multi-vehicle mission control. The architecture is being implemented on a team of Brumby Mk III fixed wing UAVs. Particular focus is on the testing procedure, through simulation, HardWare-In-the-Loop testing and real time flight tests. Results from recent flight testing of a path-following guidance algorithm used for feature orbiting are shown and future implementation of cooperative UAV strategies are presented.

Keywords: UAVs, Guidance and Control Architecture

1 Introduction

The utilisation of teams of UAVs is an ever growing research area. Applications (both civil and military) include search, surveillance, mapping and exploration. Such *information gathering* missions typically require detection, classification and tracking/estimation algorithms applied to ground based features or targets. The objective of the architecture presented in this paper is to enable the modular testing and demonstration of decentralised control strategies for teams of UAVs in an unstructured natural environment.

Control architectures for both individual and teams of UAVs need to provide a link between a mission description and the low-level control actions of the UAV(s). Similar architectures have used modular, hierarchical, and multi-levelled approaches to this problem [1], [3], [5], [8]. This paper also presents a modular solution to deal with the different aspects of control. The focus of this

paper will be on the current demonstrations of the Mission, Guidance, and Low-Level Control Modules in particular. Although the algorithms presented are generic in terms of hardware (flight vehicles, sensors, etc), this implementation has been developed for testing and demonstration on a fleet of Brumby Mk III UAVs.

This architecture is designed specifically for information gathering missions such as those mentioned above. Such missions are reliant on the data received from onboard mission sensors. The quality of this data is extremely important to the success or failure of the mission. The proposed architecture builds upon the current Decentralised Data Fusion (DDF) implementation [7], making the feature space¹ integral in the control of the UAVs.

The architecture also aims to preserve the decentralised nature of DDF. Multiple vehicles can more efficiently perform a task such target tracking by sharing information and dividing the workload. In terms of control this requires team members to communicate with each other to determine the individual actions which produce the best outcome for the team. Keeping the team structure decentralised maintains the characteristics of modularity, scalability and survivability.

The structure of this paper is as follows. Section 2 describes the platforms, the sensors, and the environment used for testing and evaluating the UAV control algorithms. Section 3 outlines the modular algorithmic architecture used within this project. Section 4 discusses specifically the Mission, Guidance, and Low-Level Control modules within the architecture and their application towards information gathering missions. Section 5 focuses on the testing procedures and provides results for recent flight tests of path following guidance algorithms. Concluding remarks and comments about future work are given in Section 6.

2 The Experimental System

The three main components for the real-time testing of the onboard algorithms are the platforms, the sensors, and the environment.

2.1 Platforms

The platforms used in this project are a fleet of Brumby Mk III UAVs developed specifically for research at The University of Sydney (Figure 1). They have a delta wing configuration, weigh approximately 45 kg and have a typical mission time of 40 minutes. The flight vehicles use an Inertial Measurement Unit (IMU) and Differential GPS (DGPS) solution for localisation [6]. Flight Control and Localisation algorithms are run on a 700 mhz PC-104 stack.

¹ This paper uses the terms *feature* and *feature space*, to describe the objects of interest and the corresponding states of interest.

Each Flight Vehicle (FV) has full-duplex communications with the Ground Station (GS) and with each other. Remote control signals, DGPS corrections, and Mission commands are received from the ground via a radio modem. The FV system status and navigation data are sent to the GS for monitoring. Wireless ethernet communication between the FVs is used to share information relevant to the mission (e.g. DDF and negotiation communications).



Fig. 1. Two Brumby Mk III Flight Vehicles

2.2 Sensors

A sensor payload bay is located in the nose of each aircraft. Interchangeable sensors (including Colour and Infra-Red (IR) Vision, and Millimetre Wave Radar) can be configured in various orientations for a given mission (Figure 2). A black and white camera is also located in the fuselage pointing downwards. The current sensor setup for the UAVs is both a colour² and IR³ camera pointing along the lateral axis of the left wing. This configuration allows for continuous observations of any region on the ground by orbiting around it. The guidance and control algorithms presented in Section 4 enable the FV to autonomously select and orbit around ground features of interest. Sensor images are used by the feature detection/estimation algorithms.

DDF is a process used on the UAVs which enables them to share information about a common state-space without the need for a central processor or communication source [7]. Within in this project the feature space comprises of the position/velocity of ground features of interest. The sharing of information is extremely important in the types of missions considered here where team members are required to work together. The same communication network set up by DDF can also be used to share information regarding UAV actions as well (Section 4.3).

2.3 Environment

The testing and demonstration environment is part of a 7000 hectare farm (Figure 3). A 300m runway and permanent ground station/mission control enclosure have been built specifically for the FV tests.

² Digital firewire, 1024×768 pixel resolution, 20 Hz frame rate.

³ Analogue video, 255×255 pixel resolution, 50 Hz frame rate.

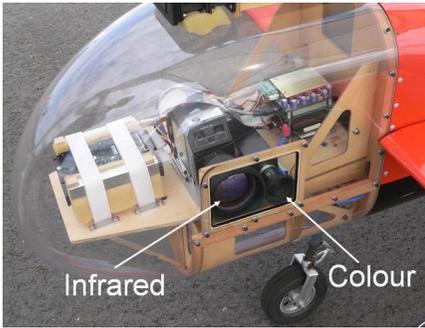


Fig. 2. Colour and IR cameras configured to point out the left wing of the UAV. A hard-disk and PC-104 stack for vision processing are also located in the nose.



Fig. 3. An image of the environment used for testing taken by the onboard colour camera.

The farm hosts a large number of interesting natural features within the area of flight operation. These include: a creek, a number of dams, individual and groups of trees, livestock, wombat holes, and ground with varying degrees of vegetation.

Man made features include: a number of sheds and buildings, fences, dirt roads, the runway, ground vehicles and ground vehicle tracks. In addition to this white targets (1 m^2) and large patterned cylinders are placed at specific locations for the calibration and testing of estimation and classification/detection algorithms. These features allow us to simulate the types of missions described in Section 1.

3 Algorithmic Architecture

This section provides an overview of the architecture which allows for the testing and demonstration of control strategies for teams of UAVs. Section 4 describes in more detail the modules specific to control. The aim of the architecture is to act as a modular test bed for controlling UAV teams from low level single vehicle control to multi-vehicle mission control.

Figure 4 depicts the interaction between the onboard modules of each UAV and the communication between UAVs in the team. An outline of each of the modules is given below:

Localisation Module: An INS/GPS solution for localising the FV. Outputs the FVs current state (position, velocity, attitude), for use by all other modules.

Sensor Node: The sensor node is responsible for controlling payload sensors and extracting useful information about features of interest from raw sensor observations. This information takes the form of a likelihood over the feature space and is communicated to the DDF module.

DDF: This module fuses information about the feature space from onboard sensors and from other FVs. The result is a prior distribution over the feature space. Although Gaussian representations for probability distributions are assumed here, general distributions may also be used.

Guidance and Control Modules: Form the lower levels of individual vehicle control. As a whole they are responsible for generating actuator commands in order to follow specific paths or trajectories (Sections 4.1 and 4.2).

Mission Controller and Path Planning Module: This is the highest level of autonomous control. Responsible for determining the FV path and which features/areas to observe. For Decentralised Control (DC) this module is responsible for communicating/negotiating with the other FVs so that planned actions/observations are beneficial to the team as a whole. This module creates paths to send to the Guidance Module (Section 4.3).

State Machine: The State machine is responsible for changing the state/mode of the FV. State changes are dependent on commands received from the ground as well as variables monitored onboard (Section 4.4).

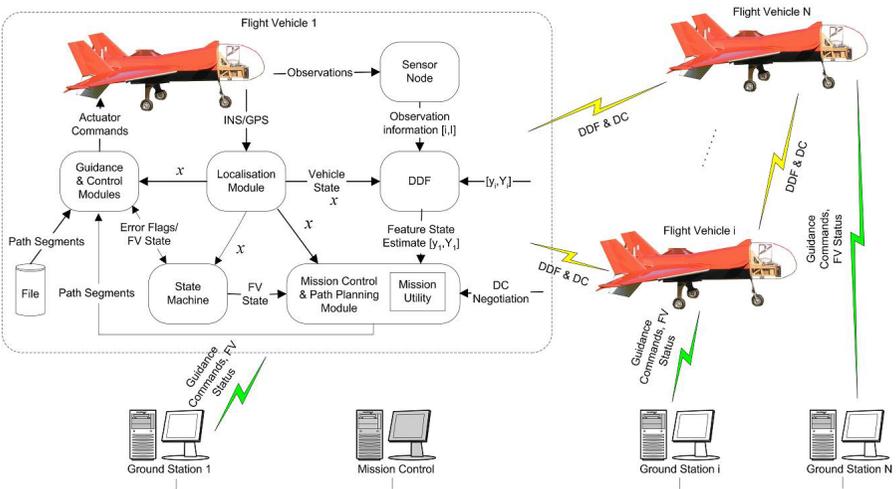


Fig. 4. Architecture for control of teams of UAVs.

4 The Mission, Guidance, and Control Modules

This section describes in more detail the modules directly responsible for the control of the vehicle, from the low level control module to the mission controller.

4.1 Control Module

The Control Module implements the inner-most loop of the flight vehicle guidance and control architecture. The basic architecture of the control module is

shown in Figure 5. The function of the control module is to apply FV specific data and functions to a generic set of control laws in order to track the commanded inputs from the guidance module (airspeed, height above sea level, and bank angle). The resulting outputs are 16 bit servo commands for the FV's actuators.

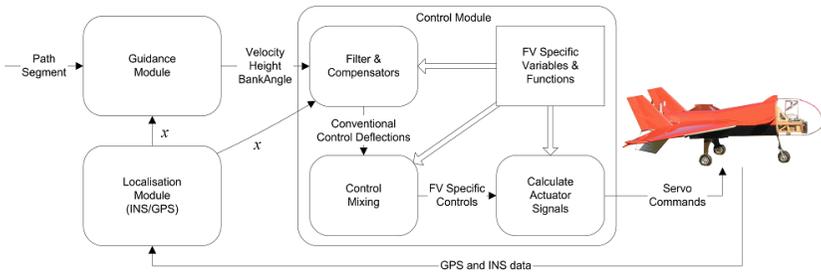


Fig. 5. The control module takes command from the Guidance Module and produces commands for the FV servos. Feedback from the Localisation Module closes this inner loop.

The general nature of the control algorithm allows the same control module to be used for a variety of fixed-wing UAVs, by providing a separate set of control parameters for each platform type.

4.2 Guidance Module

The Guidance Module is responsible for calculating demanded inputs to the Control Module in order to follow a desired path. The path can be either from a pre-loaded file, the online path planner, or the ground station. The investigation into control strategies for the detection/localisation of ground features prompted a change from a way point controller to the path following guidance controller. With side mounted vision sensors and the ability to follow a circular path defined in space, the FV is able to constantly keep a given feature within view.

The resulting angular variation of sensor feedback with respect to the feature is beneficial to both localisation [4] and classification/detection algorithms [9]. Figure 6 shows the flight envelope which the Brumby Mk III is able to fly autonomously around a feature. The view angle can be varied between 12 and 55 degrees.

A generic path following controller has been developed which calculates the required Control Module inputs in order to follow any given path. The current implementation can handle line segments, arc segments and way points. However the list can be easily extended to include other path types such as clothoids or splines. This modularity facilitates the investigation into using families of curves to approximate optimal paths for localisation/classification.

A PD control law is used to calculate the required heading rate ($\dot{\psi}$) to correct any errors to the path (1). The bank angle (ϕ) to achieve this heading

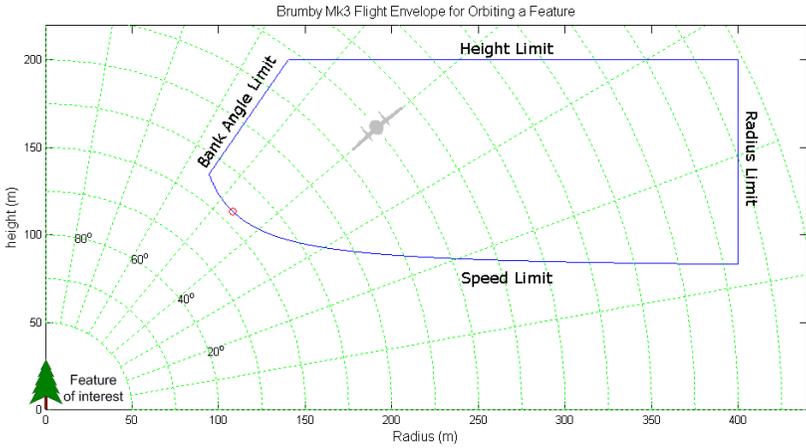


Fig. 6. A cross-section through the flight envelope for the Brumby Mk III FV orbiting a feature. The graph shows the possible height, radius and corresponding bank angle for orbits which the FV can fly in order to constantly view a feature at the origin. The borders of the flight envelope are defined by safety margins and the physical capabilities of the FV. The circle on the lower left of the envelope depicts the point with closest sensor range to the feature.

rate is then calculated (2), and then limited to maximum values based on maximum vertical load factor and the current FV speed. Figure 7 shows how the error to the path (ε - Normal displacement, η - Error in flight path angle) are calculated. The natural rate of change of heading of the path is represented by $\dot{\psi}_0$.

$$\dot{\psi}_{dem} = -K_{\eta}\eta - K_{\dot{\eta}}\dot{\eta} - K_{\varepsilon}\varepsilon - K_{\dot{\varepsilon}}\dot{\varepsilon} + \dot{\psi}_0 \tag{1}$$

$$\phi_{dem} = \arctan\left(\frac{\dot{\psi}_{dem}V_{true}}{g}\right) \tag{2}$$

Real-Time results for the path following algorithms are shown in Section 5.3.

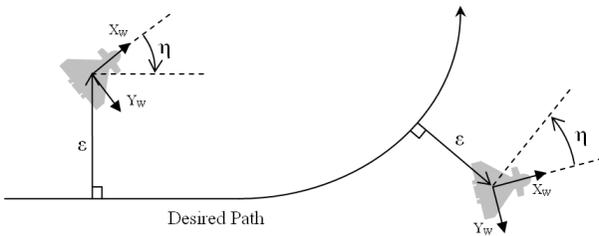


Fig. 7. Path errors used in the PD controller. Normal displacement error ε , and flight path angle error η (orientation error is measured w.r.t. the FV velocity vector, i.e. the FV wind axis $[X_w, Y_w]$).

4.3 Mission Controller and Dynamic Path Planning

This module is responsible for deciding the high level actions of the UAV. This includes which areas to search/observe and what manoeuvres to make to get the best observation with respect to the mission. These decisions are made in an information-theoretic manner. The control decisions are made using the current feature state provided by the DDF module (Figure 4).

Information-theoretic missions can often be described by a Mission Utility function, J . It is the aim of the mission controller to choose an action which maximises this utility function,

$$\hat{a} = \arg \max_a J([\mathbf{Y}, \mathbf{y}], \mathbf{x}, a). \quad (3)$$

Where \mathbf{y} and \mathbf{Y} are the feature information state and information matrix [4], and \mathbf{x} is the current FV state. The optimal action, \hat{a} , may be as simple as selecting which feature to observe next, or more complicated such as choosing the parameters for a path which will provide the best observations of a given feature. For the task of target tracking the utility function can be simply the information gain resulting from observations made due to action a .

Another aspect which this module addresses is decentralised control. Employing teams of UAVs to conduct a single mission means each UAV will have the same utility. In order to achieve the mission efficiently the UAVs must communicate with each other values relevant to the mission utility. One such type of communication is to send preferences for actions which the UAV is considering (e.g. which feature the UAV is to observe next). A solution can be found through bargaining or decentralised auction techniques.

Alternative methods include a *negotiation filter* which communicates the future expected information state (\mathbf{y}, \mathbf{Y}) given the action the UAV is considering [2], [4]. By fusing this information each vehicle can plan their actions by maximising the utility function over a distribution which accounts for the other vehicles actions. The decentralised control scheme is built upon the pre-existing DDF network and shares the same communication network.

4.4 The State Machine

The State Machine performs logic operations to decide what state/mode the FV should be in. The possible guidance states are shown in Figure 8. The transitions between states are triggered by changes in commands from the GS and/or internal flags relating to the FV status.

The guidance states are split between Remote and Autonomous. While in Remote mode the FV is controlled via the pilot's handset. The Autonomous states are separated into two categories; Scenario and Emergency. Scenario states are used for the normal operation of the Autonomous mode. Emergency states are reserved for when part of the software or hardware fails. Events which can trigger entry into an emergency state include *Engine Failure*, *Loss of Communications*, and an *Invalid Path Segment*.

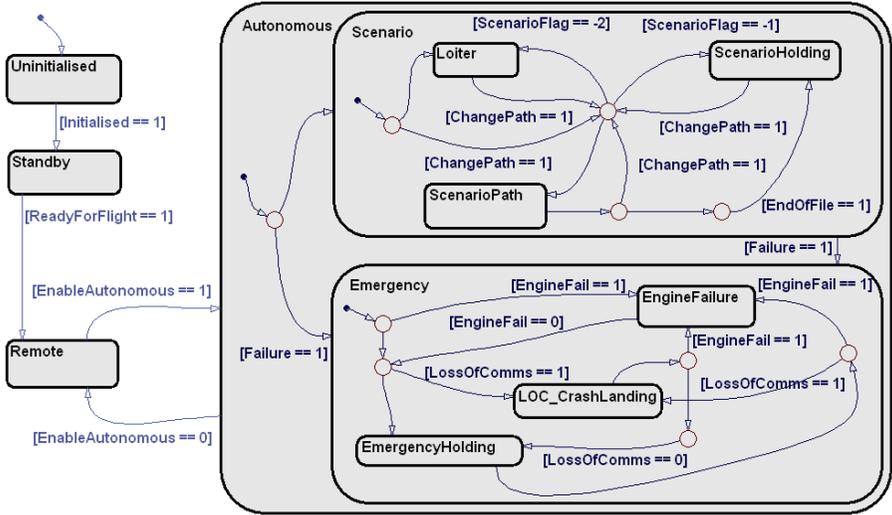


Fig. 8. FV Guidance Stateflow generated using the Matlab Stateflow toolbox. A more detailed version of the above figure was used in Matlab to auto-generate C code for integration into the State Machine.

5 Testing and Evaluation

This section outlines the various degrees of testing which occur for any change to a module within the system. Generally there are 3 stages which are used for verification and validation: A C++ real-time simulator, a HardWare-In-the-Loop simulation and actual flight tests.

5.1 Real-Time Multi-UAV Simulator (RMUS)

RMUS is a C++ simulator environment which enables real-time flight vehicle code to be tested and evaluated. The simulation also enables the user to view expected sensor data for a given path. High resolution aerial map data of the Brumby Mk III testing and demonstration environment is used to generate a view from the sensor frame. Figure 9 shows results of a UAV following a path which has it orbiting around a ground feature of interest.

5.2 HardWare In the Loop Simulation (HWIL)

HardWare-In-the-Loop (HWIL) simulations are a robust way of verifying both software and hardware without the costs and risks associated with a flight test [3], [5]. A complete HWIL simulation is conducted prior to every mission. This involves real-time testing of every component working within the system as a whole. The HWIL setup is shown in Figure 10. All FV software modules run on the onboard processors of the plane.

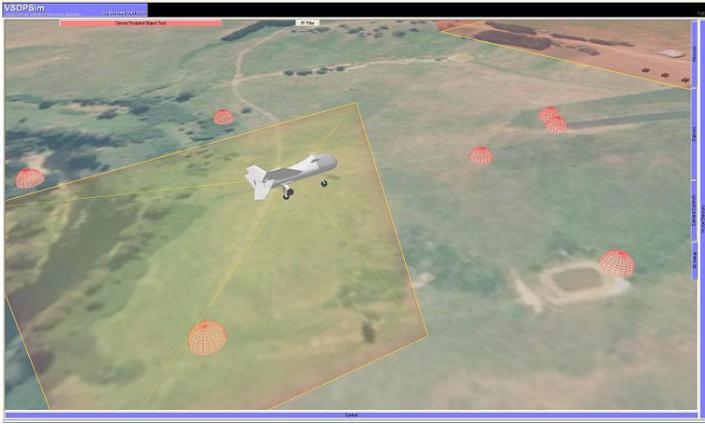


Fig. 9. RMUS: A Brumby Mk III UAV orbiting a ground feature. The sensor footprint is projected onto the map of the testing environment (Figure 3).

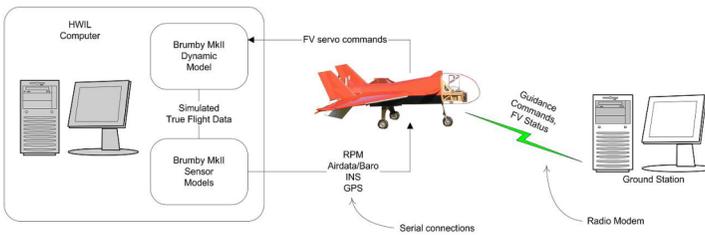


Fig. 10. HWIL setup: The servo commands are sent to the HWIL computer which uses a detailed dynamic model of the FV and sensor models to generate expected sensor outputs which are sent back to the Flight Control Computer.

5.3 Flight Tests

Recent flight tests were conducted to validate the path following guidance algorithms and gather sensor data of ground features. Flight paths were provided to the Guidance Module from a predefined file. Figure 11 shows the result of orbiting around a dam at a demanded height of 200 m and airspeed of 42 knots⁴. The methods described in section 4.2 kept the dam within the camera frame for the duration of the orbit. Figure 12 shows two of the frames obtained from opposite sides of the orbit. Orbit data was collected from three different points within the envelope for each of 16 different ground features.

6 Conclusion and Future Work

An architecture has been developed for the testing and demonstration of control strategies for a UAV and teams of UAVs. The focus has been to provide a modular test bed for control algorithms for information gathering missions.

⁴ These values correspond to the top left corner of the flight envelope (Figure 6).



Fig. 11. An orbit around a feature (dam). The desired path (black line, 170 m radius) and navigation solution (white line) are indicated. At 1 second intervals, the position of the FV (white circles), and the projected centre of the sensor frame on the ground (white crosses) are shown.



Fig. 12. Two images taken of the dam from opposite sides of the autonomous orbit.

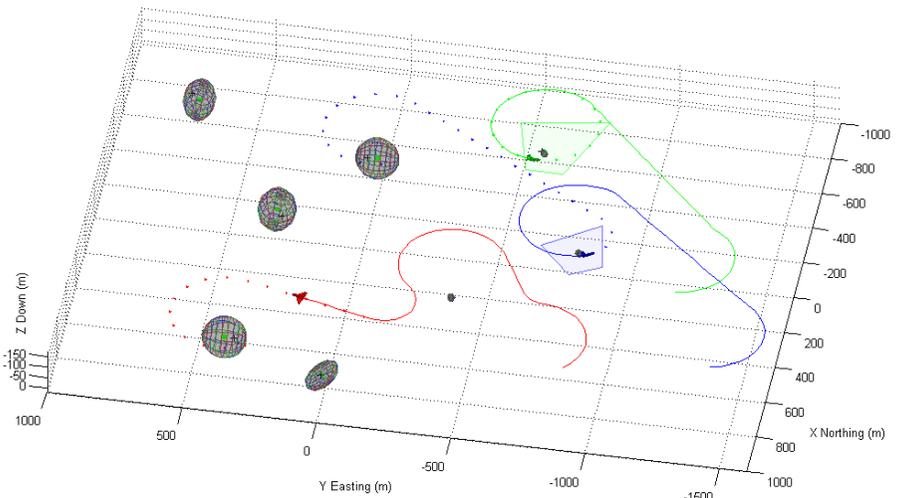


Fig. 13. Three UAVs performing DDF negotiate over which features to observe. Ellipsoids represent the current estimate and uncertainty of the features. Shaded quadrilaterals are the sensor footprints on the ground. The planned path of each UAV is represented by a dotted line. Decisions are made based on the current feature information and the time involved in observing the feature.

The development of the Mission, Guidance, and Control Modules has been described along with the methods for testing and demonstrating them. Re-

cent flight tests have validated a path following guidance module for observing ground features.

Further planned flight tests include validating online path planning and multi-vehicle negotiation. Figure 13 shows results from a 6 DoF Matlab simulation of multiple UAVs performing DDF and negotiating over actions to observe ground features. The UAVs plan paths to orbit feature estimates in order to maximise feature information. The simulation uses the same Mission, Path Planning and Control modules described in Section 4. Simulated sensor observations are modeled on the colour camera used onboard the UAVs. The UAVs orbit each feature until the information reaches a certain threshold.

Future work on the project also includes investigating automatic take-off and landing for the UAV enabling completely autonomous missions from start to finish; extending the supported path segments to include other types of curves and 3 dimensional paths, and the use of the designed architecture to investigate further cooperative control strategies for UAV teams.

Acknowledgments

This work is supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government. The project is also partly funded by BAE Systems UK.

References

- [1] S. Bayraktar, G.E. Fainekos, and G.J. Pappas. Experimental cooperative control of fixed-wind unmanned aerial vehicles. In *43rd IEEE Conference on Decision and Control*, pages 4292–4298, Atlantis, Paradise Island, Bahamas, 2004.
- [2] F. Bourgault and H.F. Durrant-Whyte. Communication in general decentralized filters and the coordinated search strategy. In *Proceedings of the Seventh International Conference on Information Fusion*, Stockholm, June-July 2004.
- [3] J. Evans, G. Inalhan, J.S. Jang, R. Teo, and C. Tomlin. Dragonfly: a versatile uav platform for the advancement of aircraft navigation and control. In *Proceedings of the 20th IEEE Digital Avionics Systems Conference*, October 2001.
- [4] B. Grocholsky. *Information-Theoretic Control of Multiple Sensor Platforms*. PhD thesis, The University of Sydney, 2002.
- [5] J. How, E. King, and Y. Kuwata. Flight demonstrations of cooperative control for uav teams. In *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, Chicago, Illinois, September 2004.
- [6] J.H. Kim, S. Wishart, and S. Sukkarieh. Real-time navigation, guidance, and control of a uav using low-cost sensors. In *The 5th International Conference of Field and Service Robotics*, pages 95–100, Yamanashi, Japan., July 2003.
- [7] E.W. Nettleton. *Decentralised Architectures for Tracking and Navigation with Multiple Flight Vehicles*. PhD thesis, The University of Sydney, 2003.
- [8] Rathinam, Zennaro, and Mak. An architecture for uav team control. In *IAV Conference 2004, IFAC symposium on intelligent autonomous vehicles*, 2004.
- [9] P. Thompson and S. Sukkarieh. Development of an angular characterisation system for cooperative uav and ugv applications. In *The 5th International Conference on Field and Service Robotics*, 2005.

Trajectory Generation on Rough Terrain Considering Actuator Dynamics

Thomas M. Howard¹ and Alonzo Kelly²

¹ Carnegie Mellon University, Robotics Institute, 5000 Forbes Avenue, Pittsburgh, PA 15213 thoward@cs.cmu.edu

² Carnegie Mellon University, Robotics Institute, 5000 Forbes Avenue, Pittsburgh, PA 15213 alonzo@ri.cmu.edu

Summary. Trajectory generation has traditionally been formulated on the assumption that the environment is flat. On rough terrain, however, deviations of the angular velocity vector from the vertical lead to errors which accumulate in a manner similar to the accumulation of attitude errors in odometry. In practice, feedback control can compensate for these errors in modeling by adjusting the path in real time. In many realistic cases, however, the 3D shape of the terrain is known beforehand, so it is possible to incorporate terrain shape into the predictive model — rather than treat it as an unknown disturbance. This paper presents an algorithm for trajectory generation which compensates for terrain shape in a predictive fashion. The numerical implementation makes it adaptable readily to a broad class of vehicles and even a broad class of predictable disturbances beyond terrain shape. In support of the latter, we demonstrate the ability to invert models of actuator dynamics and wheel slip concurrently with 3D terrain. An example application for a the Rocky 7 Mars rover platform is presented.

Keywords: Trajectory Generation, Rough Terrain, Robot Control, Nonholonomic, Mobile Robots

1 Introduction and Notation

Trajectory generation for mobile robots is related to the two point boundary problem of classical differential equation theory. It can be defined as the problem of finding a set of controls which satisfy initial and terminal position, pose, or posture constraints. Position is defined as a location in space (x, y, z) ; pose adds orientation $(x, y, z, \phi, \theta, \psi)$; and posture/state includes rates of orientation $(x, y, z, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi})$. The rates of orientation can be expressed in Euler angles $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ or locally referenced angular velocities $(\omega_x, \omega_y, \omega_z)$. This paper addresses the most general problem of trajectory generation between two arbitrary postures. Among other motivations, the appearance of higher

derivatives in the constraints allows for smoother transitions between adjacent trajectory segments.

1.1 Motivation

Future missions of planetary exploration will require rovers to navigate difficult terrain with limited human supervision [3]. The use of an actuator dynamics and terrain interaction models allows for more capable, reliable, and competent autonomous navigation. Better behaved and more intuitive trajectories are generated, and the need for teleoperation and supervisory control is reduced. By allowing the rover to navigate more accurately and safely through rough terrain, it may even enable a new class of missions.

Research into fast and efficient trajectory generators is important because of limited computing resources on planetary rovers. Current trajectory generation algorithms adopt the flat plane assumption, neglecting the influence of attitude in the solution, and they ignore such matters as steering delay and wheel slip. When incorrect, all of these omissions can lead to infeasible or unsafe generated paths.

Errors induced by rough terrain and dynamics can be treated as disturbances in the system. However, there are cases where, on the scale of a few vehicle lengths, disturbances cannot be compensated by feedback [7]. Such following errors can result in collisions with obstacles near the path or incorrect terminal postures for instrument placement. Conversely, algorithms incorporating sufficiently predictive models for these effects can improve path following performance by compensating for the known component of these models in the generation process.

1.2 Prior Work

Prior work in trajectory generation algorithms involved using polynomial spirals parameterized by arc length to represent curvature. Clothoids (curvature primitives which vary linearly with arc length) have been used to generate trajectories for years; however clothoids have three degrees of freedom, which is insufficient to generally satisfy constraints which would ensure curvature continuity between paths. A method was developed to generate trajectories between postures based on a composite of clothoids in [5], but it required an intermediate posture to be determined by intersecting circles and was not able to solve for a path between any two arbitrary postures.

A method using higher order curvature polynomial spirals was developed by [4]. This method used energy minimization to successively deform a curve until it satisfied the constraints. In [1], a real-time algorithm is proposed which solves the planar trajectory generation problem between two postures by inverting the forward model of the vehicle. This method is adapted here to the 3D problem. A solution for a third-order curvature control parameterized in arc length ($\kappa(s) = \kappa_0 + as + bs^2 + cs^3$) is found iteratively by modifying the

guess of parameters ($\underline{p}=[a,b,c,s]^T$), by an amount determined from inverting a linearization of the state equations, until the error in the terminal posture ($\Delta \underline{x}$) is sufficiently small:

$$\Delta \underline{x} = J \Delta \underline{p} \tag{1}$$

$$\Delta \underline{p} = J^{-1} \Delta \underline{x} \tag{2}$$

$$\underline{p}' = \underline{p} + \Delta \underline{p} \text{ until } \Delta \underline{x} \cong 0 \tag{3}$$

This method assumed that the robot operated on the x-y plane, which is equivalent to setting the attitude (ϕ, θ) to (0,0). The flat terrain assumption decouples the state equations — satisfying the ($z, \phi, \theta, \dot{\phi}, \dot{\theta}$) constraints everywhere. The x and y state equations take the form of Fresnel integrals, so their derivative must be approximated numerically. An approximate Jacobian is found by performing a finite difference of the x and y state equations:

$$\frac{\partial f_{ij}}{\partial p_k} = \frac{f_{ij}(p_k) - f_{ij}(p_k + e)}{e} \tag{4}$$

The problem of path planning in rough terrain was addressed in [2], where arcs calculated on a locally flat plane were used to connect vehicle positions between starting and ending poses. This method globally accounted for rough terrain but locally employed the flat-plane assumption.

2 Models

This section presents a general set of models for predicting the terminal state of a mobile robot from its command inputs. Models are presented in ascending order, beginning with wheel-terrain interaction and actuator dynamics and ending with the terminal pose integral.

2.1 Rate Kinematics, Actuator Dynamics, and Wheel Slip

The purpose of this model is to predict the difference (caused by wheel slip and actuator dynamics) between the requested and actual linear and angular velocities of the vehicle body. The forward rate kinematics model determines the wheel contact point velocity vector \underline{A}_{in} for each wheel from the body-frame linear and angular velocity inputs ($\underline{V}_{in}, \underline{\Omega}_{in}$):

$$\underline{A} = [\underline{v}_{c_1}^T \ \underline{v}_{c_2}^T \ \cdots \ \underline{v}_{c_n}^T]^T = h(\underline{V}_{in}, \underline{\Omega}_{in}) \tag{5}$$

A response wheel contact velocity vector \underline{A}_{out} is determined from the actuator dynamics and wheel slip models $\underline{A}_{out} = i(\underline{A}_{in})$. The inverse rate kinematics model then determines the body frame linear and angular velocity outputs ($\underline{V}_{out}, \underline{\Omega}_{out}$) based on the output wheel velocity vector \underline{A}_{out} . A flowchart describing the model can be seen below in Figure 1:

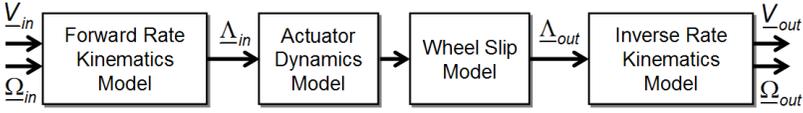


Fig. 1. Actuator Dynamics and Wheel Slip Model: Actuator dynamics and wheel slip is modeled at the wheel level. Forward and inverse rate kinematics are used to transfer between wheel frame and body frame velocities.

2.2 Suspension Kinematics

The previous model determined the linear and angular velocities of the body in the body frame. The purpose of the suspension kinematic model is to determine the attitude of the vehicle under suitable assumptions of terrain contact so that these velocities can be mapped to the world frame in the next step. In general, the attitude cannot be determined in closed form from the terrain. The mechanism used is to start with a forward model, defined to produce wheel contact point coordinates with respect to the body frame, given the suspension variables ($\underline{B}=\beta_1, \beta_2, \beta_3, \dots$). This model is inverted to produce the body attitude by enforcing a terrain contact constraint and determining the robot configuration which minimizes the error between the wheel contact points and the terrain (see for example [6]). The attitude (ϕ, θ) and altitude (z) of a robot is a function of the pose (x, y, ψ), suspension variables (\underline{B}), and terrain shape ($z(\cdot)$):

$$[\phi \ \theta \ z]^T = g(x, y, \psi, \underline{B}, z(\cdot)) \tag{6}$$

2.3 Kinetic Motion Model

The kinetic motion model maps linear and angular velocities in the body frame to linear velocities and Euler angle rates in the world frame. This paper follows the SAEJ670e convention, where the x-axis points forward, y-axis to the right, and the z-axis straight down. The Euler angles yaw (ψ), pitch (θ), and roll (ϕ) represent subsequent rotations coinciding with the robot frame about the z, y, and x axis respectively. The world frame velocity $\underline{\dot{R}}$ is found by the product of the rotation matrix and the robot frame velocity \underline{V} :

$$\underline{\dot{R}} = rot_z(\psi)rot_y(\theta)rot_x(\phi) \cdot \underline{V} \tag{7}$$

$$\underline{\dot{R}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{8}$$

The angular velocities in the robot fixed frame are determined by transforming the individual rotation rates ($\dot{\phi}, \dot{\theta}, \dot{\psi}$) from their intermediate frames to the robot-fixed frame:

$$\underline{\Omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \text{rot}_x(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \text{rot}_x(\phi)\text{rot}_y(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (9)$$

$$\underline{\Omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & s\theta \\ 0 & c\phi - s\phi c\theta \\ 0 & s\phi & c\phi c\theta \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (10)$$

Roll, pitch, and yaw rates in terms of the angular velocities can be found by inverting this relationship:

$$\underline{\dot{\Psi}} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & t\theta s\phi & -t\theta c\phi \\ 0 & c\phi & s\phi \\ 0 & -\frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (11)$$

Throughout this paper, \underline{X} will represent the vehicle configuration vector $[\underline{R}, \underline{\Psi}]^T$. Equations (8) and (11) can be concatenated to produce the kinetic motion model:

$$\begin{bmatrix} \dot{R} \\ \dot{\Psi} \end{bmatrix} = f(\underline{V}, \underline{\Omega}, \underline{X}) \quad (12)$$

Simply Actuated Kinematic Motion Model

For a terrain-following mobile robot, the only controllable angular velocity is ω_z , so we will assume that $(\omega_x, \omega_y) = (0, 0)$. Likewise, a body-frame linear velocity aligned with the x-axis of the robot (v_x) will be assumed to be the only controllable linear velocity, so we will assume that $(v_y, v_z) = (0, 0)$. Under these assumptions, the attitude and altitude are determined by the suspension kinematic model and the differential equation which governs the pose $\underline{P} = (x, y, \psi)$ simplifies to:

$$\underline{\dot{P}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(\psi)\cos(\theta)v_x \\ \sin(\psi)\cos(\theta)v_x \\ \frac{\cos(\phi)}{\cos(\theta)}\omega_z \end{bmatrix} \quad (13)$$

2.4 Trajectory Kinematics

The forward model of trajectory generation is generated by integrating the kinetic motion model (the world-frame velocities (\dot{R}) and robot-frame orientation rates ($\dot{\Psi}$)):

$$\underline{X} = \int f(\underline{V}, \underline{\Omega}, \underline{X}) dt \quad (14)$$

3 Trajectory Generation Algorithm

This section describes an algorithm to solve the two-point boundary value problem generating the control inputs consistent with a desired terminal state.

3.1 Inverting the Trajectory Kinematics

The problem of trajectory generation is that of determining a set of controls that will satisfy a set of posture constraints. A general method for solving the coupled, nonlinear equations in equation (13) is presented in Figure 2. This is a method which will find the nearest local solution. It does not converge in general to the global optimum but we have not had any real difficulty with incorrect local minima.

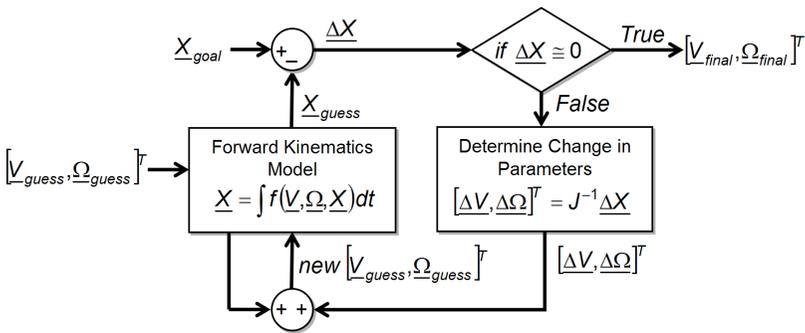


Fig. 2. Inverse Trajectory Generation Solver: From an initial guess of controls $(\underline{V}, \underline{\omega})$, the forward trajectory is evaluated and the control is adjusted based on the product of the system Jacobian and the constraint errors $(\underline{\Delta X})$. This iterative method continues until the terminal conditions are reached.

In this approach, an initial guess of velocity controls $\underline{V}_{guess}, \underline{\Omega}_{guess}$ is modified based on the product of the system Jacobian and the constraint errors $(\underline{\Delta X})$ until the termination requirements are reached. As in [1], partial derivatives are computed numerically using equation (4).

3.2 Control Primitives

The trajectory generation problem reduces to that of finding a set of controls $(v_x(t), \omega_z(t))$ which satisfy the seven constraints $(x_f, y_f, \psi_f, v_{x_0}, v_{x_f}, \omega_{z_0}, \omega_{z_f})$. The terminal pose is described as the relative pose of the terminal and initial postures. A third-order polynomial spiral and a linear velocity profile provide enough degrees of freedom to satisfy the seven constraints:

$$\omega_z(t) = \omega_{z_0} + at + bt^2 + ct^3 \tag{15}$$

$$v_x(t) = v_{x_0} + dt \tag{16}$$

Notice that, for systems that do not model actuator delays or wheel slip, three of the constraints $(v_{x_0}, v_{x_f}, \omega_{z_0})$ are automatically satisfied using this set of control primitives (since $d = \frac{v_{x_f} - v_{x_0}}{t_f}$). For systems that do incorporate such dynamics, the right-pseudo inverse $(J^T(JJ^T)^{-1})$ can be employed in the inverse trajectory generation solver to find for five variables (a, b, c, d, t) that satisfy the four constraints $(x_f, y_f, \psi_f, \omega_f)$. The right-pseudo inverse can also be employed to solve for controls which are higher-order polynomial spirals than the ones in equations (15) and (16).

4 Application to the Rocky 7 Rover

This section demonstrates how the general methods discussed in section 2 can be applied to a specific model. The Rocky 7 prototype Mars rover (Figure 3) employs a rocker-bogie suspension to provide contact for all six wheels with most terrains.

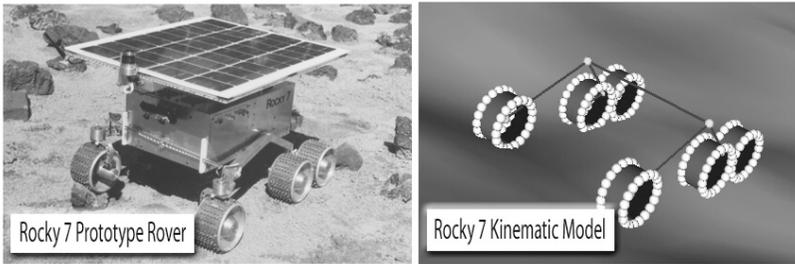


Fig. 3. Rocky 7 Prototype Mars Rover Kinematic Model: On the left is a picture of the Rocky 7 rover, a six-wheeled robot which has a rocker-bogie suspension to navigate complex terrain while maintaining wheel contact. On the right is a representation of our kinematic model of the suspension, which has the three passive joints (ρ, β_1, β_2) .

Figure 3 shows how our kinematic model compares to Rocky 7. It was based on the work done in [6], but computed the forward kinematics with respect to the world frame for our suspension model solver. Rocky 7 employs a rocker-bogie suspension that has three degrees of freedom (ρ, β_1, β_2) which correspond to the major and two minor rocker-bogie angles respectively.

4.1 Rocky 7 Rate Kinematics, Actuator Dynamics, and Wheel Slip

The Rocky 7 rover has eight actuators: a steering actuator on each of the front two wheels and six drivable wheels. In order to get a realistic model of actuator dynamics, a first-order lag was assumed.

$$v_{c_{out}}(t) = \frac{\alpha_1 dt}{\tau} [v_{c_{in}}(t) - v_{c_{out}}(t - \tau)] + v_{c_{out}}(t - \tau) \quad (17)$$

$$\psi_{1,2_{out}}(t) = \frac{\alpha_2 dt}{\tau} [\psi_{1,2_{in}}(t) - \psi_{1,2_{out}}(t - \tau)] + \psi_{1,2_{out}}(t - \tau) \quad (18)$$

The wheel slip model assumes that only a fraction of the requested velocity is achieved. Inverse rate kinematics generate the achieved V_{out}, Ω_{out} due to the actuator dynamics and wheel slip model.

4.2 Rocky 7 Suspension Kinematics

Just as in the general solution, the suspension model provides a mapping of the linear and angular velocities estimated by the actuator dynamics and wheel slip models from the body frame to the world frame, except now the suspension model is known. Taking advantage of the fact that there are six controls (attitude (ϕ, θ) , altitude (z) , and the three rocker-bogie angles (ρ, β_1, β_2)) and six constraints $(z_{c_1} - z_{c_6})$, the same numerical method used in the inverse trajectory generation solver is applied. An error vector $\underline{\Delta z} = [\Delta z_{c_1}, \dots, \Delta z_{c_6}]^T$ is formed from the difference between the elevation of the terrain and the contact point of each of the six wheels. The initial guess of control is adjusted by the product of the inverse Jacobian of the forward kinematics of the contact points with respect to the world frame and the elevation error vector until the terminal conditions are met.

4.3 Rocky 7 Kinetic Motion Model

Since the Rocky 7 rover is a terrain-following mobile robot, the kinetic model follows the form of equation (13). The attitude and altitude are determined from the suspension model for a given pose.

4.4 Rocky 7 Trajectory Kinematics

The forward model of trajectory generation for the Rocky 7 rover is generated by integrating equation (13).

5 Implementation

5.1 Forward Solution of Trajectory Kinematics

In order to determine the terminal pose (x_f, y_f, ψ_f) of the rover, the system dynamics described in equations (14)-(16) must be integrated with respect to time. These state equations are coupled and nonlinear. We have found simple Euler integration to be sufficient:

$$x_{t+\Delta t} = x_t + v_{x_{out_{t+\Delta t}}} \cos(\theta(x_t, y_t, \psi_t)) \cos(\psi_t) \Delta t \quad (19)$$

$$y_{t+\Delta t} = y_t + v_{x_{out_{t+\Delta t}}} \cos(\theta(x_t, y_t, \psi_t)) \sin(\psi_t) \Delta t \quad (20)$$

$$\psi_{t+\Delta t} = \psi_t + \frac{\cos(\phi(x_t, y_t, \psi_t))}{\cos(\theta(x_t, y_t, \psi_t))} \omega_{z_{out_{t+\Delta t}}} \Delta t \quad (21)$$

$$\omega_{z_{out_{t+\Delta t}}} = f(\omega_{z_{in_{t+\Delta t}}}) \quad (22)$$

$$v_{x_{out_{t+\Delta t}}} = g(v_{x_{in_{t+\Delta t}}}) \quad (23)$$

Notice that the output linear and angular velocities of the rate kinematics are used in this model. In this method, a configuration at each new pose must be found. The algorithm can be sped up dramatically by using the previous configuration at time $t - \Delta t$ as the seed for the configuration at the current time t .

It is essential that Δt be small enough to accurately model the integral of the system dynamics and capture the terrain along the path. For example, in trajectory generation over the scale of a few rover lengths, 12-15 iterations may be enough to accurately model the system dynamics but these may still miss small terrain disturbances that may influence the path enough to matter.

5.2 Trajectory Generation Using Inverse Trajectory Kinematics

Utilizing the forward model of trajectory kinematics developed in the previous section, along with the controls and methods of section 3.1 and 3.2, an inverse trajectory kinematics solver which accounts for both rough terrain and actuator dynamics is obtained.

5.3 Initialization/Termination

Since the numerical method used in this work does not guarantee global convergence, a heuristic which places the terminal posture of the initial guess near the goal posture is required. It was found that the solution to the two-dimensional trajectory generation problem places the terminal posture of the initial guess within 15% of the goal on a variety of interesting terrains. This is close enough to ensure convergence in most applications. In the case of

an exceptional terrain disturbance which incurs a large terminal posture perturbation, line searches or scaling of the change of parameters (Δp) can be implemented to prevent overshoot and divergence from the solution.

The set of termination conditions used for the trajectory generation algorithm were similar to those in [1], stopping iterations when $[\Delta x_f, \Delta y_f, \Delta \psi_f, \Delta \omega_f]^T = [0.001m, 0.001m, 0.01rad, 0.01 \frac{rad}{sec}]^T$. Likewise, the suspension model required millimeter residuals in Δz_{c_i} .

6 Results

6.1 Rough Terrain Trajectory Generation Example

This section demonstrates the need of rough terrain trajectory generation by examining an example situation. In this example, the Rocky 7 platform is asked to find a continuous trajectory between two postures as seen in Figure 4. The relative terminal posture $[x_f, y_f, \psi_f, \omega_f]^T$ is equal to $[3.0m, 5.0m, \frac{\pi}{2.0}rad, 0.0 \frac{rad}{sec}]^T$. In order to isolate the effect of neglecting the influence of attitude in the trajectory generator, rate kinematics are ignored in this example.

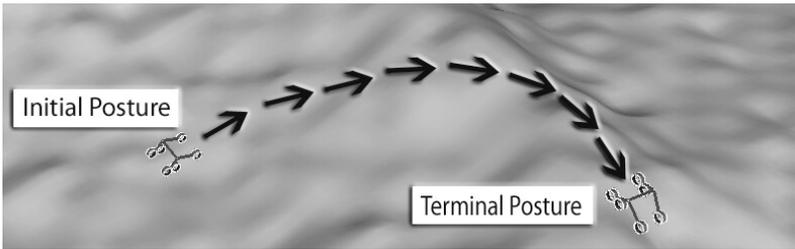


Fig. 4. Example Rough Trajectory Generation: This figure shows an example trajectory generation problem, where a continuous path is desired between an initial posture inside a crater and the final posture just over the lip of the crater.

First, the two-dimensional continuous curvature path is solved to millimeter accuracy and fed into the three-dimensional forward solution. The two-dimensional solution incurs a terminal position error of 6.2% (45.3cm) of the entire arclength of the solution. The three-dimensional trajectory generator finds a new path that is continuous in angular velocity, with an initial and final velocity of 1 m/sec, for millimeter accuracy in only 3 iterations. Figure 5 shows the difference between the two-dimensional system model and three-dimensional system model paths and how the latter generates the correct trajectory.

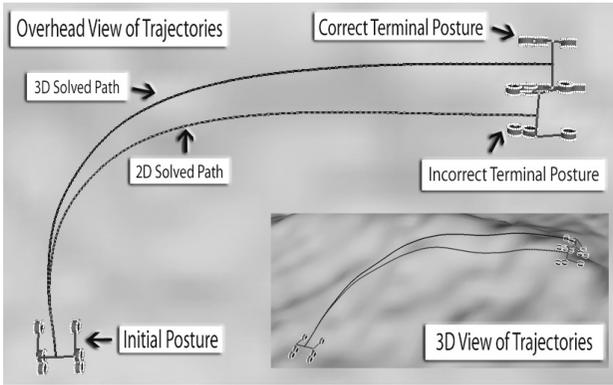


Fig. 5. Trajectory Generation Solution: This figure shows the result of neglecting attitude in the forward model (two-dimensional solution) and the new solution based on a three-dimensional system model. By neglecting attitude, the terminal position is off by 6.2% when compared to the total arclength of the solution.

6.2 General Results

To evaluate the need, performance, and behavior of this algorithm, several thousand tests were run to understand rates of convergence and range of errors to expect. One behavior that was recognized is that even though error would increase dramatically with rougher terrain, the number of iterations required to meet the termination conditions did not. The numerical method that we are using attempts to remove all error in a single iteration, so this behavior suggests that the first order approximation is a good one.

7 Conclusions

In the context of mobile robots which must already expend significant effort to understand terrain complexity, the use of a flat terrain assumption in trajectory generation is difficult to justify. However, as the paper has shown, the use of terrain information requires a certain amount of effort to develop a more complex generation algorithm. While space was not available to address a computation comparison, the additional computation for 3D models is not a significant factor in practice.

A very general algorithm has been presented which can generate continuous paths for mobile robots obliged to drive over rough terrain while subject to additional nonidealities such as wheel slip and actuator delays. The essential problem is to invert a model of how parameterized control inputs, terrain shape, terrain interaction and actuator dynamic models determine the terminal state of a vehicle at all future times. A numerical technique was adopted due to the assumed inability to express terrain shape in closed form. However, once a numerical approach is adopted, it also means that any forward model

can be inverted to produce continuous controls subject only to the capacity of the numerical linearization to converge. In principle, a full Lagrangian dynamics model can be inverted using our technique, for example.

The Rocky 7 prototype rover was used to illustrate the application of the general models of suspension and rate kinematics to a specific robot. For any vehicle, only forward rate kinematics and forward suspension models are needed to use the rest of the algorithm. Our results suggest there is much to gain and little to lose by moving to fully 3D models. Such predictive models lead to improved performance by removing as much model error as possible at planning time — so that path following controls are used only to compensate for truly unpredictable disturbances.

While the algorithm has been presented in the context of planning computations, it promises to be equally valuable for the generation of corrective trajectories in feedback path following controls. Future work will assess the value of the algorithms for this purpose in the hope of developing short term path followers which maximally exploit the model and terrain information which can be assumed to be present in most present and future mobile robots.

Acknowledgment

This research was conducted at the Robotics Institute of CMU under contract to NASA/JPL as part of the Mars Technology Program.

References

1. Nagy, B., Kelly, A., "Trajectory Generation for Car-Like Robots Using Cubic Curvature Polynomials", Field and Service Robots 2001 (FSR 01), Helsinki, Finland - June 11, 2001.
2. Simeon, T., Dacre-Wright, B., "A Practical Motion Planner for All-terrain Mobile Robots". 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems, Yokohama, Japan - July 26-30, 1993.
3. Iagnemma, K., Shibly, H., Rzepniewski, A., Dubowsky, S., "Planning and Control Algorithms for Enhanced Rough-Terrain Rover Mobility". International Symposium on Artificial Intelligence and Robotics and Automation in Space 2001 (i-SAIRAS 01), St-Hubert, Quebec, Canada - June 18-22, 2001.
4. H Delingette, M. Herbert, and K Ikeuchi, "Trajectory Generation with Curvature Constraint based on Energy Minimization", Proc, IROS, pp 206-211, Osaka, Japan, 1991.
5. Shin, D.H., Singh, S., "Path Generation for Robot Vehicles Using Composite Clothoid Segments" tech. report CMU-RI-TR-90-31, Robotics Institute, Carnegie Mellon University, December, 1990.
6. Tarokh, M., McDermott, G., Hung, J. "Kinematics and Control of Rocky 7 Mars Rover." Tech Report. August 1998.
7. Volpe, R., "Navigation Results from Desert Field Tests of the Rocky 7 Mars Rover Prototype." International Journal of Robotics Research, 1998.

Results in Combined Route Traversal and Collision Avoidance

Stephan Roth, Bradley Hamner, Sanjiv Singh, and Myung Hwangbo

Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA.
{stephan+@cs.cmu.edu, bhamner@andrew.cmu.edu, ssingh@ri.cmu.edu,
myung@andrew.cmu.edu}

Summary. This paper presents an outdoor mobile robot capable of high-speed navigation in outdoor environments. Here we consider the problem of a robot that has to follow a designated path at high speeds over undulating terrain. It must also be perceptive and agile enough to avoid small obstacles. Collision avoidance is a key problem and it is necessary to use sensing modalities that are able to operate robustly in a wide variety of conditions. We report on the sensing and control necessary for this application and the results obtained to date.

Keywords: Outdoor mobile robot, path following, collision avoidance

1 Introduction

While the use of mobile robots in indoor environments is becoming common, the outdoors still present challenges beyond the state of the art. This is because the environment (weather, terrain, lighting conditions) can pose serious issues in perception and control. Additionally, while indoor environments can be instrumented to provide positioning, this is generally not possible outdoors at large scale. Even GPS signals are degraded in the presence of vegetation, built structures and terrain features. In the most general version of the problem, a robot is given coarsely specified via points and must find its way to the goal using its own sensors and any priori information over natural terrain. Such scenarios, relevant in planetary exploration and military reconnaissance are the most challenging because of the many hazards – side slopes, negative obstacles and obstacles hidden under vegetation – that must be detected. A variant of this problem is for a robot to follow a path that is nominally clear of obstacles but not guaranteed to be so. Such a case is necessary for outdoor patrolling applications where a mobile robot must travel over potentially great distances without relying on structure such as beacons and lane markings. In addition to avoiding obstacles, it is important that the vehicle stay on the designated route as much as possible.

Perception is typically the bottleneck in outdoor navigation, especially at speeds higher than a few meters/sec. This is primarily because perception of small obstacles (as small as 15 cm high) at or beyond the stopping distance ahead of the robot is typically only possible using laser ranging. Laser ranging produces detailed shape of the terrain but is limited in sampling and scanning speed.

Here we report on the perception and guidance that we have developed for an outdoor patrolling robot (Figure 1) that uses two low-cost laser scanners to develop an understanding of the world around it. In specific, we report on methods of obstacle detection and collision avoidance for this robot while it travels at speeds at up to 5 m/s.



Fig. 1. Grizzly is a navigation test-bed built upon a commercially available All Terrain Vehicle (ATV). It uses two laser scanners to perceive the shape of the world. The vehicle is equipped with differential GPS and a six-axis inertial measurement unit that provides accurate attitude.

2 Related Work

There has been a great deal of attention paid to parts of the problem of autonomous operation in semi-structured environments such as in ports [6], underground mines [9], and highways [3]. In some of these cases, the environment can be controlled enough that obstacle detection can be simplified to ensuring that the machines are capable of stopping for people or vehicle sized obstacles. Autonomous machines operating in natural environments, however, must be able to detect several different types of obstacles including side slopes and negative obstacles. This is accomplished by using sensors that can determine the shape of the world around them. Stereo vision [11], color segmentation [1], radar [8] and laser range finders [5] have all been used for obstacle detection. Unfortunately, passive vision suffers from lighting, color constancy, and

dynamic range effects that cause false positives and false negatives. Radar is good for large obstacles, but localization is an issue due to wide beam widths. Single axis laser scanners only provide information in one direction, and can be confounded by unmeasured pitching motion and mis-registration. Two axis scanners are also used, which provide more information, but are very costly.

Several systems have demonstrated off road navigation. The Demo III XUV drives off-road and reaches speeds up to 10 meters per second. The speeds are high, but the testing environments are rolling meadows with few obstacles. Obstacles are given a clearance which is wider than the clearance afforded by extreme routes. When clearance is not available, the algorithm plans slower speeds [5]. Sandstorm, a robot developed for desert racing, has driven extreme routes at speeds up to 22 meters per second, but makes an assumption that it is traveling on slowly varying roads. If an obstacle is encountered in the center of a road, the path cannot change rapidly enough to prevent collision [4].

Our work is related to several previous research themes. The first connection is to the research in autonomous mobile robots for exploration in planetary environments [10][11] that uses traversability analysis to find obstacles that a vehicle could encounter. The second connection is to a method of scanning the environment by sweeping a single-axis laser scanner [2] that allows detection of obstacles even when the vehicle is translating and pitching. A third connection is to a method of collision avoidance that is based on models of human navigation in between discrete obstacles [7].

3 Approach

Here we discuss the two main parts of our approach – obstacle detection and collision avoidance.

3.1 Obstacle Detection

For high speed navigation, the sensors required depend on the vehicle's speed, stopping distance and minimum obstacle size. At higher speeds, where stopping distances are greater, the obstacles must be detected at a greater distance. In order to detect smaller obstacles, the measurement density of the sensor must be correspondingly greater. Our goal is to enable the vehicle to travel at speeds of up to 5 m/s while detecting obstacles as small as 20cm × 20cm. In other work with lower speed vehicles moving at 2 m/s [2], we find that a single sweeping laser is sufficient for detecting obstacles. The sweeping laser system consists of a single Sick laser turned so it is scanning a vertical plane. A motor mechanically sweeps the vertical plane back and forth, thus building a 3-D map of the terrain in front of the vehicle. However, for the higher speed obstacle detection in this application, we find that the sweeping laser alone cannot provide a sufficient density of laser measurements to detect

small obstacles at higher speeds. Accordingly, a second fixed laser is deemed necessary (Figure 2).

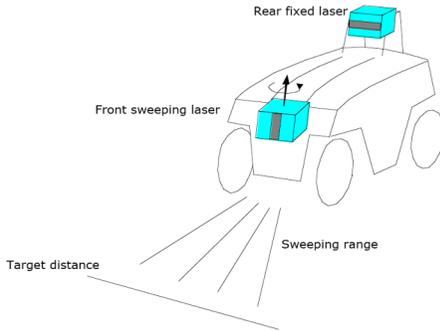


Fig. 2. Configuration of lasers scanners on the vehicle. The fixed laser concentrates its scans 10m in front of the vehicle, giving an early detection system. The sweeping laser concentrates its data closer to the vehicle, giving the ability to track obstacles that are closer to the vehicle.

The addition of a second fixed laser provides several advantages over the single sweeping laser. Primarily, the fixed laser is pointed 10m in front of the vehicle and increases the density of laser data at points far from the front of the vehicle. Now smaller obstacles are detected at a distance sufficient for safe avoidance. The sweeping laser system concentrates its data closer to the vehicle, so obstacles nearer the vehicle are tracked. A second advantage of the two laser system is that they collect orthogonal sets of data. The sweeping laser is best suited for detecting pitch type obstacles, while the fixed laser is best suited for detecting roll type obstacles. The two laser systems complement each other by performing best for these two different types of obstacles.

The addition of a second laser by itself is not enough to guarantee detecting obstacles in all cases. When following curved paths, we find it is not enough to simply sweep the laser in a fixed range. It is necessary to bias the sweeping laser so it points into turns. Figure 4 shows a representation of the number of laser hits that would be received by a $15\text{cm} \times 15\text{cm}$ obstacle located a distance greater than the vehicle's stopping distance from the front of the vehicle. Areas of red indicate a high number (>60) of hits, and areas of blue indicate a lower number (10-20). The first picture shows the number of hits when the laser is swept between a fixed 20 degree range centered about the front of the vehicle.

It is clear from the figure that there is sufficient laser data to detect obstacles along the straight section. However, along the turn the number of hits decreases dramatically. The lower density of laser data increases the chances that an obstacle will not be detected while the vehicle is turning. Figure 4(b)

shows the number of hits when the sweeping laser is biased to point into the turn. Compared to the unbiased case, the number of laser hits on the obstacle greatly increases in the area where the vehicle is turning.

With data from two lasers, we use two obstacle detection algorithms: a traversability analysis and a line scan gradient analysis. In the traversability analysis, data from both lasers is used to produce a point cloud of the terrain in front of the vehicle. Vehicle-sized planar patches are fit to the point cloud data, and the fitted data gives three measures useful in identifying obstacles: plane orientation (roll, pitch), roughness (the residual of the fit) and the height of data points above the plane. These measured values are used to classify areas as *untraversable* or *clear*. While the traversability analysis is a simple way of detecting obstacles, it can produce false positives due to inaccurate calibration of the two lasers and/or incorrect synchronization with positioning. To supplement the traversability analysis, the slope of segments of individual line scans from the sweeping laser is also calculated as in [2]. If the slope of a scan segment is above a given threshold, it is tagged as a *gradient* obstacle. Because the gradient analysis uses piecewise segments of an individual line scan, it is not susceptible to misregistration as the traversability analysis can be.

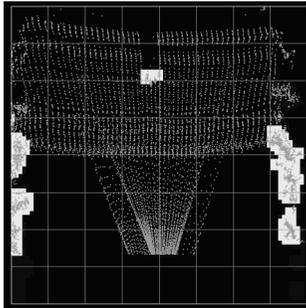


Fig. 3. Overhead view of laser data from from the two scanners. Data over a window of time are registered to a common reference frame and obstacles are found by analyzing traversability and gradient of the individual line scans.

To classify an object as a true obstacle, both the gradient and traversability analyses must agree. The combination of the two obstacle detection algorithms compensates for the weaknesses of the two individual algorithms and dramatically reduces the false obstacle detection rate. Because the gradient analysis looks at only an individual line scan from the sweeping laser, it cannot take advantage of integrating multiple scans over time like the traversability analysis can. However, by only using single line scans, the gradient analysis is relatively immune to mis-registration problems that plague the traversability analysis.

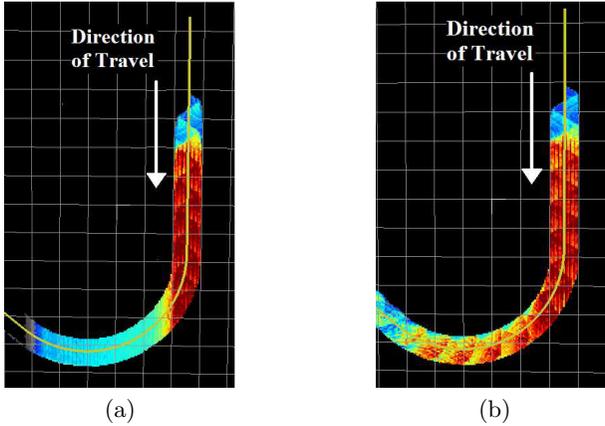


Fig. 4. Grid representation of laser hits by both the fixed and sweeping lasers on a $15\text{cm} \times 15\text{cm}$ obstacle when sweeping with and without biased laser at 4m/s . (a) shows a representation of the number of hits without biasing the laser when going around turns. (b) shows the number of hits when biasing the laser. Areas of blue indicate a low number of laser hits (10-20). Red areas indicate a high number of hits (>60). Biasing the laser when going around turns increases the laser hit density.

3.2 Collision Avoidance

The goal of our collision avoidance system is to follow a path and avoid obstacles along the way. When an obstacle is detected in front of the vehicle, the vehicle should swerve to avoid it and return to the path in a reasonable fashion. If there are multiple obstacles on the path, the vehicle must navigate between them. Sometimes an obstacle may block the entire path. In this case, the vehicle must stop before colliding with it. An ideal collision avoidance algorithm would accept a map of hazards and determine steering and speed to navigate in between these. Since this algorithm must run many times a second, ideally it would have low computational complexity.

Fajen and Warren report a reactive method of collision avoidance based on experiments to determine how humans avoid obstacles [7]. The method uses the positions of a goal point and obstacle points relative to the current vehicle position to derive an instantaneous steering angle. We developed a path-following obstacle avoidance algorithm that extends this method. Since the vehicle simply avoids obstacles without planning a full path, we call the algorithm *Dodger*.

Consider the vehicle and a desired goal point. If the goal is at a large angle to the current vehicle heading, as in Figure 5(a), then the vehicle must steer sharply. Smaller angular differences, as in Figure 5(b), mean that the vehicle does not have to steer as hard. Similarly, for greater distances to the goal, as in Figure 5(c), slight steering is sufficient. Based on these principles, Fajen and Warren develop a goal attraction function,

$$f_a(\psi_g, d_g) = \psi_g(e^{-c_g d_g} + c_s)$$

where d_g is the translational distance to the goal, ψ_g is the angular distance to the goal, c_g is a goal distance decay constant, and c_s is a scale constant to assure the goal attraction is never zero.

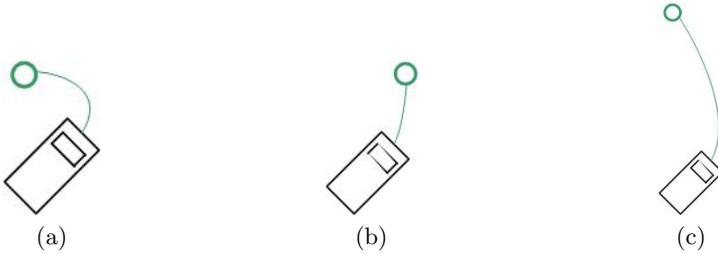


Fig. 5. Three scenarios involving driving to a goal, indicated by the green circles. The vehicle must steer proportionally to the distance and angle to the goal.

Repulsion from obstacles uses similar logic. When an obstacle is at a large angular distance, as in Figure 6(a), the vehicle does not need to turn sharply to avoid it. When the obstacle is far from the vehicle, as in Figure 6(b), a small steering angle is sufficient. The vehicle must steer sharply only when the obstacle is close and in front of the vehicle, as in Figure 6(c). These principles can be combined into a single obstacle repulsion function,

$$f_r(\psi_o, d_o) = \psi_o(e^{-c_{o1}|\psi_o|})(e^{-c_{o2}d_o})$$

where d_o is the translational distance to the obstacle, ψ_o is the angle to the obstacle, c_{o2} is a distance decay constant, and c_{o1} is an angular decay constant.

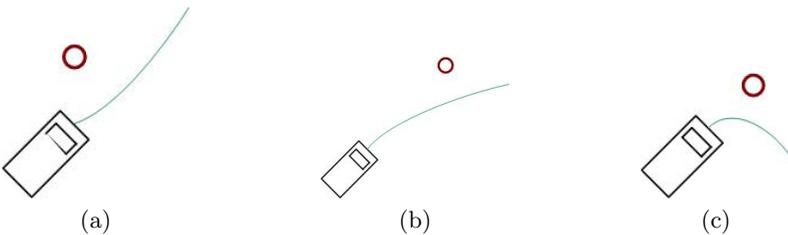


Fig. 6. Three scenarios involving avoiding obstacles, represented by the red circles. The vehicle must steer proportionally to the distance and angle to obstacles.

This function is applied to every obstacle, and the result is summed together. Note that this treats obstacles as individual points. To represent real

obstacles, we discretize them into collections of points spaced ten centimeters apart (Figure 7).

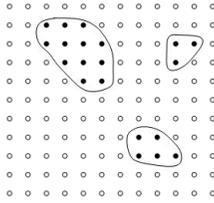


Fig. 7. We represent obstacles as collections of points spaced ten centimeters apart. The obstacle repulsion function is applied to each black obstacle point individually.

The goal attraction and obstacle repulsion are combined to get the control equation:

$$\dot{\phi}^* = -k_g f_a(\psi_g, d_g) + k_o \sum_{o \in O} f_r(\psi_o, d_o)$$

where k_g and k_o are relative weighting constants and $\dot{\phi}^*$ is the commanded steering velocity.

We have extended the original formulation by Fajen and Warren in several ways. First, the original obstacle repulsion function is multiplied by the angle to the obstacle. This means that if the vehicle is headed straight towards an obstacle, the angular repulsion term is zero. The theory is that the vehicle will turn slightly away from the obstacle at first (crossing in front if necessary), the angle will increase, and eventually the vehicle will fully turn away from the obstacle. However, at high speeds, there may not be enough time for that to happen. We modify the function to have high repulsion at small angles, and accept the consequences of getting into local minima more easily. The new obstacle repulsion function becomes

$$f_r(\psi_o, d_o) = \text{sign}(\psi_o)(e^{-c_{o1}|\psi_o|})(e^{-c_{o2}d_o})$$

Another problem occurs in areas of dense obstacles, such as the path illustrated in Figure 8(a). Here, there are obstacles everywhere in front of the vehicle. The leftward repulsion of the obstacles on the right side of the path may be greater than the rightward repulsion of the single obstacle on the path. Were it not for our speed control (see below), the vehicle would collide with the obstacle on the path. The problem is that the base system does not use all of the available information. The obstacles are directly in front of the vehicle, and therefore look threatening, but the path curves away from them. Similarly, the single obstacle may be at a large angular distance, but it is directly between the vehicle and the goal point. We introduce a new term to the obstacle repulsion function, which considers whether the obstacle is blocking the goal,

$$\text{dist}(v, g, o) = \frac{|(g_x - v_x)(v_y - o_y) - (v_x - o_x)(g_y - v_y)|}{\|g - v\|}$$

$$f_r(\psi_o, d_o, d_{vgo}) = \text{sign}(\psi_o)(e^{-c_{o1}|\psi_o|})(e^{-c_{o2}d_o})(1 + c_{o3}(d_{max} - \max(d_{max}, d_{vgo}))^2)$$

where d_{vgo} is the perpendicular distance from the obstacle to the vector between the vehicle and the goal calculated by $\text{dist}(v, g, o)$, and d_{max} is some maximum distance from that vector. The obstacles to the right are far away from the goal vector, so their repulsion is the same as before. However, now the single obstacle has greater repulsion, assuring that the vehicle will not drive towards it (Figure 8(b)).

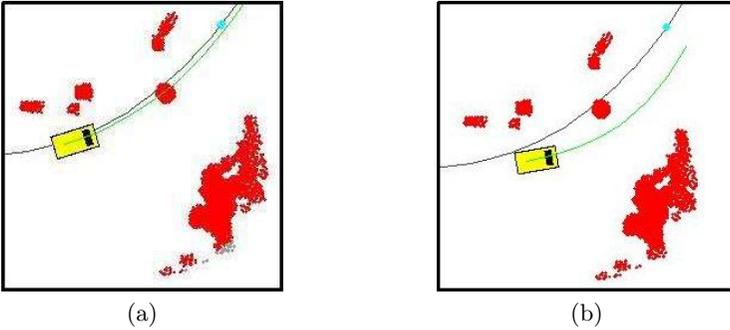


Fig. 8. The dark line is the desired path. The lighter line represents the vehicle's future path when using the Dodger algorithm. The dot on the desired path is the goal point used by Dodger. In (a), without using the goal vector term, the obstacles on the right side of the desired path collectively have a much larger repulsion than the single obstacle that is actually on the path. That problem is corrected in (b), where the goal vector term greatly increases the repulsion by the single obstacle.

Following a path using Dodger is done by first finding the point on the path closest to the vehicle. The goal point is set to a point some distance down the path. When an obstacle appears in front of the vehicle, this distance is increased so as to allow the vehicle to maneuver around the obstacle. Fajen and Warren's experiments showed that humans consistently kept the same speeds as they traveled. However, when obstacles appear, we would like the vehicle to slow down, to allow for greater possible steering angles, and thus greater maneuverability. This is a simple proportional function based on the largest obstacle repulsion. If the largest obstacle score is high enough, that is, if there is an obstacle directly in front of the vehicle, then we stop the vehicle before a collision.

Speed control is also done by predicting the course that Dodger would take in the future. Using Dodger's output steering angle and speed, we run a forward integration of the vehicle model interleaved with the steering control, to predict where the vehicle will be a short amount of time later. We build a

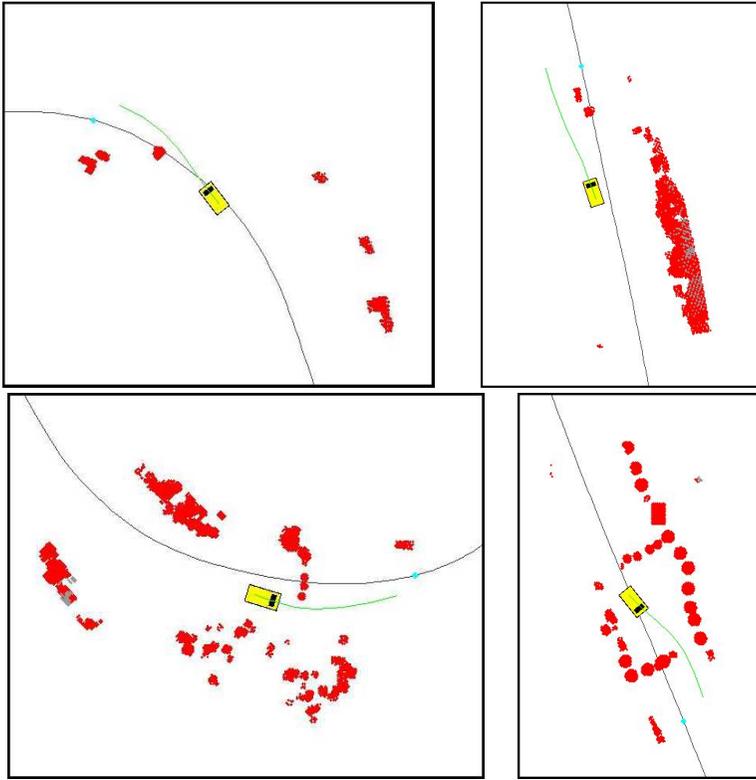


Fig. 9. In these situations, Dodger safely guides the vehicle around the detected obstacles.

path from these predictions over four seconds (shown as the light line extending forward from the vehicle in the figures). This predicted path accounts for curvature limits based on the vehicle's speed. Then we check along the path for collisions. If there is a collision along the path, then we can slow the vehicle immediately, rather than waiting until it gets closer to the obstacle. Again, the slow-down allows the vehicle more maneuverability and a greater chance of the collision being avoided. Dodger works well for avoiding single obstacles, some situations with multiple obstacles, including slaloms, on straight-aways, and around corners (shown in Figure 9).

However, there are specific situations in which Dodger does not find a path around the obstacle, and the vehicle is forced to stop. When the obstacle is wide, there are points on both sides of the vehicle which counteract each other, so the vehicle never gets all the way around the obstacle (Figure 10(a)). Also, when there is an obstacle around a corner, Dodger prefers to go outside the turn around the obstacle, rather than inside. This is because the obstacle points on the inside of the turn are closer to the goal vector, and therefore have

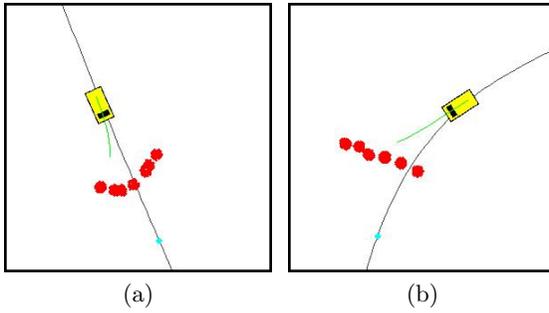


Fig. 10. In (a), due to the curved shape and width of the obstacle, some of the rightward repulsion is cancelled out by a leftward repulsion. Then Dodger does not find a way all the way around the obstacle, and stops before a collision. In (b), there is enough room to avoid this obstacle to the left. However, the obstacle points closer to the goal vector exhibit a larger rightward repulsion. The obstacle is too wide for the vehicle to avoid around the outside, so Dodger stops the vehicle before collision.

more repulsion. This causes a problem when the obstacle covers the outside of the corner (Figure 10(b)).

Using the predicted path, the system can detect situations in which Dodger fails to direct the vehicle around the obstacle. When the predicted path stops in front of an obstacle, the system invokes a planning algorithm, like D^* , to get a new goal point which will help Dodger around the obstacles. First, the planning algorithm constructs a small map of the area in the vicinity of the vehicle (Figure 11(a)). The goal location passed to D^* is Dodger's goal point. Next, the planning algorithm constructs an optimal path around the obstacles to that goal location. The system then starts at the goal point and walks backwards along the optimal path, stopping when there are no obstacles on a straight line to the vehicle. This unblocked position is selected as a new goal for Dodger, and the Dodger algorithm is run again. The new goal point is closer than the old one, and is off to one side of the problem obstacles, so it has more influence than the original goal point. When Dodger is run again, the new goal point pulls the vehicle to one side of the obstacles. In essence, the planning algorithm chooses a side for Dodger to avoid on. The system continues this hybrid method until Dodger, using its normal goal point, gives a predicted path that safely avoids the obstacles (11(c)). The D^* augmentation to Dodger is especially useful in complex obstacle configurations, as shown in Figure 12. Running Dodger with the planning algorithm takes more computation time, so to be safe, we also slow the vehicle down when the planning algorithm is running.

In both of the above cases, we can detect the impending collision and stop the vehicle in time. However, there are some cases in which Dodger would exhibit undesirable behavior while not actually colliding with an obstacle. For

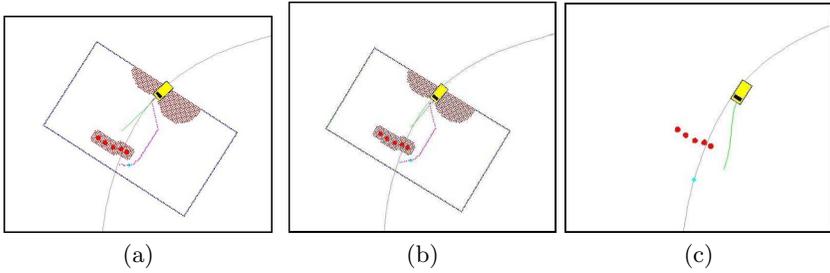


Fig. 11. In (a), the system predicts a collision and invokes D^* . The map covers only a small area between the vehicle and the original goal point. Obstacles are added to the map, and points within the vehicle's minimum turning radius are also marked as untraversable. The optimal path from D^* goes around the obstacle, and the furthest visible point along the D^* path is set as the new goal point. Dodger is run again using this goal. In (b), the new D^* goal point has pulled the vehicle a little to the left, but not far enough yet, since the system still predicts collision. D^* continues to be invoked. In (c), the vehicle is far enough to the left that the system no longer predicts a collision if the regular goal point is used with Dodger, so D^* is no longer necessary.

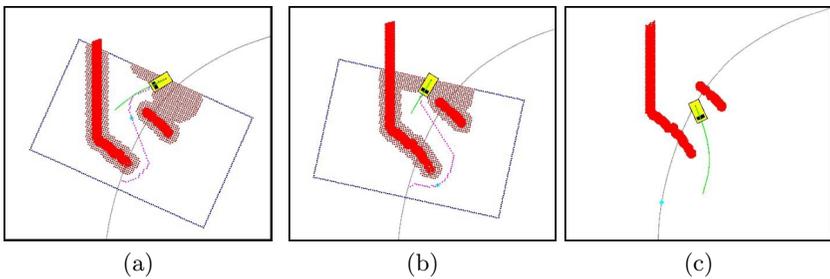


Fig. 12. The D^* augmentation to Dodger can also lead the vehicle through complex configurations of obstacles. In (a), Dodger finds no way around the wall of obstacles, so D^* is invoked. In (b), the goal obtained from the D^* path pulls the vehicle to the left. In (c), Dodger alone can navigate the vehicle past the remaining obstacles.

example, Figure 13(a) shows a case where obstacles on both sides of the path are actually to the left of the vehicle and repel the vehicle off the desired path around the obstacles, even though the desired path is clear. To prevent the vehicle from unnecessarily diverging from the desired path, we use a "ribbon" method. We construct a ribbon of fixed distances down the path and to either side. If there are no obstacles on this ribbon and the vehicle is currently within the ribbon, then we zero any obstacle repulsion. The result is a steering angle entirely based on the goal attraction, and the vehicle successfully tracks the path (Figure 13(b)).

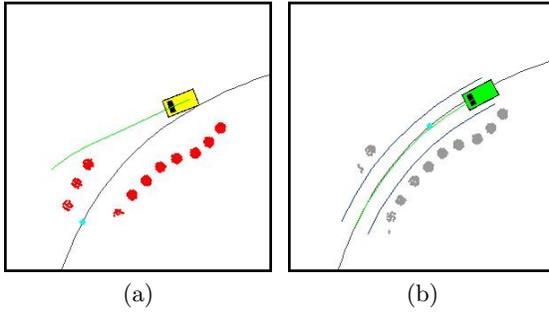


Fig. 13. In (a), the obstacles on both sides of the path repel the vehicle rightward. As a result, the vehicle leaves the path, even though there is no obstacle on the path. In (b), the ribbon method is being used. The dark lines on either side of the path denote the ribbon. There are no obstacles within the ribbon, so the total obstacle repulsion is set to zero, and the vehicle follows the path.

4 Results

The system presented here is able to perform high speed off road navigation at speeds up to 5m/s. The tightly coupled GPS + IMU localization system provides reliable position estimates in areas with limited GPS availability. The combination of two laser systems, one fixed and the other sweeping, enables us to detect obstacles as small as 30cm high and 30cm wide. The obstacle avoidance algorithm allows us to avoid these obstacles even while traveling at 5m/s. The system described here has successfully performed over 100 km of autonomous travel.

5 Conclusions

We have developed a method of obstacle detection and collision avoidance that is composed of low cost components and has low complexity but is capable of state of the art performance. The advantage of being able to actuate the laser scanning is that it provides for an even distribution of laser range data as the path turns.

So far we have used shape to separate obstacles from clear regions. The next step is to allow for recognition of materials so that vegetation can be appropriately recognized.

References

1. P. H. Batavia. and S. Singh. Obstacle detection using adaptive color segmentation and color homography. In *Proceedings of the International Conference on Robotics and Automation. IEEE*, May 2001.

2. P. H. Batavia and S. Singh. Obstacle detection in smooth, high-curvature terrain. In *Proceedings of the International Conference on Robotics and Automation*, Washington, D.C., 2002.
3. T. Jochem C. Thorpe and D. Pomerleau. The 1997 automated highway free agent demonstration. In *IEEE Conference on Intelligent Transportation Systems*, November 1997.
4. M. Clark T. Galatali J.P. Gonzalez J. Gowdy A. Gutierrez S. Harbaugh M. Johnson-Roberson H. Kato P.L. Koon K. Peterson B.K. Smith S. Spiker E. Tryzelaar C. Urmson, J. Anhalt and W.L. Whittaker. High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004. Technical report.
5. D. Legowik S. A. Murphy D. Coombs, A. Lacaze. Driving autonomously offroad up to 35 km/h. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2000.
6. H.F. Durrant-Whyte. An autonomous guided vehicle for cargo handling applications. *International Journal of Robotics Research*, 15, 1996.
7. B. Fajen and W. Warren. Behavioral dynamics of steering, obstacle avoidance, and route selection. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2), 2003.
8. D. Langer and T. Jochem. Fusing radar and vision for detecting, classifying, and avoiding roadway obstacles. In *Proceedings of the IEEE Symposium on Intelligent Vehicles*, 1996.
9. et al. S. Scheduling. An experiment in autonomous navigation of an underground mining vehicle. In *IEEE Transactions on Robotics and Automation*, 1999.
10. et al. S. Singh. Recent progress in local and global traversability for planetary rovers. In *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2000.
11. C. Urmson and M.B. Dias. Vision based navigation for sun-synchronous exploration. In *Proceedings of the International Conference on Robotics and Automation*, May 2002.

Adaptation to Rough Terrain by Using COF Estimation on a Quadruped Vehicle

Shogo Okamoto¹, Kaoru Konishi², Kenichi Tokuda³, and Satoshi Tadokoro⁴

¹ Graduate School of Information Sciences, Tohoku Univ. 6-6-01 Aramaki Aza Aoba, Aoba-ku, Sendai Japan okamoto@rm.is.tohoku.ac.jp

² Graduate school of Science and Technology, Kobe Univ. 1-1 Rokkodai, Nada, Kobe 657-8501 Japan k_konishi@r.cs.kobe-u.ac.jp

³ Dept. Opto-Mechatronics, Wakayama Univ. 930 Sakaedani, Wakayama-city 640-8510 Japan tokuda@rescue-robot.org

⁴ Graduate School of Information Sciences, Tohoku Univ. 6-6-01, Aramaki Aza Aoba, Aoba-ku, Sendai Japan tadokoro@rescuesystem.org

Summary. Foot groping is one way to evaluate the stability of footholds for legged locomotives on rough terrain. For further acquisition of ground information, we installed active ankles with two active joints on the experimental quadruped vehicle, RoQ2. To compensate the loss of passive adaptation of ankles to terrain, active adaptation using COF estimation is implemented. COF is a center of pressure on a sole and estimated by sole sensor, which consists of four FSRs. Sole sensors for COF can determine the sole plane when adapting to rough terrain. This paper also shows that our new proposition can detect an edge of a beam or a step on the ground without thrusting a foot to the objects.

Keywords: COF, quadruped vehicle, rough terrain, adaptation, RoQ

1 Introduction

1.1 Research Background

Since Great Hanshin-Awaji earthquake affected Kobe and inflicted terrible damage on the urban area on Jan. 17th, 1995, the discipline of rescue robots has become more lively in Japan. We are engaged on development of the rescue robots. Quadruped vehicles have higher capability of climbing over steps and obstacles than other proposed rescue robots, such as crawler-type [1][2] or snake-shaped [3] robots. Hexapedal robots like RHex[4] or Sprawlita[5] achieved a moving velocity of a few body lengths per a second. They introduced one actuator per a leg and compliant elements to its hips or legs and cancel the unevenness of the ground and careful control. One of the most difficult

conditions for quadruped vehicles on rough terrain is obstacles laid on the ground or unsafe and breakable footholds. Overcoming steps or traversing rough terrain have been studied on multi-legged robots but they have been done only on the assumption that the footholds are stable. Disaster spots contains brittle and fragile footholds. Legged robots must support their own weight on a confined ground contact area differently from other types of rescue robots. In case of quadruped robots, weight of them have to be supported with three legs. Evaluating stability of footholds is the inevitable ability and must be installed on a quadruped vehicle for breakable terrain. But the robots shown above have not had it so far.

The stability of the ground or footholds can be evaluated by a rescue person who could see a picture of broken building through some vision systems to some extent. When a human walks on the breakable ground, he must avoid apparently dangerous sites, which can be judged by mainly his eyes and experiences. However, when he must choose a foothold among unassured ones, he should know the stability of them, which could be done by foot groping. Foot groping is actually touching the ground to make sure the foothold is stable enough to give a secure support. This idea and the lack of the robots capable of foot groping leads to our eventual idea of sole sensors based on FSRs, which can estimate the distribution of the ground reaction force and COF(Center of Force) on a sole. 6-axial torque sensor, most common sensory apparatuses for a biped robot's[6], used to estimate a zero moment point does not have spatial resolution. Strain gauges reliable as force perceptrors need an electric amplifier, which needs another power supply equipment. Morph3[7] installs four 3-axial torque sensors on a sole plane to make it possible to detect contact information of a lateral side of its feet. But 3-axial torque sensors are essentially different from FSR.

1.2 Research Trends

Desirable functions for an ankle are assumed to be

- 1) to obtain large ground contact area when walking,
- 2) to obtain large motor torque in a supporting phase and
- 3) to immediately adapt to the terrain.

To satisfy above conditions, there is a need for high power actuator that might cause hardware itself to become heavier, or novel mechanism should be considered. Therefore, generally it is hard to realize active ankles with simply serial link joints, and passive ankles are favored. It is common to design passive ankles with universal joints or spherical ones. Meanwhile passive ankle has the ability of passive adjustment to terrain, as to the enlargement of the contact area or protection of sprain, active ankles are more effective. Some active ankle mechanisms for quadruped vehicles have been proposed and active ankles using parallel links tend to be major trend[8][9], taking place of serial links.

2 Active Ankle Mechanism

Some ground information is required to determine the sole plane for adjustment to rough terrain. Our group developed sole sensors that can estimate COF on a sole and our prior study showed that this sole sensor can detect three kinds of footholds on the experimental quadruped vehicle, RoQ1 whose ankles are passive.[10] We newly developed active ankle mechanism and installed them on RoQ2(Fig.1, Fig.2).



Fig. 1. the experimental quadruped vehicle with active ankles, RoQ2

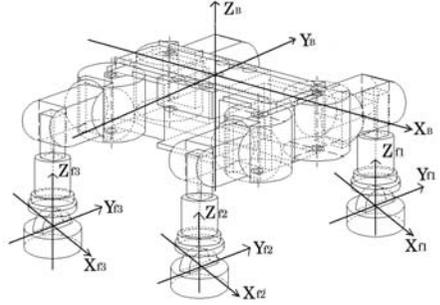


Fig. 2. the coordinate system of RoQ2

2.1 Sole Sensors for COF Estimation

A sole sensor composes of four FSRs(Force Sensing Resistance, ©Interlink Electronics) and two acrylic boards with radius 35mm(Fig.4). Four FSRs are arranged to form a square on a acrylic board and covered with another board. FSR is a kind of polymer thick films and a few micro meter thick, but has low precision. Its standard error of measurement is as much as $\pm 5\% \sim \pm 25\%$, force capacity is 10kg and is depending on history a lot. FSR is not suitable for accurate measurement but its logarithm characteristic serves a wide observable range. Our sole sensor is able to estimate COF with maximum error of 3.6mm on a certain environment.

Two acrylic boards are supported by four screws, which go through the holes in boards(Fig.3). This mechanism causes two advantages. One is to protect FSR of the strain or lateral side force, it is physically fragile toward those forces. The other is a reduction of hysteresis characteristic. Fig.5 shows a typical hysteresis loop of FSR, of which hysteresis becomes worse as the sole sensor covers wider range. It is known that the information loss due to its hysteresis can be reduced by adding slight force bias to them. A set of four screws and nuts plays a role of bias loader.

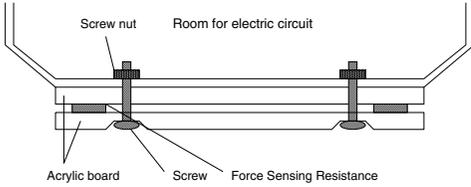


Fig. 3. cross-sectional diagram of sole sensors

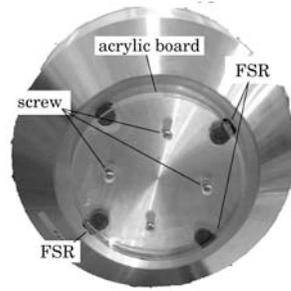


Fig. 4. the bottom view of sole sensory system

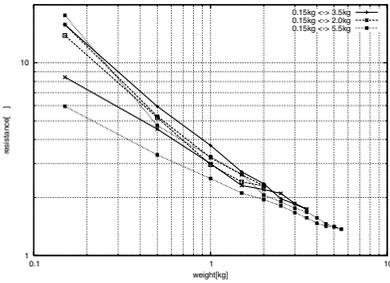


Fig. 5. typical hysteresis loops of FSR:
The bigger the hysteresis curve becomes, the wider range FSR tries to cover.

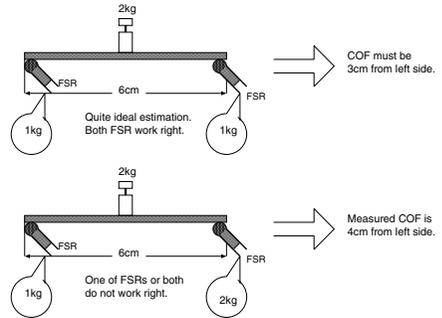


Fig. 6. COF misestimation

2.2 Error Tolerance and Appropriate Bias to Each FSR

A bias should be decided so as to cancel fatal information loss and not to narrow observable area of the force. To compute an appropriate bias, an error tolerance must be given by an operator, the typical hysteresis curve of a FSR should be known beforehand. Suppose the error tolerance of a RoQ2’s sole plane is $\pm 1cm$, the error of COF estimation as much as $\pm 1cm$ does not actually matter. The estimation error of $1cm$ can occur when a real COF locates in the center of a $6cm$ long stick with two FSRs on the sides, and one FSR observes two times value of the other does. (Fig.6) In Fig.6 the COF error of $1cm$ causes when a FSR misestimates two times greater or less of the opposite one, which is named the error tolerance of 2 times. Fig.7 shows linear approximations of hysteresis loop. Both a rapider and shallower curves of hysteresis loop are approximately described as linear functions. In Fig.7 critical error is defined as C_e . The error tolerance of 2 times stands for $log_{10}2 = 0.301$ on a log-log plane. W_b subject to $C_e = 0.301$ is necessary bias that is needed to be added to each FSR. The force bias of $10^{W_b} [kg]$ confines the misestimation of COF to error tolerance at worst. The force bias with critical error C_e is given as (1),

where r_2, r_1, c_2, c_1 decide two linear approximations of hysteresys loop and C_e defines as $C_e = \log_{10} E_t$. Above case supposes E_t of a sole plane is 2 times.

$$\begin{cases} r_2 \log_{10} w_2 + c_2 = r_1 \log_{10} W_b + c_1 \\ \log_{10} w_2 = \log_{10} w_b + C_e \end{cases}$$

$$\log_{10} W_b = \frac{c_1 - c_2 - r_2 C_e}{r_2 - r_1} \quad (1)$$

When typical hysteresys loop has $c_2 = 0.57, c_1 = 0.42, r_2 = -0.67, r_1 = -0.40$ and C_e is set to $\log_{10} 2 = 0.301$, W_b becomes $0.64[kg]$. The force bias of $0.64[kg]$ makes it possible to avoid maximum COF estimation error of $1cm$. Light gray area in Fig.7 indicates the effective area of each FSR when bias is enforced. Unfortunately there is no way to certify that screws and nuts of a sole plane loads stated bias force for misestimation reduction.

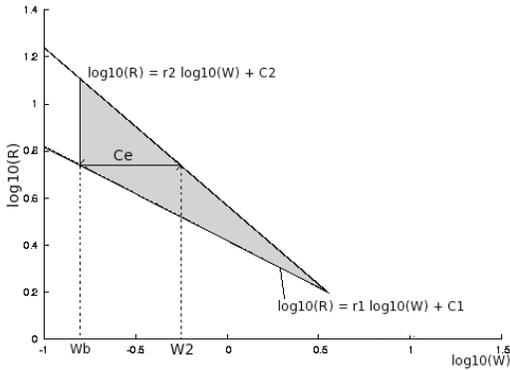


Fig. 7. The critical error and necessary bias of FSR. *Critical error C_e is the allowable maximum error caused by hysteresys loop. W_b is necessary bias added to each FSR.*

2.3 Mechanical Structure of Active Ankles

An active ankle we developed has two active joints and one ball bearing attached to the bottom of the sole(Fig.8). An active joint moves around X_f -axis and the other moves around Z_f -axis(Fig.9). A ball bearing freely rotates around Z_f -axis and enables RoQ2 to change its posture in a supporting phase. We emphasized on high motor torque rather than immediate servo control and employed planetary gear mechanism for the joint around Z_f -axis, its working range is restricted to $0^\circ \sim \pm 50^\circ$ due to electric cables, which are designed to go through a foot and a leg. The joint around X_f -axis is driven by a ball screw and its working range is $0^\circ \sim \pm 40^\circ$. The angles of joints are measured by potentiometers.

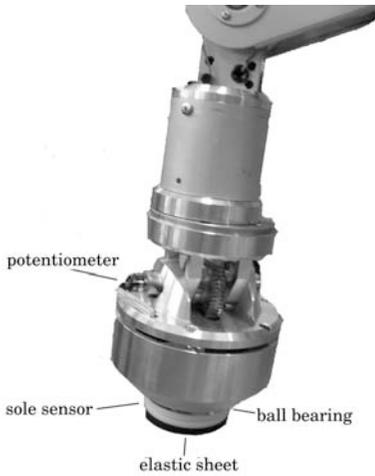


Fig. 8. a picture of an active ankle

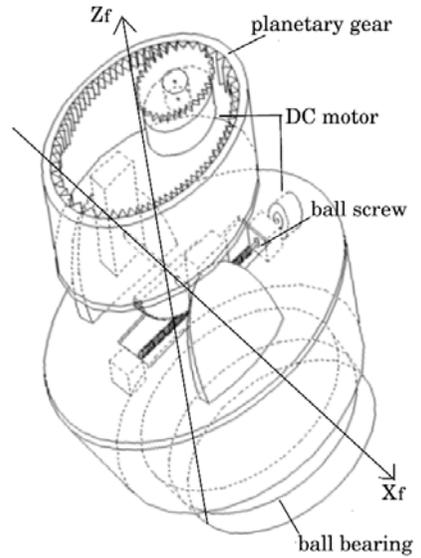


Fig. 9. two active joints and a ball bearing of active ankle

2.4 Installation of Active Ankles on RoQ2

Four active ankles should be installed on each foot so that a quadruped robot obtains an efficient gait and high locomotion on rough terrain. This topic should be discussed enough because the working ranges of active ankles are restricted to quite small. So far we put four ankles symmetrically and let all toes be almost parallel to legs (Fig.2). The idea is that the ball bearing of the each foot enables the torso to move smoothly along X_B -axis by freely rotating in a 4-leg or 3-leg supporting phase, while it is less helpful for the movement of the torso along Y_B -axis.

3 Active Adaptation to Rough Terrain Using COF Estimation

3.1 Adaptation Algorithm

Active ankles need some ground information like the gradient of slope to determine those sole planes. Our proposition is to use COF as a sole plane determiner. When COF is estimated to locate around the center of a sole, four FSRs equally share the pressure and the ankle is considered to adapt to terrain. Thus basic strategy of active adaptation is to move two active joints of an ankle so that COF closes to the center.

When COF is estimated to be in the place shown in Fig.10(a), a joint around Z_f -axis works and COF is expected to get close to Y_f -axis drawing the arc until COF is in the light gray area. When COF is in (b), a joint around X_f -axis works and COF gets close to the center of the sole. COF is being computed all the time by Java thread technology and adaptation algorithm is applied whenever COF is out of the dark gray circle. Actually these movements of the two joints do not happen separately and occur simultaneously. There is a theoretical limit to adaptation of active ankles. When an ankle can not adapt to the terrain, RoQ2 must find another stable foothold among other feasible area.

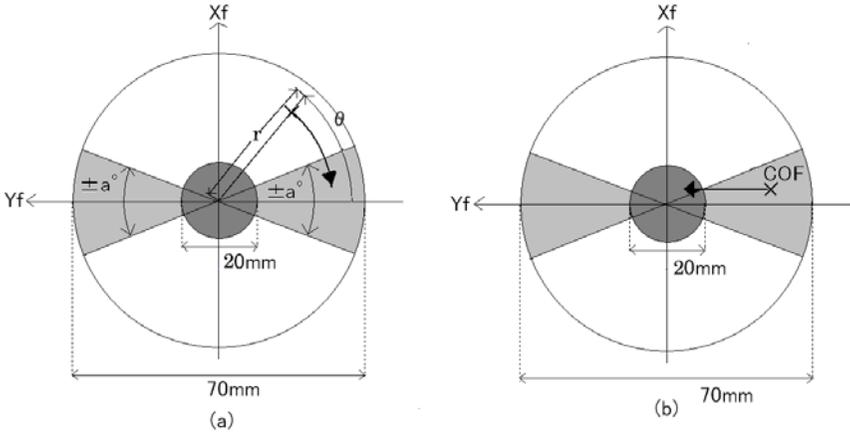


Fig. 10. the basic strategy of active adaptation algorithm using COF estimation

Fig.11 shows the planar model of a leg where the torso or other legs are omitted due to clarity. An enlargement of an active ankle is also shown. When RoQ adapts to the terrain by moving active ankles, not only joints of the ankle but also the upper leg's joints must work. The coordinate of $P(X, Y, Z)$ is computed by (2).

$$\begin{aligned}
 X &= \cos \theta_1 (L_0 + L_1 \cos \theta_2 + L_2 \cos \theta_3) + \cos(\theta_1 + \theta_4 - 15^\circ) (r \cos \theta_5 + L_3 \sin \theta_5) \\
 Y &= \sin \theta_1 (L_0 + L_1 \cos \theta_2 + L_2 \cos \theta_3) - \cos(\theta_1 + \theta_4 + 15^\circ) (r \cos \theta_5 + L_3 \sin \theta_5) \\
 Z &= L_1 \sin \theta_2 + L_2 \sin \theta_3 - r \sin \theta + L_3 \cos \theta
 \end{aligned} \quad (2)$$

The angles of ankle joints, θ_4 and θ_5 are determined by gradient of the footholds. The modification of θ_4 and θ_5 causes upper leg's joint, θ_1, θ_2 and θ_3 to change according to inverse kinematics (3).

$$\theta_1 = \arctan \frac{Y}{X} \tag{3}$$

$$\theta_2 = \pm \arccos \frac{(\sqrt{X^2 + Y^2} - L_0)^2 + L_1^2 - L_2^2 + Z^2}{2L_1\sqrt{Z^2 + (\sqrt{X^2 + Y^2} - L_0)^2}} + \arctan \frac{Z}{\sqrt{(X^2 + Y^2) - L_0}}$$

$$\theta_3 = \pm \arccos \frac{(\sqrt{X^2 + Y^2} - L_0)^2 - L_1^2 + L_2^2 + Z^2}{2L_2\sqrt{Z^2 + (\sqrt{X^2 + Y^2} - L_0)^2}} + \arctan \frac{Z}{\sqrt{(X^2 + Y^2) - L_0}}$$

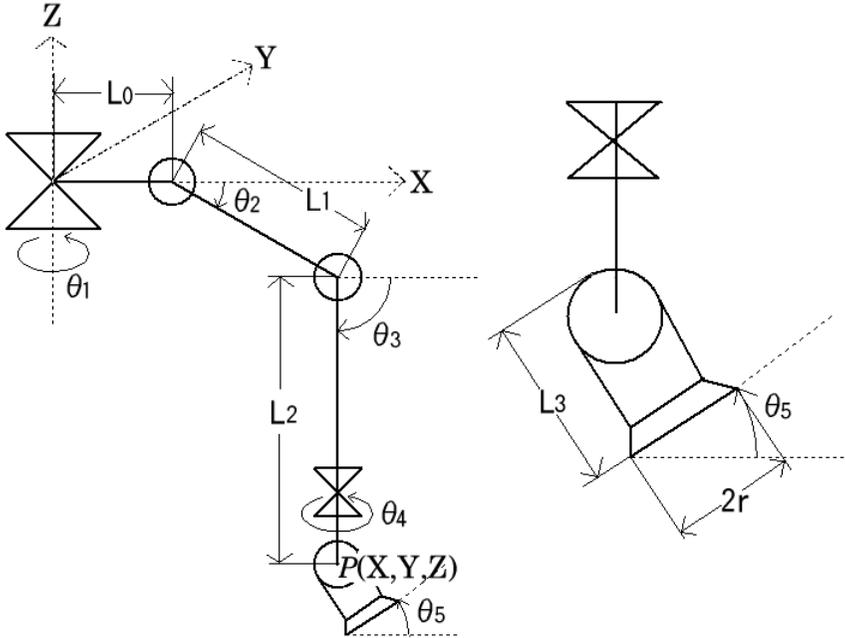


Fig. 11. the planar model of a RoQ2's leg

3.2 Experiment, Adapt a Foot to a Slight Slope

Using the algorithm described in the preceding subsection, the experiment has been done on RoQ2. The experimental condition is that RoQ2 adapts a swinging leg to the 17° inclined slope in 3-leg supporting phase. Sequential photograph of experiment is shown in Fig.12.

In Fig.12, note that not only the joint around X_f -axis but also the joint around Z-axis moves the way the potentiometers moves horizontally in the picture. Adaptation algorithm performs based on a polar coordinate of COF, time

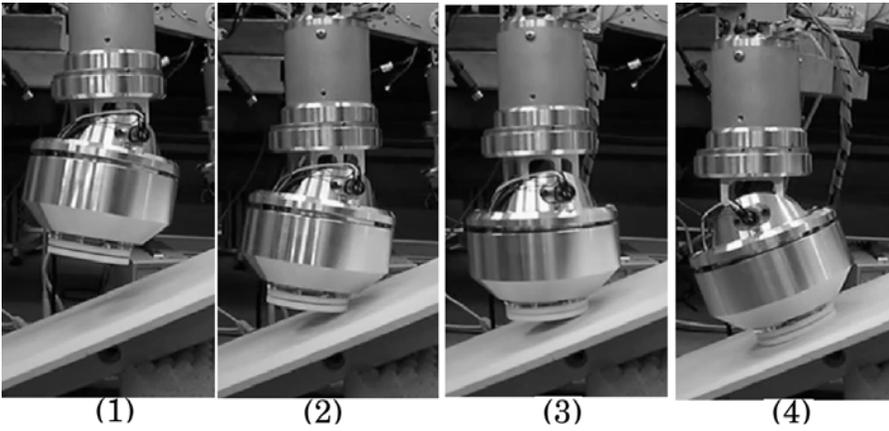


Fig. 12. the sequential photograph of adaptation to a slight slope

series graph of r (distance between COF and the center of the sole) and θ (degree of the angle) helps to understand the algorithm and foot's reactions to estimated COF. An experimental example is shown in Fig.13.

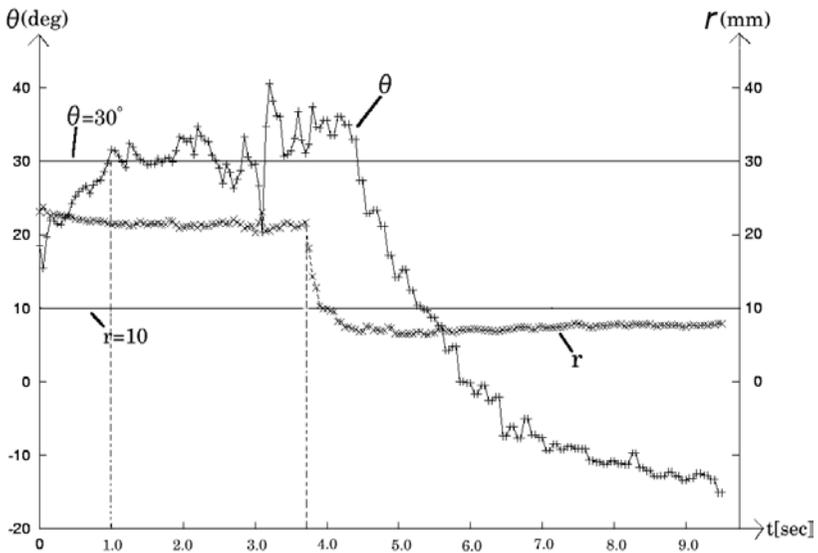


Fig. 13. time series graph of r, θ , the experimental result of active adaptation to a slope

In Fig.13, at the stroke of ground contact($t=0$), only the joint around X_f -axis moves for $\theta_{t=0}$ is less than 30° . But while ankle moves around X_f -axis θ exceeds 30° and the joint around Z_f -axis also starts moving($t=1.0$). Until $t=3.75$, the ankle tries to move around both X_f and Z_f -axes. r suddenly becomes less than 10° with $t=3.75$, then adaptation has been done. The reason why r decreased drastically at $t=3.75$ was that toe was stuck on the board and held on. Getting stuck of toe prevents smooth adaptation of the ankle but sooner of later the ankle is adapted to the terrain. Slowness of its adjustment comes from its high reduction ratios of gears and this should be improved by parallel links or other mechanical structures.

The idea of adaptation algorithm is classic, but it could prove our sensory system operates as a sole plane determiner.

4 Information Acquisition by Foot Groping

4.1 Detectable Footholds with Passive Ankles and Sole Sensors

We have already verified the efficiency of our sole sensors for stability measurement of three kinds of footholds through the prior study on RoQ1 of which ankles are passive. Unstable ground which may be rotatable abruptly, a brittle foothold which may be a thin, hard board and liable to break by huge pressure and cushion-like ground which has soft surface were studied(Fig.14). These three footholds are likely to be disturbance against the stable walking.



Fig. 14. three detectable footholds with sole sensors

4.2 Detect an Edge of a Beam or a Step

To detect an edge of a beam or a step on the ground using passive ankles, a robot must push its sole to the top of the edge. Next a sole must decline to either side by strongly being pushed to the edge, then decline to another side(Fig.15(a)). A passive ankle has some hardness on its joint not to freely swing or not to sprain. To detect an edge of a beam or a step on the ground

with passive ankles, a passive ankle has to be thrust to the edge. But an active ankle can detect them through softly groping. The advantage is to avoid the risk of sliding down on a rapid slope or tripping over, which might happen to a robot with passive ankles.

Fig.16 shows the running motions of the ankle softly groping to detect an edge. After this motion, the robot could know whether there is an edge of a beam or a step. When the ankle can sway to only one side the foot is on an edge of a step(Fig.15(b)). When the ankle can sway to both sides the foot is on an edge of a beam(Fig.15(a)).

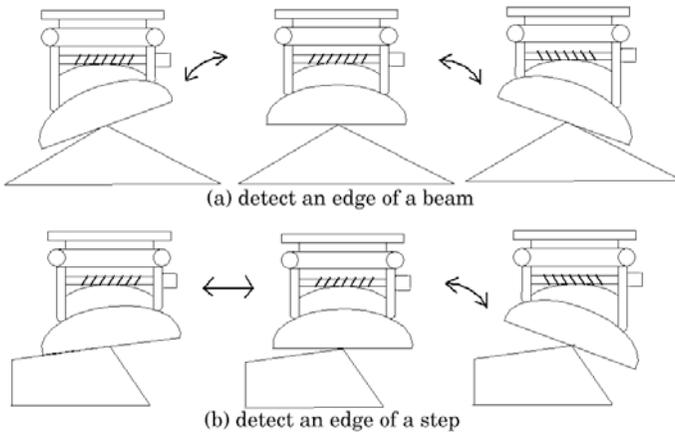


Fig. 15. detect an edge of a beam, how to tell a step from a beam

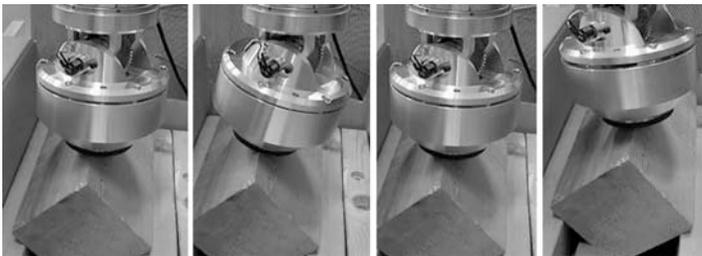


Fig. 16. the sequential photograph of swaying ankle to both sides to detect an edge of a beam or a step

During foot groping to detect an edge, the measured pressure of a sole was up to around 9.0kg several times. Even with 9.0kg pressure, the foot did not

slip or fall during experiment. If it slips, when a quadruped walking vehicle, its weight is 40kg, obtains a big safety margin, a slip of the foot with 9.0kg supported does not matter.

5 Discussion and Conclusion

We developed and installed active ankles with two D.O.F. and sole sensor composed by FSRs on RoQ2. It was verified that the active ankle can adapt to the terrain using COF and sole sensor works as a sole planes determiner in spite of its not good sensing. Active ankle also can detect an edge of a beam or a step on the ground without thrusting the ankle to the edge. Sole sensors for COF estimation are effective not only to detect dangerous footholds for RoQ2 but also to determine the sole planes.

Sole sensors composed by FSRs are less reliable as pressure sensors. Its standard error of measurement might be up to $\pm 25\%$ even with the careful tuning and calibration. More accurate torque sensor must be considered to be added to RoQ2. But our experiments showed that even poor COF estimation with its error tolerance of 1cm can be a sole plane determiner. Next necessary conditions or necessary precision of sole sensor for adjustment to rough terrain should be studied. Automation of foot groping motion and walking experiments are also future works.

References

1. Robine R. Murphy, Rescue Robots at WTC, Trans. Shinobu Makita, Journal of the Japan Society of Mechanical Engineers, Oct, 2003, pp.794-802
2. Toshi Takamori et al., Development of UMRS(Utility Mobile Robot for Search) and Serching System for Sufferers with Cellphone, proc. of SSRR, pp.47-52, 2003
3. Koichi Osuka, Search in Narrow Space by Snake-Like Robots, Journal of the Robotic Society of Japan, Vol.22, No.5, pp.554-557
4. Uluc Saranli et al., RHex: A Simple and Highly Mobile Hexapod Robot, IJRR, Vol. 20, No.7, 2001, pp.616-631
5. Jorge G. Cham et al., Fast and Robust: Hexapedal Robots via Shape Deposition Manufacturing, IJRR, Vol.21, No.10-11, 2002, pp.869-882
6. Kazuhito Yokoi et al., Experimental Study of Humanoid Robot HRP-1S, IJRR, Vol.23, No.4-5, 2004, pp.351-362
7. Tetsuo Tawara et al., Morph: A Desktop-Class Humanoid Capable of Acrobatic Behavior, IJFF, Vol.23, No.10-11, 2004, pp.1097-1103
8. Masaru Ogata and Shigeo Hirose, Study on Ankle Mechanism for Walking Robots, Proc. ROBOMECH2004, p.33, June, 2004
9. Tatsuyoshi Kano and Masahiko Yamamoto et al., Development of a Quadruped Walking Robot with Parallel Link, Proc. ROBOMECH2004, p.71, June, 2004
10. Kenichi Tokuda and Takafumi Toda et al., Estimation of fragile ground by foot pressure sensor of legged robot, Proc. AIM2003, July 2003, pp.447-453

Multi-solution Problem for Track-Terrain Interaction Dynamics and Lumped Soil Parameter Identification

S. Hutangkabodee, Y.H. Zweiri, L.D. Seneviratne, and K. Althoefer

King's College London, Strand, London WC2R 2LS, UK
{suksun.hutangkabodee, yahya.zweiri, lakmal.seneviratne,
k.althoefer}@kcl.ac.uk

Summary. A technique for identifying lumped soil parameters on-line while traversing with a tracked unmanned ground vehicle (UGV) on an unknown terrain is presented. This paper shows the multi-solution problem when identification of soil parameters – cohesion (c), shear deformation modulus (ϕ), and shear deformation modulus (K) are to be attempted using the track-terrain interaction dynamics model. The initiation of the idea of lumping the cohesion and internal friction angle terms and treating them as a single parameter to solve this problem is presented. The technique used for lumped soil parameter identification is based on the Newton Raphson method. This method is proved to be very effective in terms of prediction accuracy, computational speed, and robustness to initial conditions and noise. These identified lumped soil parameters can be used to increase the autonomy of a tracked UGV. The technique presented in this paper is general and can be applied to any tracked UGV.

Keywords: Multi-solution, Lumped, Soil parameter, Identification, Tracked UGVs, Interaction dynamics, Newton Raphson method

1 Introduction

Unmanned Ground Vehicles (UGVs) have many potential applications, including space exploration, defense, agriculture, mining and construction. Most unmanned ground vehicles are currently controlled by tele-operation. Tele-operation requires continuous and repetitive human intervention, which hampers the speed of the vehicle and the range of potential applications [1]. Further they have problems due to bandwidth limitations and communication time delays of the transmission link. Increased autonomy of ground vehicles will not only improve the safety of the operators, but also increase the range of potential applications.

The researches on off-road vehicle and its characteristics have been carried out since Bekker first pioneered in this field [2, 3]. The empirical relationships of vehicle terrain interactions for both tracked and wheeled vehicles have been modelled and redeveloped based on Mohr-Coulomb theory [2, 3, 4, 5]. In [6], the slip-based traction model is developed theoretically and is used to establish an effective control law for the rover travelling on rough terrain. The rover's traversability can be improved by controlling the wheel slip. The slip estimation of a tracked vehicle from trajectory measurement using Extended Kalman Filter is accomplished by [7]. The slip estimate together with the knowledge of the terrain parameters will allow intelligent autonomous control of a tracked vehicle through vehicle trajectory modification in real time.

A potential and articulate way to improve the performance and autonomy of UGVs is to acquire information from the terrain which the UGVs traverse. This is where the soil parameter identification plays an important role. The real-time acquisition of accurate soil parameters based on a physical model and numerical techniques will enable a UGV to autonomously achieve an accurate traversability prediction and effective traction control. The key numerical technique employed and validated in this work is the Newton Raphson method.

In recent years, there has been increasing interest in parameter identification in various engineering applications. An on-line identification of link parameters (mass, inertia and length) and friction coefficients of a full scale excavator arm is presented by [8]. An on-line soil parameter identification based on the Newton Raphson method is applied to autonomous excavation by [9]. The aim was to increase the excavation autonomy based on knowledge of soil parameters. The Linear Least Square estimator is employed as an on-line identification technique by [10] to identify two key soil parameters using on-board rover sensors. This estimator is applied to a simplified linearized model of the rover's wheel-terrain interaction. In [11], Iagnemma applies the soil parameter identification scheme to high-speed vehicle traversing rough terrain. The Newton Raphson method are employed by [12] and [13] for soil parameter identification for a tracked vehicle traversing an unknown terrain. The author attempts to identify all three soil parameters which are cohesion, c , internal friction angle, ϕ , and shear deformation modulus, K [13]. However, only ϕ and K can successfully be identified and the weight of the tracked vehicle which is classified as extremely heavy is proved to be the reason for c not converging to its actual value.

In this paper, further investigation on the track-terrain interaction dynamics model used in [13] is carried out. It is found that the multi-solution manner of the track-terrain interaction dynamics model is the true cause for c not converging to its actual value. The multi-solution problem can be solved by treating a problem-originating term ($Ac + W\tan\phi$) as a single parameter called lumped soil parameter. This term takes into account both c and ϕ . The identification of this lumped term and shear deformation modulus, K is presented in this paper.

2 Analytical Model

To identify soil parameters, track-terrain interaction dynamics model in (1) is employed [4]. This model characterizes the interaction between the vehicle's tracks and particular types of terrains such as loose sand, saturated clay, dry fresh snow, and most of the dispersed soils. The soil parameter identification scheme which is a novel extension from Wong work [4] is based on this model. The track-terrain interaction dynamics relationships for other types of terrain are depicted in appendix A.

$$F = (Ac + W \tan \phi) \left[1 - \frac{K}{il} \left(1 - e^{-il/K} \right) \right], \quad (1)$$

where $A = 2bl$ is the contact area of the tracks, $W = pA$ is the normal load due to the weight of the tracked vehicle, p is the vertical pressure over the terrain, and i is the track slip.

The detailed derivation of this force-based track-terrain interaction dynamics is illustrated as follows.

Figure 1 shows the measured and approximated shear stress vs. shear displacement behavior for loose sand, saturated clay, dry fresh snow, and most of the dispersed soils. The dashed line in Fig. 1 is drawn to fit the measured data for the derivation of the more accurate shear stress model (compared to Mohr-Coulomb model). As a result, the shear stress dynamics is derived in an exponential form as follows,

$$\tau = \tau_{max} \left(1 - e^{-il/K} \right), \quad (2)$$

where $\tau_{max} = c + \sigma \tan \phi$ is the maximum shear stress, c is the soil cohesion, ϕ is the angle of the internal shearing resistance of the soil, j is the shear displacement, and K is the shear deformation modulus.

This equation (τ_{max}) is based on Mohr-Coulomb theory as seen from the term " $c + \sigma \tan \phi$ ". This term is purely an empirical relationship and is based on Amonton's law of friction for sliding between two plane surfaces. It postulates that the soil at a point will fail if the shear stress at that point in the medium satisfies the condition " $c + \sigma \tan \phi$ ". Cohesion of the material is the bond that cements particles together irrespective of the normal pressure exerted by one particle upon the other. On the other hand, particles of frictional masses can be held together only when a normal pressure exists between them. This leads to the existence of the normal pressure term, σ with the internal friction angle term, ϕ in the term " $\sigma \tan \phi$ ".

Replacing " σ " (Normal pressure) with $p(x, y)$ (pressure distribution varying along x and y) yields

$$\tau = (c + p(x, y) \tan \phi) \left(1 - e^{-il/K} \right). \quad (3)$$

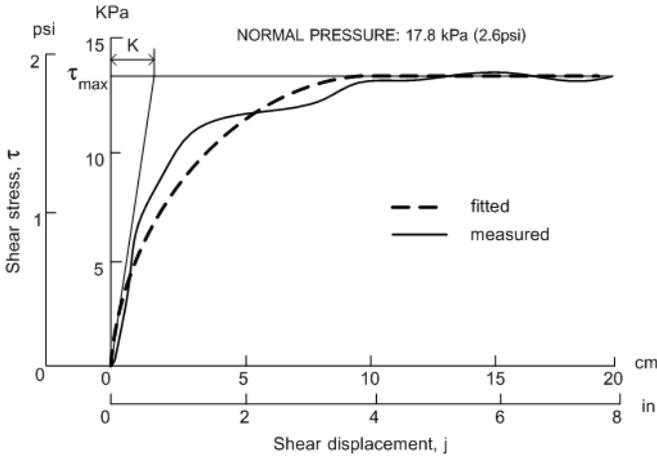


Fig. 1. Plot of shear stress against shear displacement for a tracked UGV traversing loose sand, saturated clay, dry fresh snow, and most of the dispersed soils [4]

The tractive force developed by a track is obtained by integrating (3) over the track-terrain contact area as below,

$$F = \int_0^l \int_{-b/2}^{b/2} (c + p(x, y) \tan\phi) \left(1 - e^{-il/K}\right) dy dx, \quad (4)$$

where b is the track width, and l is the track length.

Assumptions

1. Uniform normal pressure distribution under the tracks.
2. Uniform shear deformation along y direction (along the width of the tracks).
3. Steady slip “ i ”.

The term “ j ” can be replaced by “ ix ” since the system model is for straight line motion in x direction which makes “ j ” constant in y direction. By replacing “ j ” with “ ix ” and using assumption 1 and 2, (4) is reduced to

$$F = b \int_0^l (c + p \tan\phi) \left(1 - e^{-il/K}\right) dx. \quad (5)$$

With assumption 3, (5) becomes

$$F = (Ac + W \tan\phi) \left[1 - \frac{K}{il} \left(1 - e^{-il/K}\right)\right]. \quad (6)$$

3 Multi-solution Manner of Track-Terrain Interaction Model

To investigate the characteristics of the track-terrain interaction dynamics model, (1), the random test to find the possible solutions for this model is carried out. This is done by substituting each soil parameter (cohesion, c , internal friction angle, ϕ , and shear deformation modulus, K) within their real practical range into (1). The range of each parameter as well as their increased interval is shown below.

$$\begin{aligned} c &= [0 - 10] \text{ kPa} && (0.01 \text{ kPa increased interval}) \\ \phi &= [5 - 50] \text{ degree} && (0.01 \text{ degree increased interval}) \\ K &= [0.001 - 0.04] \text{ m} && (0.01 \text{ m increased interval}) \end{aligned}$$

According to three sets of measured data (tractive force, F and slip, i) from [4], the ranges of each measured F with respect to each i are chosen to allow the margin of error for resultant F values from the exact measured F . The ranges of tractive force with respect to each slip, i are as follows.

$$\begin{aligned} F_1 &= [209 - 214] \text{ kN} && (\text{for } i_1 = 0.0248) \\ F_2 &= [231 - 235] \text{ kN} && (\text{for } i_2 = 0.0344) \\ F_3 &= [252 - 256] \text{ kN} && (\text{for } i_3 = 0.07) \end{aligned}$$

Note : The real measured data sets from [4] are

$$\begin{aligned} F_1 &= 211.46 \text{ kN} && ,i_1 = 0.0248, \\ F_2 &= 233.10 \text{ kN} && ,i_2 = 0.0344, \\ F_3 &= 254.18 \text{ kN} && ,i_3 = 0.07. \end{aligned}$$

The random test is set in such a way to extract the soil parameters that make the tractive forces of i_1 , i_2 , and i_3 fall into ranges F_1 , F_2 , and F_3 , respectively. These soil parameters are the possible solutions of the test. After, completing the test, 54,990 sets of soil parameters (c , ϕ , and K) appear to be the possible solutions of our test. This clearly shows that the track-terrain interaction dynamics have multi-solution problem. The solutions are tabulated in Table 1 below.

The manner of the multi-solution problem for the track-terrain interaction dynamics model is described here. The soil parameters of concern are c , ϕ , and K . After trying thousands of different soil parameters, K that gives F close to the measured data for the corresponding i falls into the range of its actual value (0.0175 – 0.0185 m) whereas c and ϕ fall into the broad range.

Table 1. Solutions of random test

| Set of Solution | Solution for c (kPa) | Solution for ϕ (degree) | Solution for K (m) |
|-----------------|---------------------------|---------------------------------|-------------------------|
| 0–10000 | 0–1.89 | 39.49–40.42 | 0.0175–0.0185 |
| 10001–20000 | 1.89–3.75 | 38.88–39.82 | 0.0175–0.0185 |
| 20001–30000 | 3.75–5.57 | 38.28–39.22 | 0.0175–0.0185 |
| 30001–40000 | 5.57–7.36 | 37.68–38.62 | 0.0175–0.0185 |
| 40001–50000 | 7.36–9.13 | 37.07–38.02 | 0.0175–0.0185 |
| 50001–54990 | 9.13–10 | 36.77–37.43 | 0.0175–0.0185 |

Note : the actual values of soil parameters are

$$c = 0.55 \text{ kPa}$$

$$\phi = 40.1 \text{ degree}$$

$$K = 0.018 \text{ m}$$

From the interaction model, (1), there are two separate terms on the right hand side of the equation – the term $(Ac + Wtan\phi)$ and K -associated term. The analysis of this model shows that the term $(Ac + Wtan\phi)$ is the cause of multi-solution problem since for any arbitrary c , there is always ϕ that makes the overall value of $(Ac + Wtan\phi)$ the same. To prove this, the samples of c and ϕ values from each solution set range (the mid value of each solution set range is chosen) from Table 1 is substituted into the term $(Ac + Wtan\phi)$ and the results are drawn in Table 2.

Table 2. Investigation of $(Ac + Wtan\phi)$ value derived from c and ϕ sampled from different solution set range

| Set of Solution | c Sample (kPa) | ϕ Sample (degree) | Resultant Value of $Ac + Wtan\phi$ term (kN) |
|-----------------|---------------------|---------------------------|---|
| 5000 | 0.95 | 39.85 | 277.55 |
| 15000 | 2.82 | 39.46 | 279.59 |
| 25000 | 4.67 | 38.70 | 278.10 |
| 35000 | 7.36 | 38.08 | 280.67 |
| 45000 | 8.25 | 37.62 | 279.20 |
| 52500 | 9.57 | 37.14 | 278.93 |

From Table 2, it can be seen that resultant values of $(Ac + Wtan\phi)$ term derived from c , ϕ samples from different solution set range are very close with only 1.11% gap between the maximum value (280.67 kN) and the minimum value (277.55 kN). Hence, it can be concluded that $(Ac + Wtan\phi)$ is always constant for a certain type of terrain on which a particular tracked UGV traverses. The slight differences of this term among different solution sets come from the initial set-up of the measured tractive force range, i.e. $F_1 = [209–214]$ kN for $i_1 = 0.0248$, so on for F_2 and F_3 . However, with different

tracked UGV specification (different vehicle), the value of this term changes despite the vehicle traversing the same terrain.

To attack this multi-solution problem, the term $(Ac + Wtan\phi)$ is lumped and treated as a single parameter. By doing that, the multi-solution effect vanishes and there will be only two parameters to be identified – K and the lumped term. The track-terrain interaction dynamics model, (1), can then be modified to

$$F = Lump \left[1 - \frac{K}{il} \left(1 - e^{-il/K} \right) \right], \tag{7}$$

This model is the central focus for lumped soil parameters identification.

4 Implementation of Identification Technique

The detailed description of the Newton Raphson method is presented in [13].

The Newton Raphson method is implemented to identify soil parameters for UGV track-terrain interaction dynamics. The Newton Raphson method implementation is shown in Fig. 2. Vector \mathbf{p} has two soil parameters which are lumped parameter ($Lump$) and shear deformation modulus (K). Measurement vector \mathbf{x} contains two sets of measured data (tractive force, F and slip, i) from [4]. The track-terrain interaction model described by (7) in Sect. 3 and measurement vector \mathbf{x} are used to identify unknown soil parameters. Applying the Newton Raphson method, (7) can be expressed as

$$\begin{bmatrix} f_1(Lump, K, F_1, i_1) \\ f_2(Lump, K, F_2, i_2) \end{bmatrix} = 0, \tag{8}$$

where $\mathbf{p} = [Lump, K]^T$, and $\mathbf{x} = [F, i]^T$.

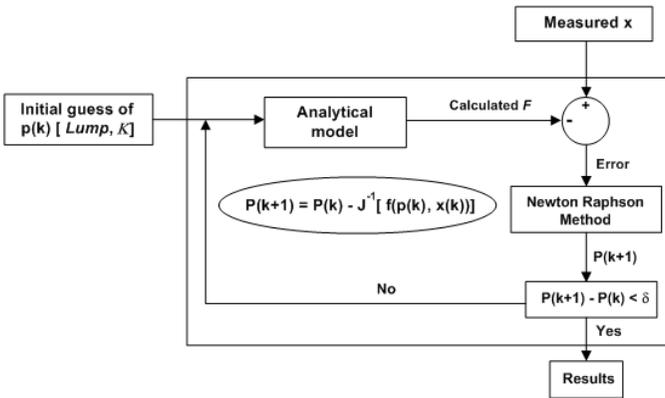


Fig. 2. Diagram showing implementation of the Newton Raphson method for lumped soil parameter identification

Taylor Series expansion is used to approximate the non-linear equation of the functions and the expansion for the first function is as

$$\begin{aligned}
 [f_1(Lump, K)]_{(k+1)} &= [f_1(Lump, K)]_k + \\
 &\left[\frac{\partial f_1}{\partial Lump} \Big|_{Lump, K} \Delta Lump + \frac{\partial f_1}{\partial K} \Big|_{Lump, K} \Delta K \right] + \\
 &[\text{Higher order terms}]. \tag{9}
 \end{aligned}$$

For function f_2 , similar expansions can be derived. Higher order terms of the series are neglected because the series will be calculated in an iterative manner to approximate the function. Note that the partial derivatives are evaluated at the estimated values of the parameters and therefore computable to a simple numerical value.

Let $\mathbf{p}_0 = [Lump, K]_0^T$ be an initial guess. Applying the Newton Raphson method to (8), the matrix representation for the Newton Raphson method for our case is presented as

$$\begin{bmatrix} Lump \\ K \end{bmatrix} = \begin{bmatrix} Lump \\ K \end{bmatrix}_0 - J^{-1} \begin{bmatrix} f_1(Lump, K, F_1, i_1) \\ f_2(Lump, K, F_2, i_2) \end{bmatrix}, \tag{10}$$

where J (Jacobian Matrix) =
$$\begin{bmatrix} \frac{\partial f_1}{\partial Lump} & \frac{\partial f_1}{\partial K} \\ \frac{\partial f_2}{\partial Lump} & \frac{\partial f_2}{\partial K} \end{bmatrix}_0.$$

5 Identification Results and Discussion

The lumped term, $(Ac + Wtan\phi)$ and shear deformation modulus, K are two unknown parameters to be identified. Two sets of measured data from [4] pp. 176 are used in the identification scheme ($F_1 = 211.46$ kN, $i_1 = 0.0248$ and $F_2 = 254.18$ kN, $i_2 = 0.070$). The identification results are depicted in Table 3. In this table, the identification errors from the actual soil parameter values and the elapsed time are also presented.

Table 3. Identification results of soil parameters

| Soil Parameters | Actual Values | Identified Values | Error (%) |
|------------------|---------------|-------------------|-----------|
| Lumped term (kN) | 278.75 | 278.24 | 0.183 |
| K (m) | 0.0180 | 0.0185 | 2.78 |
| Elapsed time (s) | | 0.03 | |

Note : The actual value of the lumped term $(Ac + Wtan\phi)$ is calculated from the actual c and actual ϕ with the tracked UGV’s specification A and W .

It can be seen from Table 3 that the accuracy of the identification for the Newton Raphson method is very good observing from just 0.185% error for Lumped parameter and 2.78% error for K . The speed of convergence of the Newton Raphson method is very fast with 0.03second elapsed time. Therefore, the Newton Raphson method has promising potential for on-line soil parameter identification.

A robustness test is carried out in order to examine whether the Newton Raphson method will converge to the correct solution when different initial guesses (\mathbf{p}_0) are used. It is found that the Newton Raphson method is very robust as all the initial conditions within parameter ranges shown in Table 4 give true converged soil parameter values.

Table 4. The range of initial conditions of soil parameters that produces the converged solution

| Lower bound | Soil paramters | Upper bound |
|-------------|-----------------------|-------------|
| 50 | Lumped parameter (kN) | 700 |
| 0.001 | K (m) | 0.04 |

To check the sensitivity of the Newton Raphson method to sensor noise, the measured data (both F and i) were superimposed by 3% white noise. The percentage errors of the identified soil parameters with the influence of the white noise superimposed to the original measured data were shown in Table 5 in comparison to those computed directly from the measured data.

Table 5. The influence of white noise applied on the measured data

| | With actual measured data | With measured data and white noise |
|--|---------------------------|------------------------------------|
| Error of identified Lumped parameter (%) | 0.185 | 3.179 |
| Error of identified K (%) | 2.78 | 6.11 |

From Table 5, when the measured data were superimposed with the white noise, the errors of the estimate increase from 0.185% to 3.179% and 2.78% to 6.11% for lumped soil parameter and K , respectively. As expected, increasing the sensor noise levels leads to increased parameter identification error. However, the percentage errors are still in the acceptable levels and the Newton Raphson method is considered to be relatively robust to sensor noise.

The identified lumped soil parameter, $(Ac + Wtan\phi)$ and K using the Newton Raphson method from Table 3 are used to predict back tractive force and compare the results with the measured data [4] for validation purposes. Fig. 3 shows the comparison between measured tractive force and tractive

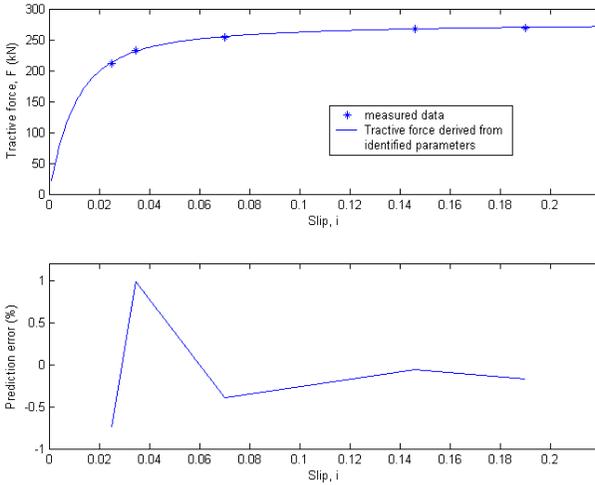


Fig. 3. Comparison between measured and predicted tractive force using the Newton Raphson method and the prediction error

force predicted from the identified lumped soil parameter, $(Ac + Wtan\phi)$ and K . The prediction error ranges from -0.7% to 1%. This reflects a very good prediction accuracy of the tractive force. Thus the identified soil parameters can be used for UGV traversability prediction and trajectory planning in real time based on accurate predicted tractive force. This is beneficial for autonomy purposes of UGVs.

6 Conclusion and Future Work

The multi-solution problem of the track-terrain interaction dynamics model is acknowledged by the random test. The investigation and analysis show that this problem originates from the term $(Ac + Wtan\phi)$ in which c and ϕ compensate each other in a number of ways (multi-solutions) to make the same value of $(Ac + Wtan\phi)$. This occurrence initiates the idea to treat this term as a single soil parameter called “Lumped soil parameter” to solve multi-solution problem.

The Newton Raphson method is applied as soil parameter identification technique for the modified track-terrain interaction dynamics model to identify lumped soil parameter, $(Ac + Wtan\phi)$ and shear deformation modulus, K . The Newton Raphson method is shown to be excellent in all aspects including parameter identification accuracy, robustness to a wide range of initial conditions, robustness to noise, and computational speed.

The future work will focus on the soil parameter identification of a tracked UGV traversing different terrain categories illustrated in appendix A. The hybrid among different track-terrain interaction dynamics models will be carried out to benefit the soil parameter identification in any terrain. Also, research on traversability prediction based on the use of the identified soil parameters will be carried out.

7 Acknowledgement

The authors thank J.Y. Wong for providing useful experimental information. Also, the authors would like to acknowledge EPSRC (GR/S31402/01), Ministry of Defense (MoD), QinetiQ Ltd. and DSTL for funding this project.

References

1. Zweiri YH, Seneviratne LD, Althoefer K (2003) *Journal of Systems and Control Engineering* 217:259–274
2. Bekker G (1956) *Theory of Land Locomotion*. University of Michigan Press
3. Bekker G (1969) *Introduction of Terrain-Vehicle Systems*. University of Michigan Press
4. Wong JY (2001) *Theory of Ground Vehicles (3rd Edition)*. John Wiley & Sons, USA
5. Wong JY (1989) *Terramechanics and Off-Road Vehicles*. Springer, Elsevier Science Publishers B.V. , Netherlands
6. Yoshida K, Hamano H (2002) *IEEE International Conference on Robotics and Automation* 3:3155–3160
7. Le AT, Rye DC, Durrant-Whyte HF (1997) *IEEE International Conference on Robotics and Automation* 2:1388–1393
8. Zweiri YH, Seneviratne LD, Althoefer K (2004) *IEEE Transactions on Robotics* 20:762–767
9. Tan C, Zweiri YH, Seneviratne LD, Althoefer K (2003) *IEEE International Conference on Robotics and Automation* 1:121–126
10. Iagnemma K, Golda D, Spenko M, Dubowsky S (2004) *IEEE Transactions on Robotics* 20:5:921–927
11. Iagnemma K, Dubowsky S (2002) *SPIE Conference on Unmanned Ground Vehicle Technology IV* 4715:256–266
12. Song Z, Hutangkabodee S, Zweiri YH, Seneviratne LD, Althoefer K (2004) *SICE Annual Conference* 2255–2260
13. Hutangkabodee S, Zweiri YH, Seneviratne LD, Althoefer K (2004) *MECHROB Conference* 3:889–895

Appendix A

Shear-based track-terrain interaction dynamics models for different categories of terrains (from the one used in this paper) are described below.

A.1 Organic terrain (muskeg) with a mat of living vegetation on the surface and saturated peat beneath it

The shear stress - shear displacement relationship for this type of terrain exhibits characteristics shown in Fig. 4 (a) and its shearing behavior can be described by

$$\tau = \tau_{max} \left[(j/K_\omega) e^{(1-j/K_\omega)} \right], \tag{11}$$

where K_ω is the shear displacement where τ_{max} occurs.

A.2 Compact sand, silt and loam, and frozen snow

The shear stress - shear displacement relationship for this type of terrain exhibits characteristics shown in Fig. 4 (b) and its shearing behavior can be described by

$$\tau = \tau_{max} K_r \left\{ 1 + [1/(K_r(1 - 1/e)) - 1] e^{(1-j/K_\omega)} \right\} \left[1 - e^{(-j/K_\omega)} \right], \tag{12}$$

where K_r is the ratio of the residual shear stress τ_r to the maximum shear stress τ_{max} , and K is the shear displacement where τ_{max} occurs.

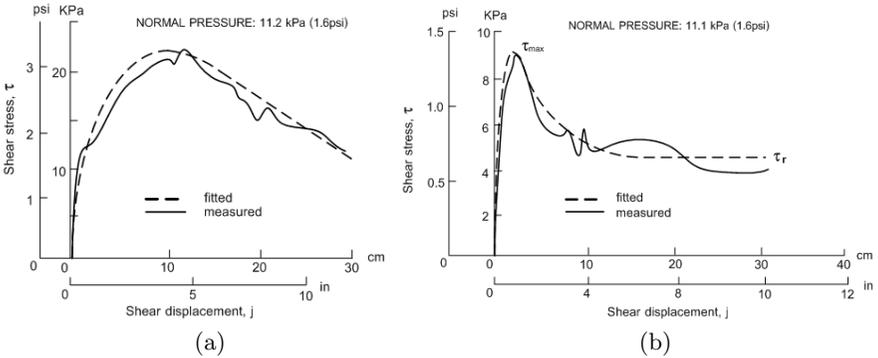


Fig. 4. (a) and (b) show plots of shear stress against shear displacement for a tracked vehicle travelling on organic terrain (muskeg) and on compact sand, silt and loam, and frozen snow, respectively [4]

3D Position Tracking in Challenging Terrain

Pierre Lamon and Roland Siegwart

Ecole Polytechnique Fédérale de Lausanne {firstname.lastname}@epfl.ch

Summary. The intent of this paper is to show how the accuracy of 3D position tracking can be improved by considering rover locomotion in rough terrain as a holistic problem. An appropriate locomotion concept endowed with a controller minimizing slip improves the climbing performance, the accuracy of odometry and the signal/noise ratio of the onboard sensors. Sensor fusion involving an inertial measurement unit, 3D-Odometry, and visual motion estimation is presented. The experimental results show clearly how each sensor contributes to increase the accuracy of the 3D pose estimation in rough terrain.

1 Introduction

In order to acquire knowledge about the environment, a mobile robot uses different types of sensors, which are error prone and whose measurements are uncertain. In office-like environments, the interpretation of this data is facilitated thanks to the numerous assumptions that can be formulated e.g. the soil is flat, the walls are perpendicular to the ground, etc. In natural scenes, the problem is much more tedious because of limited *a priori* knowledge about the environment and the difficulty of perception. In rough terrain, the change in lighting conditions can strongly affect the quality of the acquired images and the vibrations due to uneven soils lead to noisy sensor signals. When the robot is overcoming an obstacle, the field of view can change significantly between two data acquisitions, increasing the difficulty of tracking features in the scene.

To get a robust estimate of the robots position, the measurements acquired by several *complementary* sensors have to be fused accounting for their relative variance. In the literature, the localization task generally involves two types of sensors and is divided into two phases a) the first step consists in the integration of a high frequency dead reckoning sensor to predict vehicle location b) the second phase, which is usually activated at a much slower rate, uses an absolute sensing mechanism for extracting relevant features in the environment and updating the predicted position. In [1], an inertial measurement

unit is used for the prediction and an omnicaam is used as the exteroceptive sensor. The pair of sensors composed of an inertial measurement unit and a GPS is used in [2]. Even if sensor fusion can be applied to combine the measurements acquired by any number of sensors, most of the applications found in the literature generally use only two types of sensors and only the 2D case is considered (even for terrestrial rovers).

In challenging environments, the six degrees of freedom of the rover have to be estimated (3D case) and the selection of sensors must be done carefully because of the aforementioned difficulties of perception in rough terrain. However, the accuracy of the position estimates does not only depend on the quality and quantity of sensors mounted onboard but also on the specific locomotion characteristics of the rover and the way it is driven. Indeed, the sensor signals might not be usable if an unadapted chassis and controller are used in challenging terrain. For example, the ratio signal/noise is poor for an inertial measurement unit mounted on a four-wheel drive rover with stiff suspensions. Furthermore, odometry provides bad estimates if the controller does not include wheel-slip minimization or if the kinematics of the rover is not accounted for.

The intent of this paper is to show how the accuracy of 3D position tracking can be improved by considering rover locomotion in rough terrain as a holistic problem. Section 2 describes the robotic platform developed for conducting this research. In Sect. 3, a method for computing 3D motion increments based on the wheel encoders and state sensors is presented. Because it accounts for the kinematics of the rover, this method provides better results than the standard method. Section 4 proposes a new approach for slip-minimization in rough terrain. Using this controller, both the climbing performance of the rover and the accuracy of the odometry are improved. Section 5 presents the results of the sensor fusion using 3D-Odometry, an Inertial Measurement Unit (IMU) and Visual Motion Estimation based on stereovision (VME). The experiments show clearly how each sensor contributes to increase the accuracy and robustness of the 3D pose estimation. Finally, Sect. 6 concludes this paper.

2 Research Platform

The Autonomous System Lab (at EPFL) developed a six-wheeled off-road rover called Shrimp, which shows excellent climbing capabilities thanks its passive mechanical structure [3]. The most recent prototype, called SOLERO, has been equipped with sensors and more computational power (see Fig. 1). The parallel architecture of the bogies and the spring suspended fork provide a high ground clearance while keeping all six motorized wheels in ground-contact at any time. This ensures excellent climbing capabilities over obstacles up to two times the wheel diameter and an excellent adaptation to all kinds of terrains. The ability to move smoothly across rough terrain has many advantages when dealing with onboard sensors: for example, it allows limited wheel slip

and reduces vibration. The quality of the odometric information and the ratio signal/noise for the inertial sensors are significantly improved in comparison with rigid structures such as four-wheel drive rovers. Thus, both odometry and INS integration techniques can be accounted for position estimation.

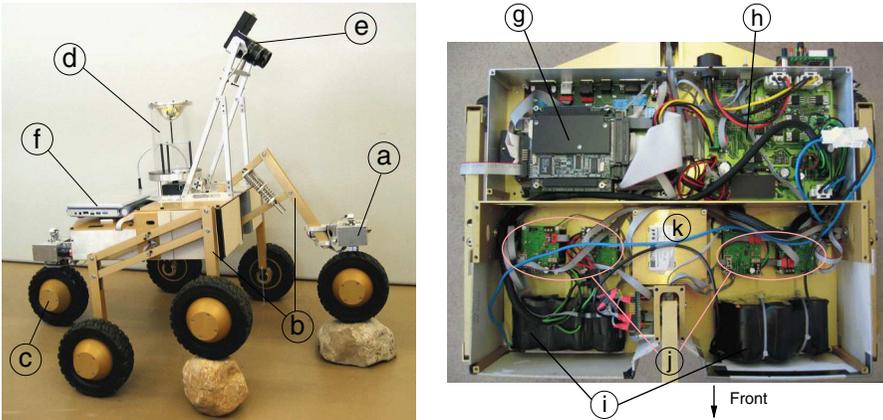


Fig. 1: Sensors, actuators and electronics of SOLERO. a) steering servo mechanism b) passively articulated bogie and spring suspended front fork (equipped with angular sensors) c) 6 motorized wheels (DC motors) d) omnidirectional vision system e) stereo-vision module, orientable around the tilt axis f) laptop (used for image processing) g) low power pc104 (used for sensor fusion) h) energy management board i) batteries (NiMh 7000 mAh) j) I²C slave modules (motor controllers, angular sensor module, servo controllers etc.) k) IMU (provides also roll and pitch)

3 3D-Odometry

Odometry is widely used to track the position and the orientation $([x, y, \psi]^T)$ of a robot in a plane π . This vector is updated by integrating small motion increments between two subsequent robot poses. This 2D odometry method can be extended in order to account for slope changes in the environment and to estimate the 3D position in a global coordinate system i.e. $[x, y, z, \phi, \theta, \psi]^T$. This technique uses typically an inclinometer for estimating the roll (ϕ) and pitch (θ) angles relative to the gravity field [4]. Thus, the orientation of the plane π , on which the robot is currently moving, can be estimated. The, z coordinate is computed by projecting the robot displacements in π into the global coordinate system. This method, which will be referred later as the *standard method*, works well under the assumption that the ground is relatively smooth and does not have too many slope discontinuities. Indeed, the system accumulates errors during transitions because of the planar assumption. In

rough terrain, this assumption is not verified and the transitions problem must be addressed properly. This section briefly describes a new method, called 3D-Odometry, which takes the kinematics of the robot into account and treats the slope discontinuity problem. The main reference frames and some of the variables used for 3D-Odometry are introduced in Fig. 2

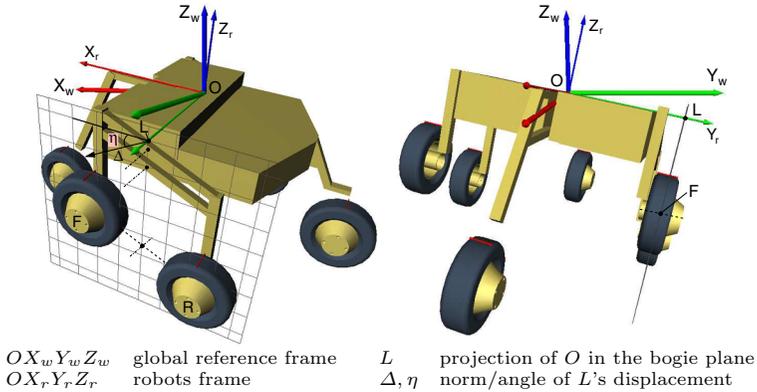


Fig. 2: Reference frames definition

The norm Δ and the direction of motion η of each bogie can be computed by considering the kinematics of the bogie, the incremental displacement of the Rear/Front bogie wheels (wheel encoders) and the angular change of the bogie (angular sensor) between two data acquisition cycles. Then, the displacement of the robot's center O , i.e. $[x, y, z, \psi]^T$, can be computed using Δ and η of the left and the right bogie, whereas the attitude $[\phi, \theta]^T$ is directly given by the inclinometer¹.

Experimental results

The robot has been driven across obstacles of known shape and the trajectory computed online with both 3D-Odometry and the standard method. In all the experiments, the 3D-Odometry produced much better results than the standard method because the approach accounts for the kinematics of the rover. The difference between the two techniques becomes bigger as the difficulty of the obstacles increases (see Fig. 3). In Fig. 4, an experiment testing the full 3D capability of the method is depicted. The position error at the goal is only $\epsilon_x = 1.4\%$, $\epsilon_y = 2\%$, $\epsilon_z = 2.8\%$, $\epsilon_\psi = 4\%$ for a total path length of around 2m.

SOLERO has a non-hyperstatic mechanical structure that yields a smooth trajectory in rough terrain. As a consequence wheel slip is intrinsically minimized. When combined with 3D-Odometry, such a design allows to use odom-

¹ The reader can refer to the original paper [5] for more details about 3D-Odometry. In particular, the method also computes the wheel-ground contact angles.

etry as a mean to track the rover's position in rough terrain. Moreover, the quality of odometry can still be significantly improved using a "smart" controller minimizing wheel slip. Its description is presented in the next section.

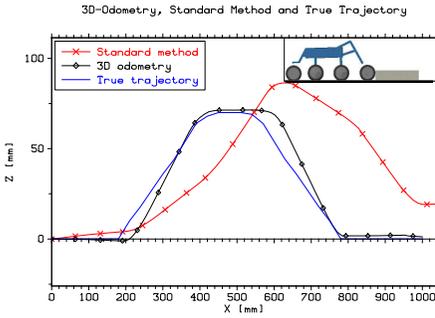
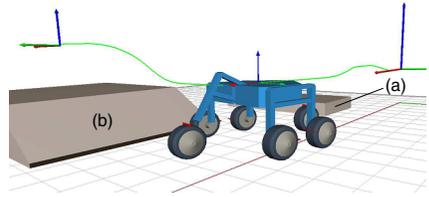


Fig. 3: Sharp edges experiment



Only the right bogie wheels climbed obstacle (a). Then, the rover has been driven over obstacle (b) (with an incident angle of approximately 20°)

Fig. 4: Full 3D experiment

4 Wheel Slip Minimization

For wheeled rovers, the motion optimization is somewhat related to minimizing wheel slip. Minimizing slip not only limits odometric error but also increases the robot's climbing performance and efficiency. In order to fulfill this goal, several methods have been developed.

Methods derived from the Anti-lock Breaking System can be used for rough terrain rovers. Because they adapt the wheel speeds when slip already occurred, they are referred to as *reactive approaches*. A velocity synchronization algorithm, which minimizes the effect of the wheels fighting each other, has been implemented on the NASA FIDO rover [6]. The first step of the method consists in detecting which of the wheels are deviating significantly from the nominal velocity profile. Then a voting scheme is used to compute the required velocity set point change for each individual wheel. However, performance might be improved by considering the physical model of the rover and wheel-soil interaction models for a specific type of soil. Thus, the traction of each wheel is optimized considering the load distribution on the wheels and the soil properties. Such approaches are referred to as *predictive approaches*.

In [7], wheel-slip limitation is obtained by minimizing the ratio T/N for each wheel, where T is the traction force and N the normal force. Reference [8] proposes a method minimizing slip ratios and thus avoid soil failure due to excessive traction. These physics-based controllers assume that the parameters of the wheel-ground interaction models are known. However, these parameters are difficult to estimate and are valid only for a specific type of soil and condition. Reference [9] proposes a method for estimating the soil parameters as the

robot moves, but it is limited to a rigid wheel travelling through deformable terrain. In practice, the rover wheels are subject to roll on different kind of soils, whose parameters can change quickly. Thus, physics-based controllers are sensitive to soil parameters variation and difficult to implement on real rovers. In this section, a predictive approach considering the load distribution on the wheels and which does not require complex wheel-soil interaction models is presented. More details about the controller can be found in [10].

Quasi-static model

The speed of an autonomous rover is limited in rough terrain because the navigation algorithms are computationally expensive (limited processing power) and for safety reasons. In this range of speeds, typically smaller than 20cm/s , the dynamic forces might be neglected and a quasi-static model is appropriate. To develop such a model, the mobility analysis of the rover's mechanical structure has to be done. It ensures to produce a consistent physical model with the appropriate degrees of freedom at each joints. Then the forces are introduced and the equilibrium equations are written for each part composing the rover's chassis. Because we have no interest in implicitly calculating the internal forces of the system, it is possible to reduce this set of independent equations. The variables of interest are the 3 ground contact forces on the front and the back wheel, the 2 ground contact forces on each wheel of the bogies and the 6 wheel torques. This makes 20 unknowns of interest and the system can be reduced to 15 equations. This leads to the following equation

$$M_{15 \times 20} \cdot \mathbf{U}_{20 \times 1} = \mathbf{R}_{15 \times 1} \quad (1)$$

where M is the model matrix depending on the geometric parameters and the state of the robot, \mathbf{U} a vector containing the unknowns and \mathbf{R} a constant vector. It is interesting to note that there are more unknowns than equations in 1. That means that there is an infinite set of wheel-torques guaranteeing the static equilibrium. This characteristic is used to control the traction of each wheel and select, among all the possibilities, the set of torques minimizing slip. The optimal torques are selected by minimizing the function

$$f = \max_i \left(\sum_i T_i / N_i \right) \quad i = 1..6 \quad (2)$$

where T_i and N_i are the traction and the normal force applied to wheel i .

Rover motion

A static model balances the forces and moments on a system to remain at rest or maintain a constant speed. Such a system is an ideal case and does not include resistance to movement. Therefore, an additional torque compensating the rolling resistance torque must be added on the wheels in order to complete

the model and guarantee motion at constant speed. This results in a quasi-static model. Unlike the other approaches, we don't use complex wheel-soils interaction models. Instead, we introduce a global speed control loop, in order to estimate the rolling resistance as the robot moves. The final controller, minimizing wheel slip and including rolling resistance, is depicted in Fig. 5.

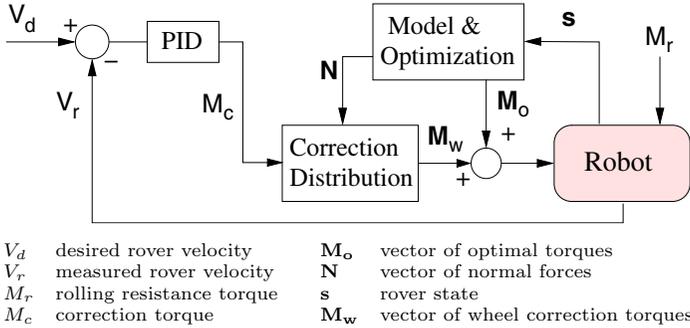


Fig. 5: Rover motion control loop.

The kernel of the control loop is a PID controller. It allows to estimate the additional torque to apply to each wheel in order to reach the desired rover's velocity V_d and thus, minimizes the error $V_d - V_r$. M_c is actually an estimate of the global rolling resistance torque M_r , which is considered as a perturbation by the PID controller. The rejection of the perturbation is guaranteed by the integral term I of the PID. We assume that the rolling resistance is proportional to the normal force, thus the individual corrections for the wheels are calculated by

$$M_{w_i} = \frac{N_i}{N_m} \cdot M_c \quad (3)$$

where N_i is the normal force on wheel i and N_m the average of all the normal forces. The derivative term D of the PID allows to account for non modeled dynamic effects and helps to stabilize the system. The parameters estimation for the controller is not critical because we are more interested in minimizing slip than in reaching the desired velocity very precisely. For locomotion in rough terrain, a residual error on the velocity can be accepted as long as slip is minimized.

Experimental results

A simulation phase using Open Dynamics Engine² has been initiated in order to test the approach and verify the theoretical concepts and assumptions. The

² this library simulates rigid body dynamics in three dimensions, including advanced joint types and collision detection with friction.

simulation parameters have been set as close as possible to the real operation conditions. However, the intent is not to get exact outputs but to compare different control strategies and detect/solve potential implementation problems. In the experiments, wheel slip has been taken as the main benchmark and the performance of our controller (*predictive*) has been compared to the controller presented in [6] (*reactive*). The reactive controller implements speed control (spd) for the wheels whereas torque control (trq) is used in our approach.

Three dimensional surfaces are used for the experiments (see Fig. 6). Because the trajectory of the rover depends on the control strategy, we consider an experiment to be valid if the distance between the final positions of both paths is smaller than $0.1m$ (for a total distance of about $3.5m$). This distance is small enough to allow performance comparison. For all the valid experiments, predictive control showed better performance than reactive control. In some cases the rover was even unable to climb some obstacles and to reach the final distance when driven using the reactive approach. It is interesting to note that the slip signal is scaled down for each wheel when using predictive control. Such behavior can be observed in Fig. 7: the peaks are generally at the same places for both controllers but the amplitude is much smaller for the reactive controller. Another interesting result is that the difference between the two methods increases when the friction coefficient gets lower. In other words, the advantage of using torque control becomes more and more interesting as the soil gets more slippery. Such a controller improves the climbing capabilities of the rover and limits wheel-slip, which in turn improves the accuracy of odometry. This way, it contributes to better position tracking in rough terrain. Furthermore, our approach can be adapted to any kind of wheeled rover and the needed processing power remains relatively low, which makes online computation feasible. Finally, the simulations show promising results and the system is mature enough to be implemented on SOLERO for real experiments.

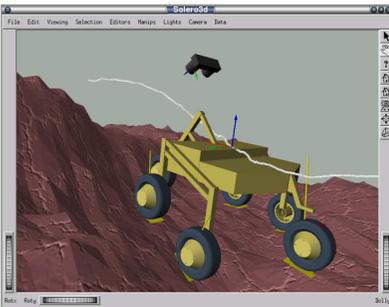


Fig. 6: Simulation environment

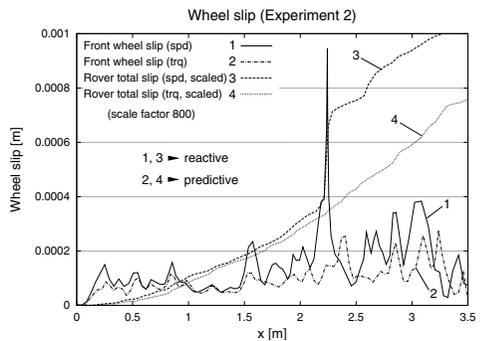


Fig. 7: Wheel slip

5 Sensor Fusion

In our approach an Extended Information Filter (EIF) is used to combine the information acquired by the sensors. This formulation of the Kalman filter has interesting features: its mathematical expression is well suited to implement a distributed sensor fusion scheme and allows for easy extension of the system in order to accommodate any number of sensors, of any kind. Fig. 8 depicts the schematics of the sensor fusion process.

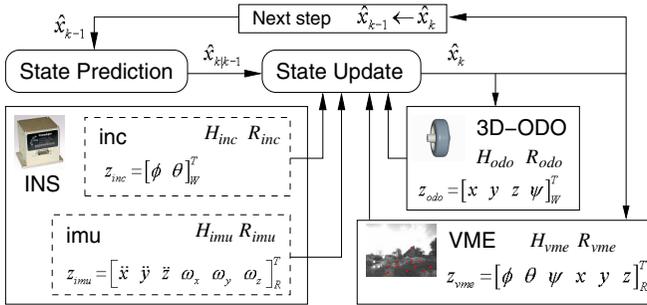


Fig. 8: Sensor fusion scheme

Sensor models: The position, velocity and attitude can be computed by integrating the measurement acquired by the IMU. However, the accelerometers and gyros are influenced by bias errors. In order to limit an unbounded growth of the error of integrated measurements, we have introduced biases in the model for the gyros ($b_{\omega x}$, $b_{\omega y}$, $b_{\omega z}$) and the accelerometers (b_{ax} , b_{ay} , b_{az}). Unlike the roll and pitch angles, the rover’s heading is not periodically updated by absolute data. Therefore, in order to limit the error growth, a special provision is included in the z-gyro model: a more accurate modeling, incorporating the scaling error $\Delta_{\omega z}$.

The robot used for this research is a partially skid-steered rover and the natural and controlled motion is mainly in the forward direction. Thus, the motion estimation errors due to wheel slip and wheel diameter variations have much more effect in the x-z plane of the rover than along the transversal direction y. Therefore, scaling errors Δ_{ox} and Δ_{oz} , modeling wheel slip and wheel diameter change, have been introduced only for the x and z-axes. The error model for the odometry is tedious to develop because the robot is subject to drive across various types of terrains. In order to avoid terrain classification and complex wheel soil interaction modeling, we set the variance of the odometry as being proportional to the acceleration undergone by the rover. Indeed, slip mostly occurs in rough terrain, when negotiating an obstacle, while the robot is subject to accelerations. Similarly, the variance for the yaw angle has been set proportional to the angular rate. More details about the models of

the IMU and 3D-Odometry can be found in [11] and reference [12] presents the error model associated to the estimations of VME.

State prediction model: The angular rates, biases, scaling errors and accelerations are random processes which are affected by the motion commands of the rover, time and other unmodeled parameters. However, they cannot be considered as pure white noise because they are highly time correlated. Instead, they are modelled as first order Gauss-Markov processes. Such modeling of the state transition allows to both consider the time correlation and to filter noise of the signals.

Experimental results

In order to better illustrate how each sensor contributes to the pose estimation and in which situation, the experiments have been divided into two parts. The first part describes the results of sensor fusion using inertial sensor and 3D-Odometry only, whereas the second part involves all the three sensors i.e. 3D-Odometry, inertial sensor and VME.

Inertial and 3D-Odometry: The experimental results show that the inertial navigation system helps to correct odometric errors and significantly improves the pose estimate. The main contributions occur locally when the robot overcomes sharp-shaped obstacles (Fig. 9) and during asymmetric wheel slip. The improvement brought by the sensor fusion becomes more and more pronounced as the total path length increases. More results are presented in [11].

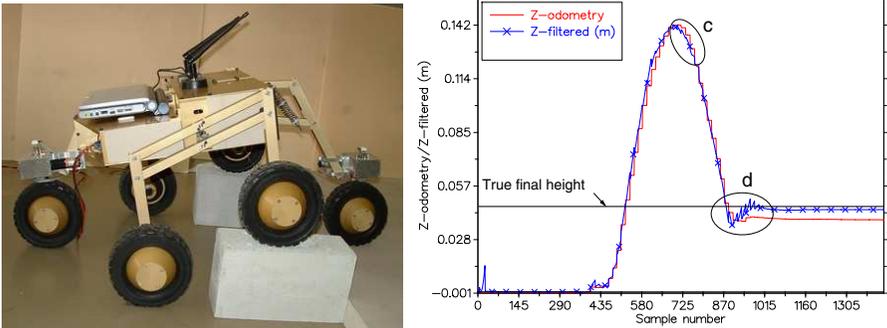


Fig. 9: Sensor fusion with 3D-Odometry and inertial sensors. The ellipses emphasize local corrections of the z coordinate.

Enhancement with VME: In the previous tests, only proprioceptive sensors have been integrated to estimate the robots position. Even if the inertial sensor helps to correct odometric error, there are situations where this combination of

sensors does not provide enough information. For example, the situation where all the wheels are slipping is not detected by the system. In this case, only the odometric information is integrated, which produces erroneous position estimates. Thus, in order to increase the robustness of the localization and to limit the error growth, it is necessary to incorporate exteroceptive sensors. In this application, we use visual motion estimation based on stereovision[12].

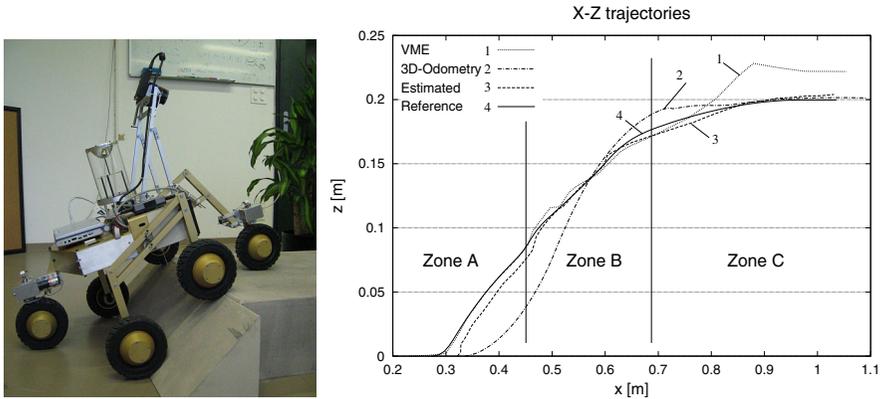


Fig. 10: Sensor fusion using 3D-Odometry, IMU and VME

In general, VME produces better estimates than the other sensors (but at a much slower rate). In particular, its estimates allow to correct the accumulated error due to wheel slip between two updates. However, in zone C (Fig. 10), less than thirty features have been matched between three subsequent images. The difficulty to find matches between these images is due to a high discrepancy between the views: when the rear wheel finally climbs the obstacle, it causes the rover to tilt forward rapidly. As a consequence, VME provided bad motion estimates with a high uncertainty. In this situation, less weight is given to VME and the sensor fusion could perfectly filter this bad information to produce a reasonably good estimate using 3D-Odometry and IMU instead. Finally, the estimated final position is very close to the measured final position. A final error of four millimeters for a trajectory longer than one meter (0.4%) is very satisfactory, given the difficulty of the terrain.

6 Conclusion

This paper showed how 3D position tracking in rough terrain can be improved by considering the specificities of the vehicle used for locomotion. 3D-Odometry produces much better estimates than the standard approach because it takes the kinematics of the rover into account. Similarly, by considering a physical model of the chassis it is possible to minimize wheel-slip,

which in turn contributes towards better localization. In rough terrain, the controller presented in Sect. 4 performs better than a controller based on a reactive approach. Finally, experimental results of sensor fusion involving 3D-Odometry, inertial sensors and visual motion estimation have been presented. They prove that the use of complementary sensors improves the accuracy and the robustness of the motion estimation. In particular, the system was able to properly discard inaccurate visual motion information.

References

1. Strelow D, Singh S (2003) Online Motion Estimation from Image and Inertial Measurements, The 11th International Conference on Advanced Robotics, Portugal
2. Nebot E, Sukkarieh S, Durrant-Whyte H (1997) Inertial navigation aided with GPS information, In the proceedings of the Fourth Annual Conference of Mechatronics and Machine Vision in Practice
3. Siegwart R, Lamon P, Estier T, Lauria M, Piguet R (2000) Innovative design for wheeled locomotion in rough terrain, *Journal of Robotics and Autonomous Systems*, Elsevier, vol 40/2-3 p151-162
4. Lacroix S, Mallet A, Bonnafous D, Bauzil G, Fleury S, Herrb M, Chatila R (2002) Autonomous rover navigation on unknown terrains: functions and integration, *International Journal of Robotics Research*
5. Lamon P, Siegwart R (2003) 3D-Odometry for rough terrain - Towards real 3D navigation, *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan
6. Baumgartner E.T, Aghazarian H, Trebi-Ollennu A, Huntsberger T.L, Garrett M.S (2000) State Estimation and Vehicle Localization for the FIDO Rover, *Sensor Fusion and Decentralized Control in Autonomous Robotic Systems III*, SPIE Proc. Vol. 4196, Boston, USA
7. Iagnemma K, Dubowsky S (2000) Mobile robot rough-terrain control (RTC) For planetary exploration, *Proceedings ASME Design Engineering Technical Conferences*, Baltimore, Maryland, USA
8. Yoshida K, Hamano H, Watanabe T (2002) Slip-Based Traction Control of a Planetary Rover, In the proceedings of the 8th International Symposium on Experimental Robotics, ISER, Italy
9. Iagnemma K, Shibley H, Dubowsky S (2002) On-Line Terrain Parameter Estimation for Planetary Rovers, *IEEE International Conference on Robotics and Automation*, Washington D.C, USA
10. Lamon P, Siegwart R (2005) Wheel torque control in rough terrain - modeling and simulation, *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, in press
11. Lamon P, Siegwart R (2004) Inertial and 3D-odometry fusion in rough terrain Towards real 3D navigation, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan
12. Jung I-K, Lacroix S (2003) Simultaneous Localization and Mapping with Stereovision, *International Symposium on Robotics Research*, Siena

Efficient Braking Model for Off-Road Mobile Robots

Mihail Pivtoraiko, Alonzo Kelly, and Peter Rander

Robotics Institute, Carnegie Mellon University
mihail@cs.cmu.edu, alonzo@ri.cmu.edu, rander@rec.ri.cmu.edu

Summary. In the near future, off-road mobile robots will feature high levels of autonomy which will render them useful for a variety of tasks on Earth and other planets. Many terrestrial applications have a special demand for robots to possess similar qualities to human-driven machines: high speed and maneuverability. Meeting these requirements in the design of autonomous robots is a very hard problem, partially due to the difficulty of characterizing the natural terrain that the vehicle will encounter and estimating the effect of these interactions on the vehicle. Here we present a dynamic traction model that describes vehicle braking on a variety of terrestrial soil types and in a wide range of natural landscapes and vehicle velocities. This model was developed empirically, it is simple yet accurate and can be readily used to improve model-predictive planning and control. The model encapsulates the specifics of wheel-terrain interaction, offers a good compromise between accuracy and real-time computational efficiency, and allows straight-forward consideration of vehicle dynamics.

Keywords: Modeling dynamics off-road robotics

1 Introduction

As developing autonomous off-road vehicle technology allows robots to travel at higher speed and negotiate rugged terrain, vehicle modeling becomes increasingly relevant for motion planning and control. An efficient braking traction model can greatly enhance vehicle autonomy by addressing two key problems: it can determine whether the path ahead, given its slope and ground characteristics, presents risks such as tip-over, and provide a precise estimate of the stopping distance. Precision of the model is very important, but it should also be very efficient computationally because it has to be continually evaluated if it is used for control or tightly coupled with the path planning algorithm. Certainly, a gross over-estimation for the problems above will likely keep the vehicle safe, however in cluttered natural terrain such approach will

either result in slow, inefficient traversal, or may cause a failure of the path planner to generate an admissible path.

1.1 Prior Work

Great overviews of automobile off-road mobility and approaches to soil modeling are presented in [1]. Quite a few fairly detailed models of the wheel-soil interaction were proposed specifically for motion planning applications. For example, [7] and [11] present approaches that model the soil as a mass-spring system. These models provide fairly good results in describing compression, shear and plastic deformations in soils. However, such approaches are yet to be thoroughly validated experimentally. Moreover, the reported run-times of these modeling methods do not appear to be fast enough to render them feasible in real-time robot control scenarios. The approaches that were shown to be suited for controlling mobile robots tend to circumvent the issue of computational efficiency by further simplification. Often the Coulomb principle of friction, or its derivative is used to estimate the amount of rolling friction that the vehicle experiences [8]. Several parameters of the terrain are used in [10] to estimate normal and lateral tangential forces at the wheel contact patch. A similar approach to traction modeling that can also be adapted on-line was presented in [9]. That work is focused on planetary applications with accompanying quasi-static assumptions. It is also assumed that the wheels are rigid. Pneumatic tires used for terrestrial applications, however, are elastic. Moreover, in off-road applications the inflation pressure is typically lower in order to avoid rigid-mode operation that may cause excessive compaction of soil [1]. In off-road robotics it is still common to ignore these effects and consider wheels to be rigid, or simplify even further by using Coulomb friction. We show that at higher speeds and rough terrain, such methods result in grave errors in characterizing braking (e.g. stopping distance). Our approach, however, is as efficient as these approximations employed in the field, but offers much better accuracy, especially at higher speeds.

1.2 A New Approach

We conducted a significant field experimentation effort with autonomous off-road robots, and this prompted an empirical approach to capturing the complexities of wheel-terrain dynamics in natural environments (Fig. 1). An initial observation was that it was generally not possible to consider the net braking force of the vehicle (with gravity effects removed) to be some constant value. In fact, in some cases on soft soil the net braking force on a slope was off by as much as 50% from its value on level ground. Depending on vehicle dynamics, this can result in a miscalculation of the stopping distance by several meters, which may be a serious error when operating in cluttered natural terrain.

We propose an approach that provides accurate estimates of tractive braking force and involves a simple and efficient model of several parameters. The

values of the parameters are determined experimentally by measuring the deceleration during vehicle braking and combining these measurements with vehicle state information. This “training” procedure can be easily done in the field, and even autonomously by the robot. For example, every time the robot has to stop, it can verify its braking model. In this manner, the model can be refined on-line and adapted as the robot moves into different type of terrain. This formulation of the model was shown to work well on off-road robots operating on a wide variety of terrain types, such as clay, soil with sod cover, gravel, coarse sands, and packed snow, as well as at various speeds and on natural slopes (typical to mid-West region, the plains and the desert).

This model can be used in model-predictive control to estimate the *stopping path* [12], the guaranteed stopping distance that is necessary for vehicle safety, which is mainly a function of a complex relationship between vehicle speed, tire-ground interface, and terrain slope. The model can also be utilized by the path planning algorithm to generate plans that respect this stopping path. Since the present model estimates major forces acting on the vehicle during braking maneuvers, it can also be used in kinodynamic motion planning approaches. Moreover, if an estimate of tire sliding friction coefficient is available, then this model can predict whether robot’s wheels are going to lock up (which generally must be avoided [4]).

2 Experimental Procedure

In this section we give the details of experiments that prompted us to formulate this model of braking. In our experiments, a terrain patch that is a good representative of the overall terrain is chosen (often natural environments have fairly uniform type of ground over large areas: meadows, field, desert, etc.). The vehicle accelerates to a certain value of velocity, v_i , and then applies the brakes with some known force (either maximum application for vehicles that have no braking force feedback, or a certain known value for those that do).



Fig. 1. The PerceptOR off-road mobile robot. The experiments were performed with this and similar vehicles.

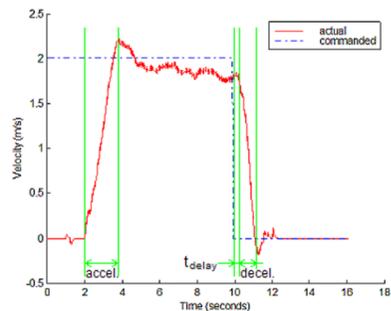


Fig. 2. Deceleration measurement experiment. Dotted line shows commanded velocity, and solid line shows system response.

Most vehicle control systems with closed-loop velocity control estimate velocity more frequently than it can significantly change, so it is possible to achieve the temporal resolution sufficient to obtain the velocity profile of vehicle stopping. The velocity data can be plotted against time as in Fig. 2. Note that actual velocity in the plot goes slightly negative after reaching zero. This is due to expansion of suspension springs that were compressed during braking.

The time when braking was initiated (when desired velocity is set to zero) is recorded, along with the time when the actual velocity reached zero, t_f . The average value of deceleration in a particular experiment is estimated as shown in (1).

$$\bar{a}_x = \frac{v_t}{t_f - t_i} \quad (1)$$

The value \bar{a}_x is the slope of the velocity drop in the figure. In this calculation, it is important to note that, as can be seen in Fig. 2, there is a certain delay after the system commands a zero velocity to when the velocity actually begins to drop. This delay of propagation of the command, t_{delay} , depends solely on hardware. It was on average 300 ms. on our robots. For braking at higher speeds, it is much less than overall Δt , yet needs to be taken into account. Therefore, we take t_i as time of zero-velocity command plus t_{delay} , and sample v_i specifically at that value of t_i to obtain an accurate estimate of the slope.

Throughout our experiments we made sure that the degree of brake engagement was constant. In particular, we were interested in maximum braking, i.e. in engaging the brakes completely.

The same experiment was then repeated with various vehicle velocities and on the ground of various slopes and terrain types. We fitted the above data gathering procedure in the robots' controller code, so that we could obtain a data point at any time when the robot made a stop. In this manner we obtained the data over several months as the robots were used for a variety of navigation and perception experiments on the PerceptOR program. Thus we obtained thousands of data points that were then analysed.

If we plot the measurements of decelerations versus slope for a choice of terrain and subtract the effects of gravity, we see that the resulting net braking force slightly increases with the increase of slope angle. An example plot is presented in Fig. 3, which shows the normalized braking force, a ratio of the net braking force to vehicle weight F_b/W , as a function of slope and velocity. The dependence of deceleration on initial velocity is also noticeable, albeit not as pronounced. Interestingly, these data points exhibit proportional dependence of normalized braking force on slope angle. Hence, a single linear model should be able to predict the braking force for both downhill and uphill braking maneuvers.

Note, however, that our observations have been made in tests on slopes well within limits of vehicle traversability, which was about 17 degree slopes

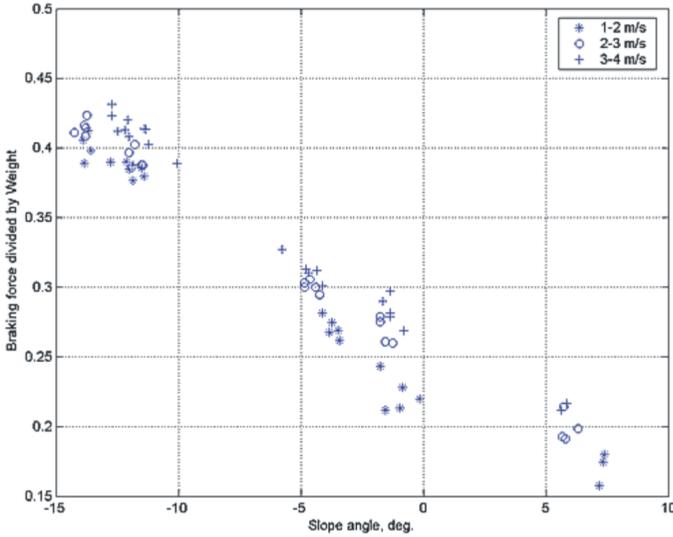


Fig. 3. Normalized braking force, F_b/W , versus ground slope and vehicle velocity.

for our hardware. It is natural to expect that beyond this range of slope values the dependence is no longer linear.

3 Discussion of Results

In this section we develop the necessary concepts to understand the factors influencing traction during vehicle braking. We then use the developed concepts in an effort to explain our experimental observations and suggest a model based on this analysis.

3.1 Vehicle Force Balance

As a starting point, we develop the force analysis of the vehicle during braking. Among the important notions that we discuss here are normal forces on tires, pressure of the tire contact patch, and the dynamic load transfer.

During braking, the major forces acting on the vehicle are related through:

$$F_b = W \frac{a_x}{g} - W \sin \theta \quad (2)$$

Here F_b is the net braking force, g is acceleration due to gravity, a_x is braking deceleration, $W = m_{veh}g$ is vehicle weight, and θ is the terrain slope angle (here we consider downhill slopes as negative, and uphill as positive). The first term on the right side of (2) is the d'Alembert force [2] (see Fig. 4).

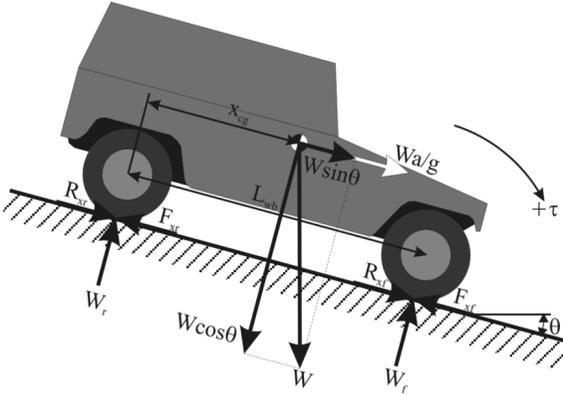


Fig. 4. Free body diagram of a vehicle braking on a slope. Positive torque is assumed clockwise.

Given that the vehicle center of gravity, (x_{cg}, y_{cg}, z_{cg}) , is known, we can express the sum of torques around the contact point of front wheels (for downhill slopes, assuming positive torque is clockwise):

$$-(L_{wb} - x_{cg})W \cos \theta + W_r L_{wb} + z_{cg}W \frac{a}{g} + z_{cg}W \sin \theta = 0 \quad (3)$$

Here L_{wb} is wheel base, and W_r is weight on the rear axle. When the vehicle is stationary on level ground, the loads on front axle, W_f , and rear axle, W_r , are determined by:

$$W_f = W \frac{L_{wb} - x_{cg}}{L_{wb}}; \quad W_r = W \frac{x_{cg}}{L_{wb}} \quad (4)$$

In case of a vehicle decelerating on a slope, we obtain the normal forces on rear and front wheels by summing the torques around front and rear wheel contact points, respectively:

$$W_f = \frac{W}{L_{wb}}(x_{cg} \cos \theta + x_{cg} \sin \theta + z_{cg} \frac{a_x}{g}) \quad (5)$$

$$W_r = \frac{W}{L_{wb}}((L_{wb} - x_{cg}) \cos \theta - z_{cg} \sin(-\theta) - z_{cg} \frac{a_x}{g})$$

We observe from (5) that during braking downhill, there is a significant dynamic load shift from rear to front axles. Note that W_r was written with the $z_{cg} \sin(-\theta)$ term to underscore the fact that for downhill slopes $\theta < 0$.

We consider pressure on the tire contact patch for front and rear wheels as the ratio of axle load to contact area. The vehicles we had available for experiments in this study had dual rear tires, so we estimate that the pressure of front tires' ground contact was twice that of rear tires for the same normal load.

3.2 Braking Force

We consider that the braking torque results in a longitudinal force F_h at the wheel-terrain interface. Since the goal of this work was to understand the effects of maximum braking that determines minimum allowable stopping distance and outlines the upper bound on dynamics effects due to braking, we understand that F_h represents full engagement of the brakes and depends solely on braking hardware, hence always constant. Here we also assume that braking happens on a straight path. We visualize the effect of this force in the detail of interaction of an off-road tire with terrain in Fig. 5.

The hardware braking force F_h is counter-acted by the terrain acting on tire tread. If the magnitude of this force exceeds the shear strength of the terrain, it will no longer be able to resist this shear force, and the wheel will skid.

The other force in the tire-ground interface that was found to have significant effect on braking is rolling resistance R_x . This resistance is always present, and in the case of pneumatic tires its value is determined by many factors, such as tire material and design, temperature, vibration, pressure of the ground contact patch (normal force on the tire). Terrain compaction (related to pressure of the patch) and bulldozing effects in soft soil are also important contributing factors to this resistance [1]. While F_h can be considered constant for a given vehicle, estimating R_x is complicated due to the variety of factors influencing it.

Through experimentation we found that we can approximate all longitudinal forces acting on the vehicle during braking by lumping them into the sum of the force due to the torque supplied by the braking hardware, and the rolling resistance. Then, the overall braking force is considered to be:

$$F_b = F_h + R_x \quad (6)$$

The key to accurately predicting the braking force is estimating rolling resistance R_x .

In our experiments it was also determined that out of all factors influencing rolling resistance, the most significant one is the pressure at ground contact. A lesser, but noticeable, effect has vehicle speed. In the following two sections we explain these two factors.

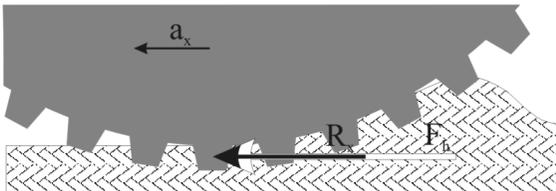


Fig. 5. Detail of wheel-terrain interface. We focus on two dominating forces: braking traction force F_h and rolling resistance R_x .

3.3 Effect of Terrain Slope

It is important to consider contact pressure here because in general rolling resistance is roughly proportional to this pressure (although this relationship is complex and highly non-linear) [1].

For the case of level ground we can decompose (6) into the contributions of front and rear wheels:

$$F_b = 2F_h + R_{xr} + R_{xf}$$

Here F_h is the same for front and rear wheels since our vehicles had the interlocked differential. Also our robots had dual rear tires, which resulted in twice the contact area and half the ground contact pressure for rear tires than for the front tires. Hence, let us suppose (only for clarification purposes in this section) that due to the difference in contact pressures, the rolling resistance values can be related through $R_{xf} = 2R_{xr}$.

As was shown earlier, during downhill braking there is a significant dynamic load shift to front wheels, $W_r < W_f$. Because of this the pressure developed at front wheel contact point greatly exceeds that at rear wheel contact, and even more so in the case of rear dual tires. R_{xf} increases dramatically, more than R_{xr} decreases (in part due to half the contact pressure). The overall value of F_b becomes greater than on level ground.

During braking uphill, similar issues come into play. However, in this case the load shift to front axle is less significant (see (5)), in fact even less than on level ground due to $x_{cg} \sin \theta$ term. In this case $W_r > W_f$, whereas for level ground we had $W_r = W_f$. However, since rear tires have nearly “half the effect” on rolling resistance than the front tires, the overall braking force is less than on level ground.

3.4 Effect of Velocity

Among the factors influencing rolling resistance is vehicle velocity [2]. The rolling resistance is directly proportional to velocity because of increased tire deformation work and vibration in the tire. The influence of velocity becomes more significant when tires with lower inflation pressures are used, as is often the case for off-road vehicles. Lower tire pressure is used to allow tires to be more elastic, since the work required for flexing the tire is much less than the work of compacting and bulldozing soft soil. Greater elasticity, however, causes greater hysteresis losses with increasing vehicle velocity. The effect of velocity on rolling resistance was found to be less significant, but still noticeable.

4 Deriving the Model

In this section we combine our experimental observations with the insights developed above to formulate our model of braking force. We describe how

this model could be easily adapted online and discuss the results of validating the model through experiments with robots.

4.1 Formulating the Model

As we discussed, the results of our experiments prompted us to make a simplifying assumption that within the range of slope values that the vehicle can safely handle, the braking force is proportional to the slope.

The essence of our model is stated as:

- The braking force (without gravity effects) can be approximated well by a linear model:

$$\frac{\partial F_b}{\partial \theta} = m$$

where θ is ground slope angle and m is a coefficient. We can fit a line $F_b = m\theta + b$ to the test data in the least-squares manner and use it to obtain future estimates of F_b based on slope.

- The coefficients m and b above also exhibits linear dependence on initial velocity of the vehicle (right before braking is initiated):

$$\frac{\partial m}{\partial v_i} = m_m; \frac{\partial b}{\partial v_i} = m_b$$

Thus, the overall model contains only four parameters: m_m, m_b, b_m, b_b :

$$m(v_i) = m_m v_i + b_m; \quad b(v_i) = m_b v_i + b_b \quad (7)$$

So that

$$F_b = m(v_i)\theta + b(v_i) \quad (8)$$

We again underscore that the development of the model was based on experimental data, which was available for a range of terrain slope roughly from -15° to 15° . While this model cannot be extrapolated outside the experimental range in which it was defined, we can reason about the character of F_b outside of this range. In particular, based on previous discussion, we estimate that for greater uphill slopes, the effect of rolling resistance will diminish due to decreasing normal force, i.e. contact pressure, and F_b will approach F_h (omitting gravity effects, as usual). At a certain point the slope becomes unsafe, when the shear capacity of the wheel-terrain interface becomes equal to F_h . For steeper downhill slopes similar arguments apply: rolling resistance will become less dominant with decreasing soil contact pressure, and at some point the shear capability may no longer support the vehicle.

In the experiments that lead to formulation of this model, we have assumed that the degree of application of the brakes was constant throughout the experiments (e.g. for emergency braking, which often determines the look-ahead distance for a path planner, maximum actuator power is used). For

other actuator modes this model is also applicable, but additional coefficients may be necessary to allow for other than maximum braking (e.g. slight, half way, etc.). On the other hand, the benefits of this expression of the model are that it is very simple and intuitive, quite easy to adjust, yet powerful enough to account for peculiarities of braking hardware and ground types, while requiring very low online computational overhead.

4.2 Experimental Results

As we saw in the previous section, the key result of the model, F_b , is obtained simply by evaluating three linear equations (8). Thus, its computational cost is minimal and comparable to the fastest approximations used in the field (e.g. using simple Coulomb friction equations). Such methods, however, do not consider the dependence of braking on changes in wheel-terrain interface due to velocity and slope, and thereby result in large errors. The PerceptOR robots previously used similar methods to estimate stopping distance, and it was routinely over-estimated with error on the order of 50% (stopping distance estimates lower than the actual value must be avoided, as they may lead to collision).

Our model was verified through a series of experiments: braking on level ground, downhill and uphill, at velocities ranging from 1 to 4 m/s, and with 10 repetitions of each test to ensure correlation (variability of measurements was less than 5%). Figure 6 a) shows the results of 200 such experiments (the horizontal axis represents test number), where the stopping distance was estimated using the model (as discussed in Section 5.1) and compared to actual measurements. Given predicted and actual values of stopping distance, statistical analysis was performed on the error of this braking model. Figure 6 b) presents the plot of mean values of this error as a function of vehicle velocity (before braking). Note that the model always slightly over-estimates

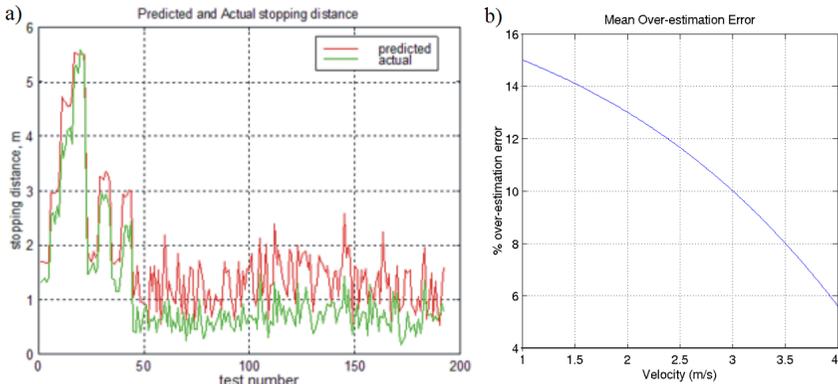


Fig. 6. a) Comparison of predicted and actual stopping distance. b) Experimental validation of the model: percent error in stopping distance as a function of vehicle velocity. The model must over-estimate for safety.

the stopping distance as desired. The variance of this error was about 2% throughout the velocity range in Fig. 6 b).

5 Applications of the Model

The main motivation for estimating a braking model was the determination of the stopping distance. Here we provide formulas for computing both stopping time and distance. Another important application of the model is estimation of vehicle dynamics during braking. Once the balance of forces is known, it becomes possible to answer questions about whether a particular slope is viable for the vehicle (e.g. in terms of tip-over hazard).

5.1 Estimating Stopping Distance and Time

Average braking deceleration a_x can be obtained from (2) once we have an estimate for braking force F_b . Since deceleration is negative change of velocity over time, we estimate the stopping time, t_s , given deceleration, a_x , as:

$$t_s = \frac{v}{a_x} + t_{delay} + t_{offset} \quad (9)$$

where t_{offset} is an offset to ensure that the result is always somewhat overestimated in order to keep the vehicle safe.

Similarly, stopping distance, s_s , is calculated based on the fact that deceleration is the second time derivative of distance:

$$s_s = \frac{1}{2}at_s^2 + t_{delay}v + s_{offset} \quad (10)$$

where s_{offset} is a similar distance offset.

5.2 Predicting Vehicle Tip-Over Condition

Calculating tip-over condition involves finding the sum of torques around the point of contact of front wheels of the vehicle (refer to (3)) and Fig. 4. To find the threshold where the vehicle will start to tip over, we need to find when the weight acting on the rear axle, W_r , vanishes. However, practically the vehicle will be in danger even before this condition occurs. When W_r becomes low enough so that the sliding friction force caused by it becomes equal to the braking force, rear wheels will start sliding and a loss of directional stability will occur [4]. To find a more suitable estimate of maximum allowed slope angle, we have to solve the equation (3) for θ so that $|W_r|$ is relatively small. Small angle approximations could be utilized to simplify this solution. The resulting θ is the maximum allowable pitch angle of the vehicle to prevent tip-over, given the longitudinal location of its center of gravity and other parameters. Since most of the components of equation (3) can be precomputed in advance, the estimation presents low computational overhead.

6 Conclusion and Future Work

We presented an empirical braking model that is very simple to estimate, yet produces quite accurate results that exhibit appreciably small errors in a very wide variety of off-road operation: high and low speeds, level ground and steep slopes that high-traction vehicles can negotiate. The model can also be extended with more analytical approaches that utilize estimation of soil sinkage and other peculiarities of navigating over soft, soils and sands. Also, popular tire models can be utilized for operations on hard surfaces. Our future work will involve testing the model on vehicles that can operate at much higher speeds and steeper slopes. We would also like to extend this study to maneuvers including steering while braking and accelerating (speeding up as well as slowing down). We hope to look into the application of more powerful learning techniques to adapting the model to the variety of natural terrain. Although the specifics can vary and further build on the simple empirical model, the spirit remains the same: a run-time characterization of vehicle dynamics that can support many intelligent decisions on the part of a vehicle path planning system operating in unpredictable off-road environments.

References

1. Bekker M (1969) Introduction to terrain-vehicle systems. University of Michigan Press
2. Gillespie TD (1992) Fundamentals of vehicle dynamics. SAE.
3. Terzaghi K (1943) Theoretical Soil Mechanics. Wiley, New York
4. Wong JY (1993) Theory of ground vehicles, 2nd ed. Wiley, New York
5. Andrade G et al. (1998) Modeling robot-soil interaction for planetary rover motion control. In: Proc. of the Int. Conf. on Intelligent Robots and Systems, Victoria, B.C., Canada
6. Chanclou B, Luciani A, Habibi A (1996) Physical models of loose soils dynamically marked by a moving object. In: Proc. of the Int. Conf. on Robotics and Automation
7. Chanclou B, Luciani A (1996) Physical modeling and dynamic simulation of off-road vehicles and natural environments. In: Proc. of the Int. Conf. on Intelligent Robots and Systems
8. Iagnemma K et al. (1999) Experimental validation of physics-based planning and control algorithms for planetary robotic rovers. In: Proc. of the Int. Symposium on Experimental Robotics
9. Iagnemma K, Shibly H, Dubowsky S (2002) On-line terrain parameter estimation for planetary rovers. In: Proc. of the Int. Conf. on Robotics and Automation
10. Jain A et al. (2004) Recent developments in the ROAMS planetary rover simulation environment. In: Proc. of the IEEE Aerospace Conf.
11. Luciani A, Chanclou B (1997) Physical models of off-road vehicles moving on loose soils. In: Proc. of the Int. Conf. on Intelligent Robots and Systems
12. Kelly A, Stentz A (1998) Rough terrain autonomous mobility - part 2: an active vision, predictive control approach. *Autonomous Robots* 5:163-198

Autonomous Excavation Using a Rope Shovel

Matthew Dunbabin and Peter Corke

CSIRO ICT Centre, PO Box 883 Kenmore QLD 4069, Australia

Summary. This paper describes automation of the digging cycle of a mining rope shovel which considers autonomous dipper (bucket) filling and determining methods to detect when to disengage the dipper from the bank. Novel techniques to overcome dipper stall and the online estimation of dipper “fullness” are described with in-field experimental results of laser DTM generation, machine automation and digging using a 1/7th scale model rope shovel presented.

1 Introduction

In previous mining excavation work[1] we have concentrated on the non-digging part of the cycle, the dragline’s dig-dump-dig motion. In this paper we describe early work on automating excavation using a rope shovel. These shovels are high-value mining assets which play a significant role in coal uncover and production, but are not achieving optimum performance. Problems include operator variability, truck positioning and timing, and sub-optimal digging paths which can result in either partly filled buckets¹ or a time-consuming stall of the machine. Automation offers the potential to factor out this variability and to provide consistent and optimised performance. Further benefits will accrue through less damage to the machine through “smarter” operation during both dig and swing by eliminating overloads and collisions with crawlers and trucks.

Robotic excavation has been investigated by a number of authors on different machine types but focused around digging, weight estimation and motion planning. Singh[11] provides a good review of the field and discusses state-of-the-art in sensing and machine/ground interaction models. He then uses a number of implemented systems as examples to illustrate different levels of autonomy: teleoperation, trajectory control, tactical and strategic planning. In other related work Singh[10] he describes early research on predicting excavation forces for the general case of a flat blade moving through soil, as well as developing predictive models of excavation forces using basis functions which are parameterizations of prior terrain and bucket path.

¹ The shovel’s bucket is called a dipper.

A significant body of relevant work was conducted at Carnegie-Mellon University by Singh, Cannon, Rowe, Stentz and others in the 1990s. The work was sponsored by Caterpillar and resulted in a considerable number of patents related to automation, motion planning and terrain profiling of a backhoe type excavator which is digging a bench and loading dirt into a truck [12] and the use of two laser scanners to monitor the face and the truck independently [13]. Parameterized scripts[8] are used to describe excavation sequences, and learning control is used to optimize the parameters to maximize productivity defined in terms of some metric which incorporates volume and time.

Two papers by Hemami discuss the loading of front-end loader buckets. In [6] the case of particulate matter is discussed. Five forces acting on the bucket are identified: (1) Weight of bucket and material in the bucket, (2) Compression of material in front of and below the bucket, (3) Friction between material entering the bucket and bucket walls, (4) Cutting resistance of the material at the teeth, and (5) Weight of material displaced by the bucket. He then develops a simulation based on a non-linear model of cutting force as a function of geometry. In [5] this theory is elaborated. Forces 1, 4 and 5 are identified as dominating, and 2 can be made zero by appropriate choice of bucket path. The rest of the paper considers the kinematics of the bucket and designs trajectories that would minimize force 2. This work was entirely theoretical.

Bernold[2] describes experimental work for the case of scooping in soil implemented using a small robot equipped with force/torque sensor and scoop. He compares position control and impedance control and finds, as expected, that for trenching excavation, using impedance control is a function of the soil properties. Shi et al.[9] discuss the problem of front-end loader bucket filling using fuzzy logic based on bucket force/torque data.

In terms of motion planning for digging, Patent US6523765 describes an automation system for a large excavator in which pretaught points are interpolated to generate a path. The methods it describes are fairly conventional robotic path planning. It has no ability to adjust its motion based on changes to the terrain. As part of our project, a comprehensive evaluation of the autonomous excavation research and OEM systems with their benefits and limitations is presented in [3].

In this paper we present a methodology and experimental results for bucket filling and determining when to disengage the dipper from the bank. We also present a laser mapping system which provides information for automatically engaging the bank. The work has been conducted using a 1:7 scale-model rope-shovel.

2 Shovel Overview

2.1 Machine

The model rope shovel used in this investigation is shown in Fig. 1. The model is unconventional looking due to the “kink” in the boom, however all the critical dimensions of the shovel: pivot height, hoist sheave position, crowd arm length and dipper size are accurately scaled from production shovels such as the P&H 4100.

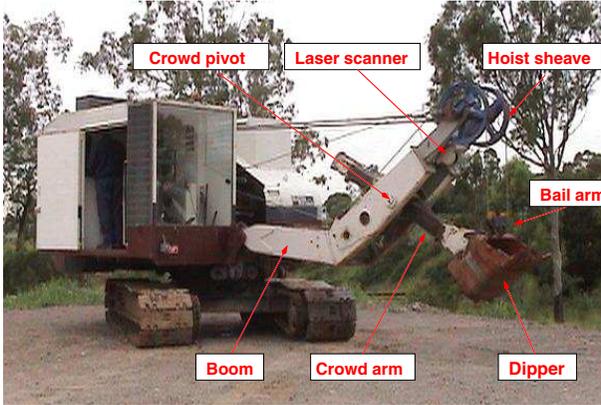


Fig. 1. The model shovel.

Rope shovels have two control inputs to perform digging: crowd and hoist extensions. A change in crowd extension alters the length of the crowd (dipper) arm tangentially from the dipper to the crowd pivot point. Hoisting moves the dipper vertically which in turn rotates the dipper arm about the crowd pivot point. As part of this project, the machine was made capable of computer control which allows autonomous control of the shovel via a single network connection to a laptop computer.

In addition to the implementation of computer control, the only other inclusion to the shovel was a SICK LMS laser scanner. In [7] we demonstrated that a laser-based scanning system can be used to generate high quality Digital Terrain Maps (DTMs) of the area around a BE 1350 dragline including the spoil piles. The DTMs are geo-referenced to the mine coordinate system and can be used for production measurement, reconciliation and as an assistance for digging to plan.

The laser scanner used in this project has a maximum range of 50m and scans over 100 degrees at 0.25 degree intervals. The laser scanner was placed on the boom of the model shovel just below the hoist sheave as indicated in Fig. 1. The base of the laser is approximately parallel to the boom and its field-of-view allows us to see well above the maximum hoist height, as well as the ground close in to the shovel.

2.2 Control Structure

The computer control structure is shown in simplified schematic form in Fig. 2. The three Baldor motor drive controllers and the encoders are connected to the computer via Momentum data acquisition modules and the onboard local area network.

The control computer is a laptop running Fedora Linux. Three programs are involved, connected by an in-house developed “blackboard” data structure [4] through which they share information:

1. *controller*: ensures crowd and hoist extension achieve the desired value and updates the drive states (extension, current, voltage) on the blackboard at 10Hz.

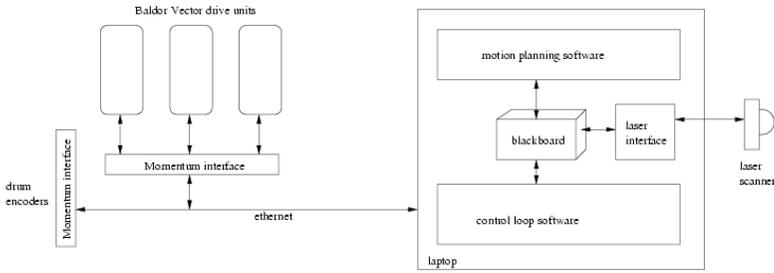


Fig. 2. System architecture.

2. *laser interface*: reads data from the SICK LMS laser scanner and places it on the blackboard.
3. *motion (task) planning*: uses information from the blackboard to plan the path that the bucket teeth should follow at a rate of 5Hz.

Kinematics

The mathematical transformation between crowd and hoist extension, and the Cartesian coordinate of the dipper tooth is a problem in kinematics. The transformation is non-linear and depends strongly on various parameters of the rope shovel.

Kinematic equations were derived based on the geometry of the rope-shovel including minimum crowd extension and hoist rope length, position of the hoist sheave, position of dipper teeth with respect to the crowd arm, offset of crowd arm centre line from the crowd pivot point, and the position of hoist bail arm with respect to the crowd arm. Forward kinematics convert the crowd and hoist extensions to a dipper tooth coordinate, and the inverse kinematics determine the required crowd and hoist extensions for a given dipper tooth coordinate which are the inputs to the control loops. These functions are described in detail in [3].

Control loops

Each of the model rope shovel's three motion axes (crowd, hoist and swing) were made capable of automatic control during this project. The crowd and hoist software level controllers are based on PD loops around position with velocity feed-forward. The swing drive is torque controlled at the motor level, and therefore it is necessary to close the velocity and position loops separately with acceleration feed-forward used to improve tracking performance. In this project, integral action was avoided due to possible integral windup during digging and dipper stall which can cause instabilities in the controller. The main control loop was run at 10Hz.

The tracking performance (shown in [3]) of each of the shovel's controlled axes (crowd, hoist and swing) was optimized by tuning each control gain on-line to achieve the desired tracking response to rapid step position set-point changes.

3 Laser-Based Self Calibration

In order to know where the dirt is with respect to the shovel, we need to know where the laser is with respect to the shovel and the geometric parameters of the shovel for the kinematic models. Various approaches were investigated to determine the actual geometric parameters of the shovel such as: (1) taking measurements using tape measures and levels, (2) taking dimensions from a CAD model and (3) scaling dimensions from a digital photographic side view. All these processes gave somewhat different numbers. Variations in the length of the stays when the boom is removed and reattached² means that the whole boom/dipper/laser assembly can be rotated by several degrees. This corresponds to a considerable error in height over the length of the dipper handle (crowd arm).

A novel method was developed to auto calibrate these key kinematic parameters. A self-calibration process was established that uses the laser to estimate the dipper handle angle and the position of the hoist rope pivot, and compares this with what would be expected from measured extensions and the kinematic model. Fig. 3(a) shows a typical laser scan where the laser measurements have been plotted in Cartesian coordinates. To simplify the analysis, we rotate all the points so that the dipper handle is horizontal and passes through the origin, see Fig. 3(b), and heuristics estimate the top of the bail arm.

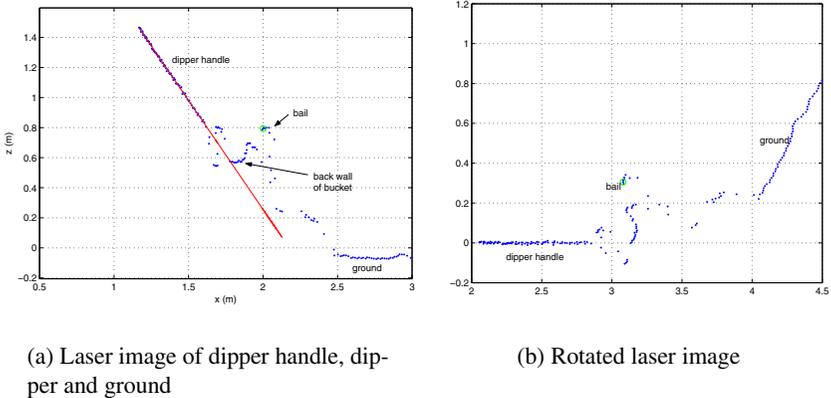


Fig. 3. Laser-based self calibration.

A numerical optimization procedure is then used to minimize the error between the laser observation and the prediction based on measured crowd extension and hoist rope length by adjusting the kinematic parameters. Using this approach we are able to estimate the position of the laser as well as hoist and crowd offset and sheave position.

² As occurred during our experimental program.

4 Laser-Based Terrain Mapping and Digging

Currently it is not possible to ‘measure’ on-line the movement of dirt by an excavator. Techniques commonly used to measure production and their limitations are:

1. Weighbridge to measure the load in each truck. However the weight of material does not necessarily reflect the original volume of material that was removed.
2. A production monitor can provide the approximate weight of each dipper. This is a difficult estimation problem that is still problematic for draglines, excavators and shovels, especially whilst the machine is engaged and digging the soil.

Online laser measurement of the dipper filling process holds the promise of being able to:

1. *detect where to initiate digging (engage point),*
2. *detect when the dipper is full in order to initiate disengage,*
3. determine when to move the base,
4. estimate the change in shape of the bank with time to allow measurement of the *real* productivity of a shovel operation which can not be measured with conventional methods.

The first two purposes were explicit objectives of our project.

4.1 Terrain Mapping

The laser scanner was mounted so that its scanning plane was vertical and included the boom axis. As the shovel rotates, the scanning line is rotated over the terrain. Fig. 4 shows a typical ‘point cloud’ data set containing all the scanned points generated by rotating the shovel through 360 degrees. The terrain map contains sufficient detail to enable digging, obstacle avoidance as well as volume estimation. The dipper has been retracted so that it does not intrude into the laser’s field-of-view.

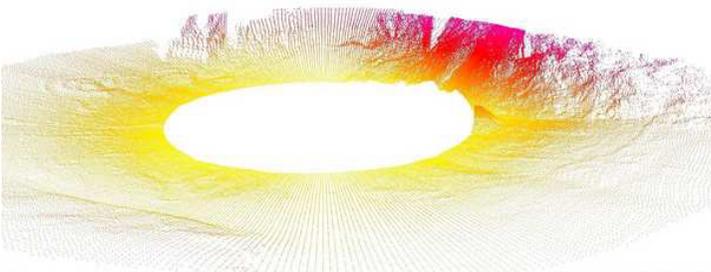


Fig. 4. Raw 3D point cloud data (Colour indicates height, red is high, yellow is low).

5 Autonomous Digging

The critical steps in autonomous shovel digging/excavation are considered to be: (1) planning the dipper trajectory through the bank of dirt, (2) detecting when the bucket is full, and (3) detecting dipper stall and “unstalling” of a stalled drive.

5.1 Dipper Stall Detection

If the onset of a stalled drive can be detected, reliable and repetitive automated dig cycles can be performed by appropriate dig path modifications to overcome the stall.

The stalling of a drive axis (crowd or hoist) is characterised by an increase in drive current with little or no motion along that axis. Since the crowd and hoist drives are velocity controlled, if the tracking error on the drive increases (ie. the drive cannot maintain desired velocity) the current increases to try and overcome this deficiency. In this analysis, it was determined that a reliable prediction of the onset of stall on one or both drive axes could be obtained by observing when the current exceeds a preset threshold and setting a flag in the software to indicate that a drive has stalled.

5.2 Stall Path Modification

If no corrective action is taken to “unstall” the dipper, the motors could over-heat and / or the motor safety protection could be tripped which may shutdown the drive. If unable to break the stall, the dipper may remain stationary rendering the system useless.

Dipper stalling was addressed in a number of ways with the most effective technique termed “crowd retract” and consisted of a three part correction process:

1. detection of a stalled drive.
2. slow down the planned motion.
3. retraction of the crowd arm whilst maintaining desired hoist control.

After the detection of a stalled drive, the second step of stall mitigation consists of simply slowing down the progress of the motion planner. It was observed during testing that most stalling occurred on the crowd motor, that is, the crowd is trying to push into the bank with not enough hoist command to break the stall. To overcome this “pushing” into the bank, a “crowd retract” feature was implemented at the control loop level to reduce the crowd push. The basis of the crowd retract is that when a stall is detected, the crowd tracking error is reduced by modifying the task level demanded crowd position at the control level. This has the effect of reducing the “push” into the bank and allowing the hoist to move the teeth up the bank. Once the motor is unstalled (ie moving again) the demanded crowd position modification is removed and normal tracking continues. The crowd retract procedure is summarised in Algorithm 1.

Fig. 5 shows the performance of the stall detection and “crowd retract” path modification implemented in this project during an aggressive digging cycle. The

Algorithm 1 Stall detection and crowd retract procedure.

1. At start of dig, initialise the retract counter to zero (ie $R = 0$).
2. On detection of a stall event increment the retract counter by a fixed amount ($\eta_{stalled}$) such that $R = R + \eta_{stalled}$.
3. The crowd demand from the task level software (q^*) is modified by subtracting a scaled value of the retract counter such that the demand used by the control level software is given by $q^*_{control} = q^* - K_R \cdot R$ where K_R is a scalar gain.
4. If $R > 0$ and no stall is detected then reduce the retract counter such that $R = R - \eta_{unstalled}$.
5. If at end of dig or dipper is full, end, else go to step 2.

top two traces show the desired and actual crowd and hoist positions with the third and fourth traces showing the stall detection flag (value is 1 if stall is detected) and the retraction counter, R , respectively. As seen from the first trace on crowd, the task level crowd extension is reduced by the control level due to stall detection and the increase in the retract counter, R .

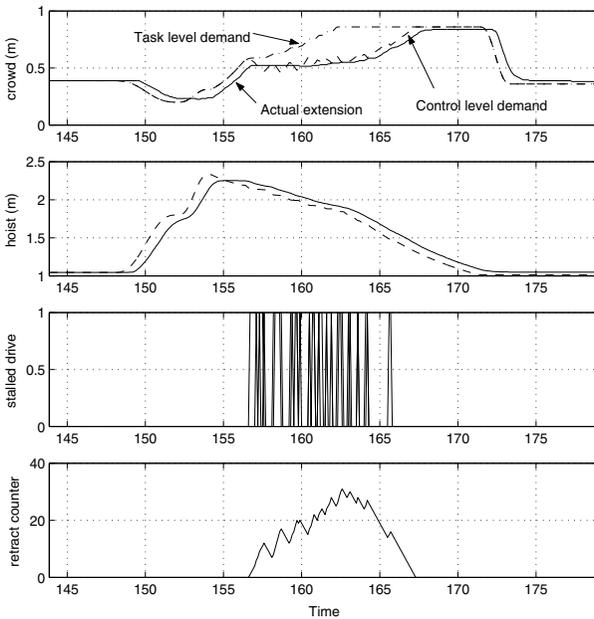


Fig. 5. Results of “crowd retraction” to reduce stall in the automated dig cycle (solid line is actual and dashed line is demanded response).

This stall detection and crowd retraction technique performed extremely well during testing, giving consistent dig cycle time performance and reliable dipper fill-

ing. The technique allows for the automated system to respond to disturbances and unknowns in the digging cycle such as buried boulders.

5.3 Digging

The automated digging cycle is based on observation of human operation of the shovel which consists of four dipper trajectory segments:

1. Move to the start of the dig, aiming to keep the dipper tucked up close to the boom as it approaches the ground (Known as Dig state 2. Note that Dig state 1 is a system ready state).
2. Move forward approximately parallel to the ground (Dig state 3).
3. Move forward and raise the dipper through the bank in a straight line at a predetermined angle until it encounters the crowd limit, at which point it becomes an arc (Dig state 4).
4. Retract horizontally once the dipper reaches maximum height (Dig state 5).

The motion planner based on the trajectory segments above was used to generate crowd and hoist position demands to perform digging. These trajectory segments can be seen in Fig. 6 which shows the demanded and actual crowd and hoist extensions converted to Cartesian coordinates at the dipper teeth. In this particular dig cycle, it can be seen that the dipper follows the demanded trajectory until the lower left corner when the crowd limit is approached. Then the dipper penetrates and moves up the dig face. In this dig cycle, stall detection and crowd retract is implemented which modifies the dipper path in the zig-zag fashion shown to unstall the dipper.

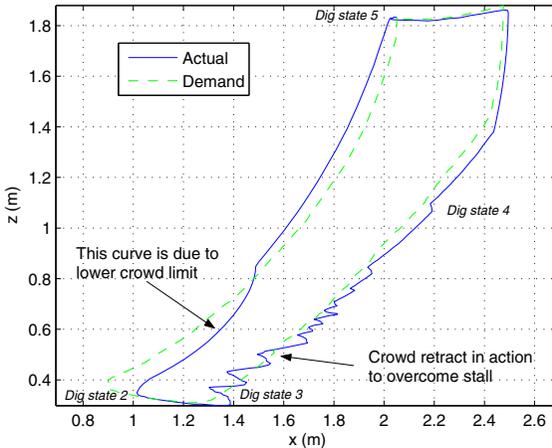


Fig. 6. Dipper teeth trajectory during a dig cycle with stall detection and crowd retraction.

Fig. 7 shows the results of a typical deep/aggressive dig cycle which completely fills the dipper. Here the tracking performance of the machine is very good, however, the currents can be seen to ramp up very quickly at the start of state 4 due to the dipper teeth engaging the bank. These hoist and crowd currents indicate the amount of energy which has been required to push the dipper through the bank. Despite the large current variation, stall was not detected during this dig cycle. A complete set of dig cycle profiles for shallow, moderate and aggressive digs are given in [3].

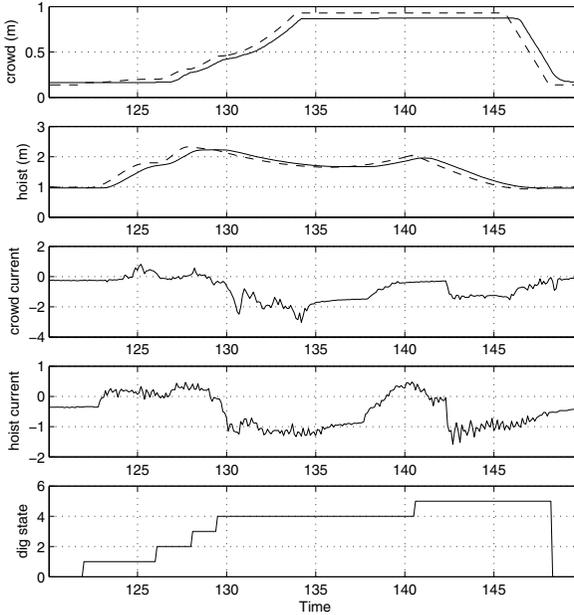


Fig. 7. Deep cut digging results (solid line is actual and dashed line is demanded response).

5.4 Detecting Dipper Fullness Using Motor Signals

Identifying a full dipper allows the motion planner to disengage from the bank earlier thereby reducing the cycle time and energy that would normally be used to complete an entire preplanned digging cycle. However, identifying a full dipper using only the information available whilst the machine is performing a dig cycle is a difficult task.

Two techniques were developed which could reliably predict the “fullness” of the bucket. The first, and most promising, was based on laser profiling of the dirt height across the centre of the dipper during digging. This technique is complicated due to the laser not being able to see directly into the dipper, however, as the dirt height becomes significant it is a reliable indicator of how the dirt is flowing into the dipper. A dipper dirt height sequence during a representative dig cycle is shown in Fig. 8 where the build up of dirt can clearly be seen as the dipper moves up the bank.

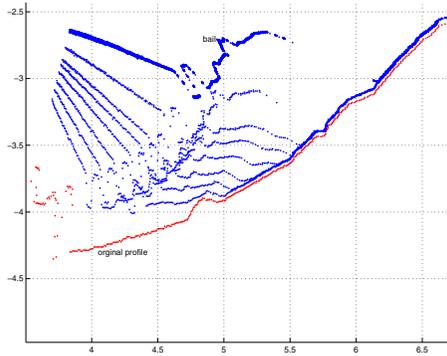


Fig. 8. Overlaid laser measurements of dirt in the dipper during a typical dig. Dipper is moving from left to right.

Using the same approach for laser-based parameter estimation (Section 3), by observing the dirt height in front of the bail a “fullness” index was developed. This index was compared against a predetermined threshold and proved a reliable indicator of the bucket fullness.

The second approach was based on estimating the power at the dipper tooth. It was observed experimentally that tracking performance does not appear to change significantly based on dipper “fullness”, however, the current and hence motor power vary depending how hard the dipper is penetrating the bank. This is consistent with the proposition that as the dipper gets full, the drives must work harder.

The tooth power was obtained by estimating the force at the dipper teeth based on the measured motor currents and resolved in the Cartesian velocity direction of the teeth using the kinematic equations for the shovel. By comparing the tooth power and the digging energy against predetermined thresholds, the technique was shown to give reasonable predictions of dipper “fullness” when compared to the laser dirt profiling described above. However, further investigation is required to examine the validity of the approach under different digging regimes and shovels.

6 Conclusions

This paper has presented the successful methodology and experimental results in the progression toward full-scale rope shovel automation through the demonstration of a fully autonomous digging/excavation cycle on a 1/7 scale-model shovel. We have addressed and presented solutions to this critical first step by demonstrating automatic dipper filling and methods to detect when to disengage the dipper from the bank when the dipper is full. We have also demonstrated a laser mapping system which provides information for automatically engaging the bank, as well as potentially providing a situation display for the operator. The system was found to operate reliably and gives consistent performance similar to human operators.

The proposed algorithms have been devised based on human observations and implemented in such a manner that they are considered scale independent and transferable to production shovels. The primary limitation of this current system is the nature and variability of material being excavated (e.g. in poorly blasted material there are often boulders on the scale of dipper size). Filling a bucket in this type of material is not considered possible at this stage. Future work will advance rope shovel automation by including autonomous truck identification and material loading.

Acknowledgments

This project was funded by the Australian Coal Association Research Program (ACARP) (Project C12030) and CSIRO ICT Centre. Thanks go to Craig Smith of Leslie Consulting, our model shovel operator, and the ACARP industry monitors, Andy Davidson (BMA) and Nick Taylor (RioTinto). Also we would like to thank the rest of the CSIRO robotics team: Leslie Overs, Pavan Sikka, Stephen Brosnan, Graeme Winstanley, Jonathan Roberts, Elliot Duff and John Whitham.

References

1. Dragline Swing Assist: ACARP C9028 - final report. Technical report, ACARP, 2003.
2. L. Bernold. Motion and path control for robotic excavation. *ASCE J. Aerospace Engineering*, 6(1):1–19, January 1993.
3. P. Corke and M. Dunbabin. ACARP Project C12030 Rope shovel automation. Technical report, ACARP, 2004.
4. P. Corke, P. Sikka, J. Roberts, and E. Duff. DDX: A distributed software architecture for robotic systems. In *Australasian Conference on Robotics & Automation*, Canberra, December 2004.
5. A. Hemami. Modelling, analysis and preliminary studies for automatic scooping. *Advanced Robotics*, 8(5):511–529, 1994.
6. A. Hemami and S. Goulet. Resistance of particulate media to excavation: Application to bucket loading. *Int. J. Surface Mining, Reclamation and Environment*, 8:125–129, 1994.
7. J. Roberts, G. Winstanley, and P. Corke. Three-dimensional imaging for a very large excavator. *International Journal of Robotics Research*, 22(7-8):467–477, 2003.
8. P.S. Rowe. Learning system and method for optimizing control of autonomous earthmoving machinery. US Patent 6,076,030, June 2000.
9. X. Shi, F.-Y. Wang, and P.J.A. Lever. Experimental results of robotic excavation using fuzzy behavior control. *Control Eng. Practice*, 4(2):145–152, 1996.
10. S. Singh. Learning to predict resistive forces during robotic excavation. In *International Conference on Robotics & Automation*, pages 2102–2107, Nagoya, May 1995.
11. S. Singh. The state of the art in autonomous earthmoving, 2002. In *Workshop on Advanced Geomechatronics*, Sendai, Japan, 2002.
12. S. Singh and H. Cannon. Method and apparatus for determining an excavation strategy. US Patent 6,108,949, August 2000.
13. Carnegie-Mellon University. System for autonomous excavation and truck loading. UK Patent GB 2,342,640, April 2000.

Automated Inspection System for Large Underground Concrete Pipes Under Operating Conditions

Norbert Elkmann¹, Bert Reimann¹, Erik Schulenburg¹, Heiko Althoff² et al.

¹ Fraunhofer Institute IFF, Magdeburg, Germany

² Emschergenossenschaft, Essen, Germany

Summary. The Emschergenossenschaft based in Germany is currently planning the Emscher sewer system, arguably the largest residential water management project in Europe in years to come. In 2002, the Emschergenossenschaft engaged the Fraunhofer Institute for Factory Operation and Automation (IFF) in Magdeburg, Germany as the general contractor to develop automatic inspection and cleaning systems to meet the requirements stipulated by the legal guidelines. The systems must operate continuously in a sewer line, which has diameters ranging from 1400 to 2800 mm and is partially filled, 25% at minimum, all the time. To construct the Emscher sewer system, the Emschergenossenschaft favors a one-pipe line in long sections. A walk-through or inspection by personnel is impossible in every phase. The Fraunhofer Institute IFF has completed an extensive concept study for the inspection and cleaning systems and has developed as prototypes all systems for motion along the sewer and all sensor systems, achieving a new quality of inspection under these difficult conditions. This article describes significant project results and important components of the inspection and cleaning systems such as the inspection systems, pipe axis measurement, system positioning and sensor systems for damage detection.

Fundamental for the development of the inspection systems are a detailed inspection going far beyond the video inspection common today and the capability of taking comparative measurements throughout the sewer system's period of operation to describe the development of damage.

Keywords: Inspection System, Positioning, Sensor Systems, Sewer

1 Introduction, Objectives and Approach

The Emscher sewer system has a length of approximately 51 km with pipe diameters of DN 1400 to DN 2800 and a depth of 5 to 40 m. The maximum distance between manholes is 600 m. The sewer material is reinforced concrete. The sewer line is uncoated and unlined. Even in dry weather, large quantities

of water will be discharged into the sewer constantly. Legal guidelines require a sewer system's structural and operational condition be inspected and recorded on a scheduled, regular and systematic basis. At present, structural condition is usually detected by optical inspection (TV inspection or walk-through inspection). Conventional inspection methods cannot be used to inspect the Emscher sewer system because of its constant partial filling. The automatic inspection and cleaning systems to be designed as part of the project should effectively do away with walk-through sewer inspections. Hence, among other things, approval of the one-pipe sewer will depend on demonstrating that inspection and cleaning can be done by using remote controlled or automatic systems.

As part of the project, the following main components were designed and tested for their feasibility and fulfillment of the requirements:

- Carrier system (motion kinematics, robot) for positioning along the sewer line;
- Sensor and measuring systems for inspecting pipe condition above and below the water line as well as for detecting deposits;
- Sewer cleaning system;
- Media supply (power, data communication, water, etc.);
- Control system, operation;
- System navigation and positioning in the sewer;
- Handling systems for positioning sensors and cleaning tools on and along the sewer wall.

A large test station with various reinforced concrete pipes and different types of damage (e.g. cracks or spalling) was set up at the Fraunhofer Institute IFF in Magdeburg. The sensors for inspection were mostly new developments.

In consultation with the EmscherGenossenschaft, the Fraunhofer IFF has developed and built a prototype of the rough inspection system as well as a test prototype of the cleaning system and a test prototype of the inspection system in order to test these in a comparable, already existing sewer system (diameter 2300 mm).

2 Inspection Strategy and Systems

The strategy for automatically inspecting and cleaning the Emscher sewer system incorporates a three-stage approach.

In the first stage, a small swimming system called the Spy is employed in the sewer for rough inspection. It inspects and measures the entire sewer line and conducts camera inspections, recording major abnormalities such as erosion, deposits, obstacles and leaks in the gas space. At the same time, it checks whether the cleaning and inspection systems detailed below can be deployed. The Spy must be able to position itself centrally even in curved pipes in the sewer covering a length of 600 m.

In the second stage, if necessary, the wheel-driven cleaning system eliminates deposits detected by the spy in the bed area and cleans the sewer wall before the inspection system is deployed.

In the third stage, the inspection system (wheel-driven for diameters < 2000 mm, swimming for diameters > 2000 mm) inspects the sewer completely, measuring the sewer (joint widths, pipe offsets, cracks) with greater accuracy than the Spy. For all three systems, concepts were designed and de-



Fig. 1. Spy prototype and test of the swimming inspection system in a real sewer

veloped for their control, operation, introduction into and extraction from the sewer line as well as for their energy and water supply and data transfer.

2.1 Rough Inspection System (Spy)

The Spy (Fig. 2) is an easy-to-operate, cable-guided swimming system for rough screening of the sewer line. The Spy detects corrosion, obstacles, deposits and cracks. The Spy detects sewer conditions with little operating effort but with less precision than the inspection system.

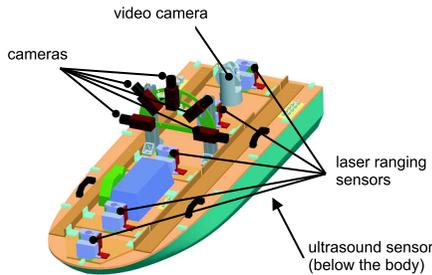


Fig. 2. Sensors on Spy

Using a camera system, the Spy can visually inspect the gas space. It is equipped with several flashlights for illumination. Flashlights are used because they yield more light while consuming less power than floodlights. The Spy is additionally equipped with ultrasound sensors for sewer measurement in the water space. The Spy prototype must have smooth swimming behavior and lie stably in the sewer line even at higher flow velocity. The objective of the successful navigation tests was to position the Spy centrally in the current in order to create good conditions for geometry measurement. It was possible to determine the position of the Spy in the sewer and measure the sewer cross section in the gas space.

2.2 Inspection Systems

In contrast to the Spy, the distinctive feature of the inspection system is its ability to achieve greater accuracy of measurement with its measuring sensors. Sensor systems are additionally integrated. Various concepts were developed for the carrier system. The two favored carrier systems are:

1. Floating systems for large sewer diameters and
2. Wheeled chassis for smaller sewer diameters

The floating systems are convincing because of the high certainty of recovery. Their operational range is limited by the required water level though. Wheel-guided car systems are used when filling level is low or nominal diameters are smaller. The test prototype is modularly constructed and represents the

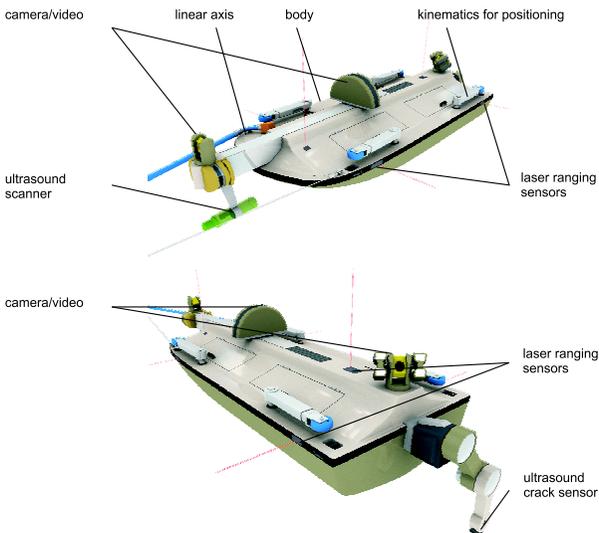


Fig. 3. Swimming inspection system

two favored carrier system concepts: Swimmer and Car. The Car test prototype consists of the Swimmer and the additional wheeled chassis subsystem. Sensors for determining position in the sewer (laser ranging sensors and inclination sensors) and sensors for damage detection (laser scanners, ultrasound scanners, camera system, ultrasound crack sensor) were installed on the inspection system. These sensors are either rigidly connected directly with the carrier system or they are moved by additional sensor kinematics. The rotation arm on the stern of the carrier system moves the ultrasound crack sensor along the sewer wall. Ultrasound scanners, laser scanners and camera system are mounted on a linear axis and can precisely measure the pipe profile over a length of approximately 1.5 m.

3 Positioning, Pipe Axis Measurement

Momentary position and orientation in the sewer have to be known at the time of any measurements with the Spy and with the inspection system. Therefore, pipe axis measurement is an essential prerequisite for exactly representing and analyzing the sensor data. To this end, an algorithm was developed, which, taking a model of a complete pipe as its starting point, measures the pipe axis exactly. The total error of pipe axis measurement is arrived at by adding up the accuracy of the laser ranging sensors, the tolerance and the pipe's surface condition as well as the systematic error caused by the Spy system's motion. For the laser and ultrasound scanners to detect damage, a positional value of the pipe axis has to be assigned for every individual reading. Accordingly, when the accuracy of measurement is being assessed, the superposition of position detection and the measuring method for detecting damage has to be assessed.

The inspection system achieves a greater accuracy of measurement of its position in the pipe because it is stationary during measurement and, as such, only the accuracy of the laser ranging sensors themselves plays a role in the overall measurement accuracy. A sensor system was conceived, which uses 15 laser ranging sensors (5 aligned vertically and 10 horizontally) to constantly record position. The sensor distance data is converted into the sensor coordinate system.

To measure the pipe axis, a cylinder with an elliptical surface area is used to model the real pipe with its surface quality and tolerances. This is clearly described in the Spy's coordinate system by the cylinder axis, the radius and the diameter. The interpretation of the measuring data would be easy if the Spy were exactly in the center of the pipe without any deviation in its angles of alignment. In reality, the systems tilt at yaw and pitch angles are out of line with the center of the pipe. The measuring points are not on a straight line but rather on a segment as the green measuring points in Fig. 4 indicate. Hence an exact model of the measurement has to be made, which allows for the curvature of the pipe. The model is based on the correlation between

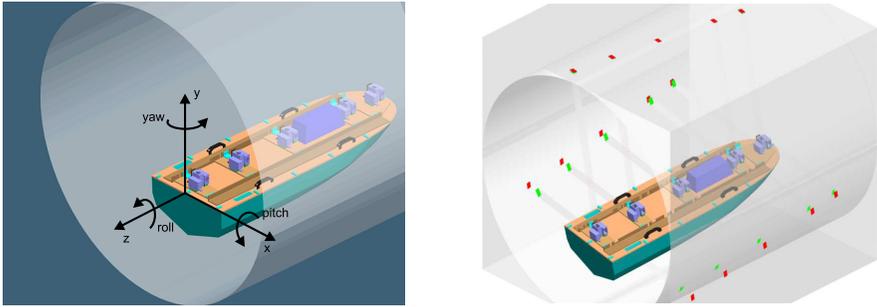


Fig. 4. Coordinate system, difference between the readings in the pipe model (green measuring points) and the model with straight walls (red measuring points).

the measured distance, pipe radius and displacement to the pipe axis as well as the yaw and pitch angles. Since this non-linear dependence is known, the alignment of the pipe axis can be determined from the distance measurement. This alignment then makes it possible to transform the measuring points onto the circular projection of the pipe and consequently to determine the position of the pipe axis in relation to the Spy and the inspection system. The pipe position is determined first by mathematically resolving the non-linear correlations. The determination of the position of the pipe axis is based on using the pipe axis alignment to plot the measuring points on the circular projection. After applying this transformation, a circle with an offset center is fit to the measuring points. The displacement of the axis of the pipe vis-à-vis the axis of the Spy is obtained from this fit.

In principle, this approach opens a method for measuring the pipe axis, which is independent of the pipe diameter as well as of the orientation and position of the measuring system. The method was modified to the effect that the sensor alignment is compensated for by parable approximations. Corresponding calibrating measurements are taken on a calibration rig.

The system's position along the axis of the sewer line is determined by measuring the lengths of cable uncoiled. In addition, the camera system references the current position at all joints with the camera system. This way, inaccuracies caused by cable sag and slippage can be compensated for and every single pipe can be approached with an accuracy of ± 50 mm.

4 Types of Damage and Selected Sensor Systems for Damage Detection

One focus of the project was the development of the sensor systems, which have the required accuracy of measurement under difficult conditions in the sewer and make it possible to take comparative measurements throughout the sewer's period of operation (120 years).

Minimum requirements for sewer inspection in Germany are stipulated in self-monitoring regulations issued by the states. Legal requirements, technical specifications and negotiations with local authorities have produced the inspection tasks displayed in Table 1. The requirements of a one-pipe line are far more demanding than those in the technical guidelines.

Table 1. Inspection tasks for the interceptors parallel to the Emscher sewer system

| Damage | Damage Specification | Accuracy | Detection Necessary | |
|------------------------------------|--|--|---------------------|-------------|
| | | | gas space | water space |
| chemical corrosion (gas space) | corrosion in tube walls and tube connections | loss of material / differentiation between - smooth pipe wall - rough pipe wall - gravel visible, - gravel eroded - reinforcement visible (comparison with previous inspections) | yes | No |
| mechanical corrosion (water space) | erosion | loss of material (10 mm) (comparison with previous inspections) | no | yes |
| inhibition of flow | obstacles, sediments, incrustations | 5% of sectional area | yes | yes |
| Deviation | horizontal/vertical | ± 8 mm | yes | no |
| | axial | ± 5 mm | yes | no |
| Cracks | longitudinal/ radial | $> 0,5$ mm | yes | no * |

(*) Cracks caused by mechanical stress can be located throughout the entire pipe. Cracks detected in the upper section of the pipe can be used to calculate the extent of cracks in the lower section.

4.1 Chemical Corrosion

Optical measuring methods detect surface corrosion of the concrete in the gas space and represent possible developments of damage. A semiautomatic procedure consisting of automatic and manual analysis by an operator is favored for corrosion detection and classification. The option of mapping the concrete wall comparatively with previous inspections is important in order to be able to map any possible development of damage. Several cameras are used to image the sewer wall completely.

An image processing algorithm with a short runtime is used to detect abnormalities immediately. The appearance of individual structural elements of the surface is inspected for abnormalities, the measured number being more important than its precise characteristic. When a variable limit value is exceeded, surface corrosion may be likely.

Direct statements can be made about the possible occurrence of corrosion by comparing the distribution of the various proportions of gray tones in the readings with calibrated values or values already ascertained from previous inspections. Fig. 5 presents characteristic gray tone distribution curves for different surfaces. Fig. 5 clearly shows the various curves for differently corroded

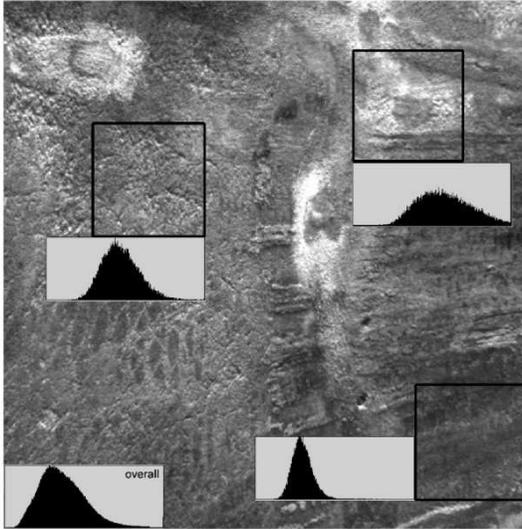


Fig. 5. Image of a corroded concrete surface with superimposed gray scale curve for subareas.

surfaces. Clearly, differently damaged subareas can be identified individually. The total assessment of potential corrosion would be obtained by averaging the entire image space.

In addition, laser scanners, which measure the cross section of the pipe, are used to detect corrosion with an accuracy of ± 4 mm.

4.2 Obstacles, Sediments, Incrustations, Mechanical Corrosion

Newly developed ultrasound scanners with an accuracy of measurement of ± 2 mm are being used to detect obstacles, deposits and mechanical erosion in the water space.

Fig. 6 shows the test setup for geometry measurement in the water space with ultrasound scanners and a scan image.

4.3 Crack Detection in Concrete Pipe

First, digital image processing systems are used to detect cracks in the gas space. Several cameras are used to identify cracks in the gas space.

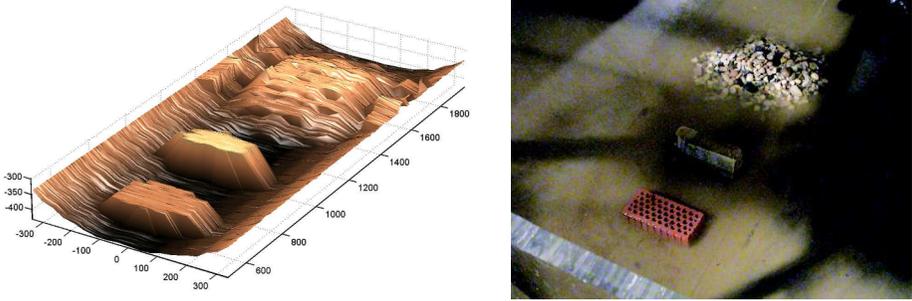


Fig. 6. Geometry measurements with ultrasound scanner (obstacles, sediments)

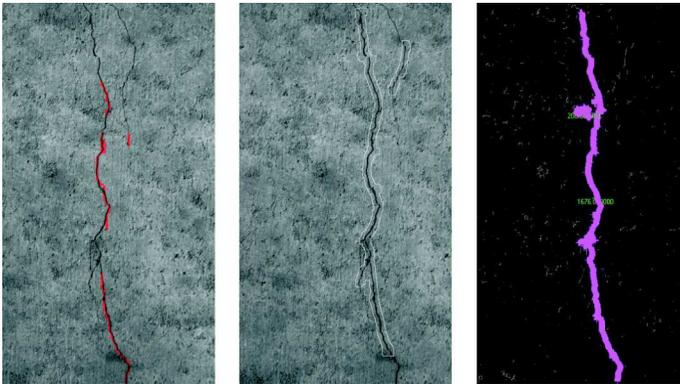


Fig. 7. Details of result images when different crack detection methods are employed

In accordance with the requirements, cracks with a width of 0.5 mm and upward have been positively identified and logged. While cracks can reach a long length, their frequently very narrow width makes great demands on the measuring system mapping them. Other measures such as comparisons with previous inspections and images of other cracks with known width as well as the superimposition of scales help make it possible to more closely determine crack width and thus more closely detect the type of damage. An important analysis module is automatic crack detection. It employs methods of image processing and pattern recognition in order to determine whether one or more cracks are possibly visible on a particular image or not. Particularly when there are small cracks, which an operator could overlook on the monitor, this automatic system constitutes a considerable advantage and increases the quality of the inspection results. Fig. 7 illustrates how different analysis modules identified a crack. In addition, each crack was graphically marked as a recognized structure for the purpose of presentation. The entire crack configuration was never identified. However, only the information of whether a

crack may be present in a particular image or not is important for supporting the user. It follows from this that the automatic analysis module can already terminate the processing of the current image and inform the user once any crack segment has been found.

Additionally, new acoustic methods (ultrasound, impact-echo) have been developed or adopted to detect cracks in the concrete in the gas and the water space. These acoustic systems are able to provide information on crack depth. The acoustic methods for crack detection additionally allow the following:

- With the right sensor system, cracks can be detected in the water space too.
- Cracks can be roughly classified (crack depth).
- Spallings can be detected and wall thickness can be determined.



Fig. 8. Acoustic sensor systems for crack detection

The use of these sensors sensor systems for crack inspection is completely new.

4.4 Deviation of Pipe Position

Horizontal and vertical deviations of position and joint gaps have to be measured. Laser scanners, aligned laterally or on the apex of the sewer, are used to detect and record the horizontal and vertical deviations of position.

In the gas space, cameras measure the joint gap. Differences in joint width compared with earlier inspections indicate an axial displacement. Inconstant joint width along the pipe circumference indicates a deformation.

Automatic measurement requires exact identification of the joint edges. To this end, image processing methods (segmentation, contour-finding) determine the pixels on the edges of the joint.

Fig. 9 (a) shows a detail of the identified pixels. When the parameters have been suitably selected, the joint edges can be identified with an accuracy of a few pixels. If these pixels are used to apply ellipse approximations, which optimally approximate the number of points, the joint edges are obtained, which support automatic measurement of the joints.

Fig. 9 (b) shows a detail of the joint image with such ellipse approximations.

A Hough transformation can be used to determine the ellipse approximations. Since positioning and joint identification already identify the joint edges, the parameters of the corresponding ellipses are also approximately known. Thus, the search area of the Hough transformation can be restricted greatly, making efficient implementation possible.

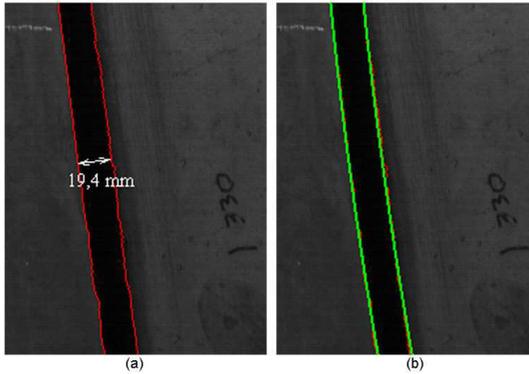


Fig. 9. Image detail with detected joint edges: When measurement is manual the joint width can be marked by hand (a). When joint measurement is automated, ellipse approximations are placed through the joint edges (b).

5 Summary and Outlook

Since 2002, the Fraunhofer IFF as general contractor has developed a comprehensive concept for inspection and cleaning systems for the Emscher sewer system. Not only have all the relevant subsystems been identified but they have also been designed in detail and subjected to all necessary tests in order to be able to provide reliable information about their feasibility. Feasibility was fully demonstrated. Foci of development were the carrier systems for movement along the sewer line guaranteeing maximum recovery certainty, the pipe axis measurement and position sensing of the systems in the sewer and the sensor systems for detecting the condition of the sewer's gas and water spaces. Different sensor systems have been developed and tested in the test station as well as in a real sewer. Erosion, incrustations and corrosion of concrete are detected with great accuracy. Cameras detect axial displacement and laser scanners detect offsets in pipe joints in the gas space. Apart from the cameras, different sensors for crack detection in the gas and water space were developed on an acoustic basis (e.g. ultrasound).

Along with the sensors, all systems were designed for the favored inspection and cleaning concept. This involved a system for rough inspection of the sewer (Spy) as well as cleaning systems and inspection systems. The control, the operation, the introduction into and extraction from the sewer and the manhole as well as the energy and water supply were engineered and the certainty of recovery in case of breakdown was guaranteed.

In consultation with the Emschergerossenschaft, the Fraunhofer IFF developed and built a prototype of the spy and test prototypes of the cleaning system and the inspection system in order to acquire more experience under real conditions in the sewer. The swimming behavior of the Spy and the floating test prototype for the inspection system were studied. The sensor behavior for crack detection and sewer cross section measurement was also tested. The collected findings and insights will now enter into the engineering and development for final prototypes.

The feasibility of automatic inspection and cleaning systems for the Emscher sewer system and the fulfillment of the legal requirements for inspection and cleaning have been demonstrated. The research on and tests of the inspection systems, the sensor systems and the cleaning technology guarantee the inspection and cleaning required by law in a one-pipe sewer.

References

1. Hertzberg, Christaller, Kirchner, Licht, Rome: "Sewer Robotics", In: Proc. From Animals to Animats 5, 5th Intl. Conf. On Simulation of Adaptive Behavior (SAB-98), R. Pfeifer and B. Blumberg and J.-A. Meyer and S.W. Wilson (eds), MIT Press, P. 427-436, 1998
2. Kuntze H.-B., Haffner H.: Experiences with the Development of a Robot for Smart Multisensoric Pipe Inspection. ICRA 1998: 1773-1778
3. Rome E., Hertzberg J., Kirchner F., Licht U., Streich S., Christaller Th.: Towards Autonomous Sewer Robots: the MAKRO Project Urban Water 1, 1999, P. 57-40
4. Kirkham R., Kearney, P. Rogers K. and Mashford J.: PIRAT - A System for Quantitative Sewer Pipe Assessment. International Journal of Robotics Research, Vol. 19, No. 11, November 2000
5. Elkmann N., Althoff H., Saenz J., Böhme T.: Kinematics Systems for Inspection and Cleaning of Sewer Canals. 6th International Conference on Climbing and Walking Robots CLAWAR, Catania, 2003
6. Elkmann N., Althoff H., Böhme T., Felsch T., Kutzner S., Saenz J., Stürze T.: Entwicklung von Robotersystemen für die Inspektion und Reinigung von Abwasserkanälen, Robotik 2004, München, 17-18 June 2004
7. Elkmann N., Althoff H.: The emscher:kanal - Development of an Automated Inspection System for Underground Concrete Pipes, 22th International NO DIG Conference, 15.-17. November 2004, Hamburg, Germany

An Autonomous Weeding Robot for Organic Farming

Tijmen Bakker¹, Kees van Asselt¹, Jan Bontsema², Joachim Müller³ and Gerrit van Straten¹

¹ Wageningen University, Systems and Control Group, P.O. Box 17, 6700 AA Wageningen, The Netherlands, tijmen.bakker@wur.nl

² Agrotechnology and Food Innovations BV, P.O. Box 17, 6700 AA Wageningen, The Netherlands

³ University of Hohenheim, Institute for Agricultural Engineering, 70593 Stuttgart, Germany

Summary. The objective of this research is the replacement of hand weeding in organic farming by a device working autonomously at field level. The autonomous weeding robot was designed using a structured design approach, giving a good overview of the total design. A vehicle was developed with a diesel engine, hydraulic transmission, four-wheel drive and four-wheel steering. The available power and the stability of the vehicle does not limit the freedom of research regarding solutions for intra-row weed detection and weeding actuators. To fulfill the function of navigation along the row a new machine vision algorithm was developed. A test in sugar beet in a greenhouse showed that the algorithm was able to find the crop row with an average error of less than 25 mm. The vehicle is a versatile design for an autonomous weeding robot in a research context. The result of the design has good potential for autonomous weeding in the near future.

Keywords: Systematic design, machine vision, GPS, robotics, intra-row weed control, autonomous weeding robot, organic farming

1 Introduction

Weeds in agricultural production are mainly controlled by herbicides. As in organic farming no herbicides can be used, weed control is a major problem. While there is sufficient equipment available to control the weeds in between the rows, weed control in the rows (intra-row weeding) still requires a lot of manual labour. This is especially the case for crops that are slowly growing and shallowly sown like sugar beet, carrots and onions. In 1998, on average 73 hours per hectare sugar beet were spent on hand weeding in the Netherlands [4]. The required labour for hand weeding is expensive and often not

available. An autonomous weeding robot replacing this labour, could mean an enormous stimulus for organic farming. This paper presents the design of such an autonomous weeding robot currently being developed at Wageningen University.

2 The Design Procedure

2.1 Method

The autonomous weeding robot is designed using a phase model as the design method[3]. In this phase model the design of a product is represented as a process consisting of a problem definition phase, alternatives definition phase and a forming phase (figure 1). The results of the different phases are solutions on different levels of abstraction.

The problem definition phase starts with defining the objective of the design. In the problem definition phase also the set of requirements is established. The requirements can be split into fixed and variable requirements. A design that does not satisfy the fixed requirements is rejected. Variable requirements have to be fulfilled to a certain extent. To what extent these requirements are fulfilled, determines the quality of the design. The variable requirements are also used as criteria for the evaluation of possible concept solutions. The last part of the problem definition phase consists of the definition of the functions of the robot. A function is an action that has to be performed by the robot to reach a specific goal. In our case, important functions are 'intra-row weeding' and 'navigate along the row'. The functions are grouped in a function structure, which represents a solution on the first level of abstraction.

The function structure consists of several functions. Every function can be accomplished by several alternative principles, e.g. mechanical and thermal principles for weed removal. In the alternatives definition phase, possible alternative principles for the various functions are presented in a morphological chart (fig. 3). The left column lists the functions and the rows display the

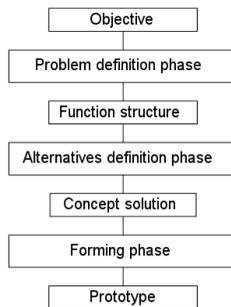


Fig. 1. The design process

alternative principles. By selecting one alternative for each function and by combining these alternatives, concept solutions can be established. These concept solutions are represented by lines drawn in the morphological chart. The best concept solution can be selected using a rating procedure. In the forming phase this selected concept solution is worked out into a prototype.

2.2 Application for the Weeding Robot

The objective of the research is formulated as 'replacement of hand weeding in organic farming by a device working autonomously at field level'. Starting from this objective, the first step in the problem definition phase was to establish the set of requirements. For this purpose interviews were held with potential users, scientists and consultants related to organic farming. The resulting requirements are as follows:

Fixed requirements:

- Replacing hand weeding in organic farming.
- Applicable in combination with other weed control measures.
- Manual control of the vehicle must be possible for moving the vehicle over short distances.
- Weeding a field autonomously.
- Ability to work both day and night.
- The weeding robot should not cross the field boundaries.
- The weeding robot must be self restarting in absence of emergency.

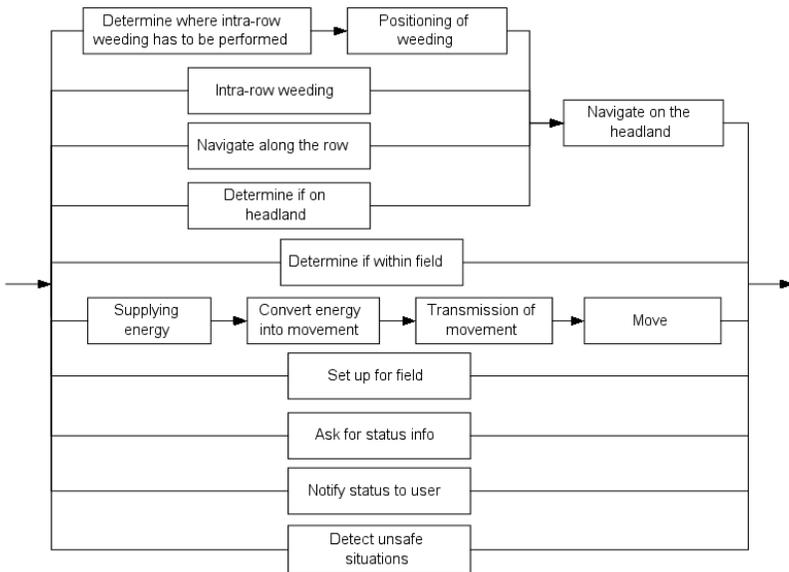


Fig. 2. The function structure

- The weeding robot informs the farmer when the weeding robot stopped definitely (e.g. due to security reasons) or when it is ready.
- The weeding robot sends its operational status to the user at request
- The weeding robot must function properly in sugar beet.

Variable requirements:

- Removing more than 90 percent of the weeds in the row.
- The costs per hectare may be at least comparable to the costs of hand weeding.
- Damage to the crop is as low as possible.
- The soil pressure under the weeding robot must be comparable or less than for hand weeding.
- Energy efficient.
- Safe for people, animals and property.
- Suitable as research platform.
- Limited noise production
- Reliable functioning.
- Easy to use.

After establishing the set of requirements the functions of the the weeding robot were identified. These functions were grouped into a function block scheme. This scheme is represented in figure 2. The lines in the scheme indicate flows of energy, material or information. Functions located in parallel lines can be performed simultaneously.

The navigation system consists of four functions. Firstly, the weeding robot should constantly determine if it is located in- or outside the field. Secondly, if within the field, it should determine if it is on one of the headlands or not. Thirdly, in case it is not on the headlands, it should navigate along the row and perform the intra-row weeding. Fourthly, if the weeding robot arrives on the headland, it should stop the intra-row weeding and start to navigate to the next crop rows to be weeded. This sequence repeats until the whole field, except the headlands, is weeded. Weeding of the headlands is left out of consideration. An increasing number of farmers in the Netherlands do not grow sugar beet at the headlands because they think it is not cost-effective.

In the alternatives definition phase possible alternative principles for the various functions are listed in a morphological chart (fig. 3). Four people involved in the project drew lines indicating possible concept solutions in the chart. These concept solutions were then weighed against each other using the variable requirements listed before. The concept solution indicated by the line in figure 3 is the final concept solution.

In the forming phase described in section 3 the concept solution was worked out into a prototype.

| | | | | | | | | | | | |
|---|--|--|--|--|--|--|--|--|--|--|--|
| Determine where intra-row weeding has to be performed | | | | | | | | | | | |
| Positioning of weeding | | | | | | | | | | | |
| Intra-row weeding | | | | | | | | | | | |
| Determine if within field | | | | | | | | | | | |
| Navigate along the row | | | | | | | | | | | |
| Determine if on headland | | | | | | | | | | | |
| Navigate on the headland | | | | | | | | | | | |
| Supplying energy | | | | | | | | | | | |
| Convert energy into movement | | | | | | | | | | | |
| Transmission of movement | | | | | | | | | | | |
| Move | | | | | | | | | | | |
| Set up for field | | | | | | | | | | | |
| Ask for status info | | | | | | | | | | | |
| Notify status to user | | | | | | | | | | | |
| Detect unsafe situations | | | | | | | | | | | |

Fig. 3. Morphologic chart

2.3 Results of the Design Process

Determine where intra-row weeding has to be performed

To determine where intra-row weeding has to be performed, pattern recognition of plant locations is going to be used. From earlier research [2] it is expected that the quality of detection of this method is at least as good as the quality of detection of other methods. Though combinations of methods like recognition of pattern, shape and colour are expected to have a potential for higher quality of detection, just pattern recognition is chosen because it is expected to be sufficient.

Positioning of weeding

To position the actuator at the location indicated by the detection system dead reckoning is going to be used. A wheel with encoder, giving a precise distance measurement, will be available already because it is also needed for the pattern recognition system.

Intra-row weeding

Intra-row weeding will be performed by a mechanical actuator. It is expected to be difficult to remove weeds growing close to a crop plant by air, flaming, electricity, hot water, freezing, microwaves or infrared without damaging the crop plants. In that respect laser would be an excellent solution. However, laser can not work under the ground surface, and has therefore less effect on certain weed species. On the other hand, not moving the soil prevents buried seeds from germinating. A greater disadvantage of laser is its high price. High power laser is needed to reach reasonable performance, and this involves high costs. Water-jet could also probably be a good solution for intra-row weeding, but this needs much more investigation than a mechanical solution.

Determine if within field

GPS is selected to determine whether the weeding robot is within the field or not. The determination if the weeding robot is located within the field or not, needs to be guaranteed correctly at any time. A combination of vision and dead reckoning can not give this guarantee as good as a solution in which GPS is used. Dead reckoning could improve the position determination by GPS. However, if a GPS is selected with sufficient accuracy, additional dead reckoning is not needed.

Navigate along the row

Machine vision is selected for navigation along the row. Machine vision makes it possible to navigate along the row by relative positioning to the row. Therefore the weeding robot can work in any field without requiring absolute coordinates of a path to be followed. Absolute positioning by means of GPS,

possibly combined with other sensors, requires knowledge of the absolute position of crop rows in a field. Navigation along the row by relative positioning to the row could be done also using tactile, ultrasonic or optical sensors combined with dead reckoning. Tactile sensors are not going to be used because in case of sugar beet they could harm the crop. Machine vision is preferred over ultrasonic or optical sensors, because of the ability to look forward, which contributes to a more accurate control of the position of the weeding robot relative to the crop row. It is not clear whether dead reckoning could substantially contribute to the navigation accuracy feasible with machine vision.

Determine if on headland

GPS is selected to determine if the weeding robot is located on the headland. Using GPS requires some labour for recording the border of the headlands in advance, but will result in a correct headland detection. If a high accuracy GPS is selected, accuracy does not have to be improved by dead reckoning. Tactile, ultrasonic or optical sensors in combination with dead reckoning could also be used to determine whether the robot is on the headland, by detecting the end of the row, i.e. if over some predefined distance no row is detected. However, another crop may grow on the headland (seeded to prevent germinating of weeds) or crop rows seeded at the headland can cross the crop row to be followed. In these situations the latter solutions can not guarantee a correct detection of the end of row, and therefore also not a correct headland detection. Machine vision could give more reliable results, but it is still difficult because headland to be detected is not so structured.

Navigate on headland

For navigation on the headland GPS is selected. On the headland the weeding robot has to make a turn and position itself in front of the next rows to be weeded. At the moment the robot arrives at the headland, a virtual path is planned to a position in front of the next rows to be weeded. Navigating over this path is going to be done by GPS.

Locomotion related functions

A diesel engine with a hydraulic transmission was selected for the locomotion related functions. For weeding quite some power could be required and the available power should not be limiting for realizing the objective of autonomous weeding of a field. A diesel engine with an hydraulic transmission is a proven concept in agriculture. A gearbox limits the possible combinations of the number of engine revolutions and driving speed and shuffling is difficult to automate. A continuously variable or hydraulic transmission is therefore preferred over a gearbox. Hydraulics makes it possible to design a compact wheel construction preventing damage to the crop.

A design with four wheels is preferred over one with three because of stability.

It was decided that four wheels is also preferred over two or four tracks. The most important advantages of tracks in practice are the better traction and the less soil compaction. But it is expected that if four wheels are used for such a light-weight vehicle (not more than 1500 kg) soil compaction will be acceptable. Traction when using wheels is expected to be good enough because of the limited weight and the limited need of traction for intra-row weeding. Four wheel drive and four wheel steering were chosen to have the possibility to investigate all kinds of driving strategies.

Communication with the user

Specific settings for a field will be defined by a board computer. Any moment a user wants to know the status of the weeding robot, the weeding robot status will be accessible via the internet. A website gives good opportunities to represent information in an orderly way and it is easily accessible from everywhere. In case the weeding robot needs help from its user, the weeding robot notifies its user by sending an SMS (Short Message Service) message by the GSM network. In the Netherlands any place is covered by the GSM network. From the alternatives listed, SMS is the solution that gives the highest assurance that the user really receives the message shortly after it is sent.

Detect unsafe situations

Detecting unsafe situations will be done super canopy all around the weeding robot. Situations in which this solution is not sufficient are hardly imaginable. Ideally the weeding robot should detect every unsafe situation, at every level and direction. Even if somebody is lying in between the crop rows below canopy level this should be detected. Because of the research effort involved in reaching the ideal objective mentioned and the possible high costs for such a solution, detecting around and only super canopy is preferred.

3 The Vehicle

The size of the vehicle was determined by the standard track width used in agriculture in the Netherlands which is 1.50 m. This track width also makes the design versatile in the sense that it is suitable for crops grown in beds like carrots and onions. See figure 4 for the resulting vehicle.

Sugar beets are grown at a row distance of 50 cm so the weeding robot covers three rows. The engine power is selected so that it has enough power for driving and steering under field conditions and for driving three actuators. The required power for the actuators was calculated based on an actuator specially designed for intra-row weeding by Bontsema *et al.* [2]. The engine is a 31.3 kW Kubota V1505-T.

The ground clearance is about 50 cm to prevent the crop from being damaged by the vehicle. The vehicle is 2.5 m. long to have enough space for mounting



Fig. 4. The weeding robot

actuators under the vehicle in the middle between the front and rear wheels. The tyre width of 16 cm leaves enough space for steering in between crop rows while soil compaction is expected to be acceptable. The weight of the vehicle is about 1250 kg.

The engine drives two hydraulic pumps. One supplies the oil for steering and driving, and the other for driving the actuators. The oil for driving and steering flows to a electrically controllable valve block with eight sections. Four are used for steering and four are used for controlling wheel speed, so wheel speeds and wheel angles can be controlled individually. The wheels are driven by radial piston motors. The required driving speed range for intra-row weeding is 0.025 m/s - 2 m/s continually variable. A desired top speed of 5.6 m/s was specified for fast moving of the robot within a field. It appeared that hydraulics could not be designed to have a variable work speed from 0.025 m/s to 5.6 m/s. A solution was found by designing the hydraulics so that two speed ranges exist. The working speed ranges up to 3.2 m/s. A maximum travel speed of 6.4 m/s is realized by changing to two wheel drive by combining the oil flows of four wheels into two flows.

Each wheel is steered by an hydraulic motor with a reduction gear. The maximum steering speed is 360 degrees per second. The angles of the wheels are measured by angle sensors. The oil for driving the wheels flows via a turnable oil throughput. This makes it possible to turn the wheels in any angle from 0-360 degrees.

The weeding robot electronics consists of 6 units connected by a CAN bus

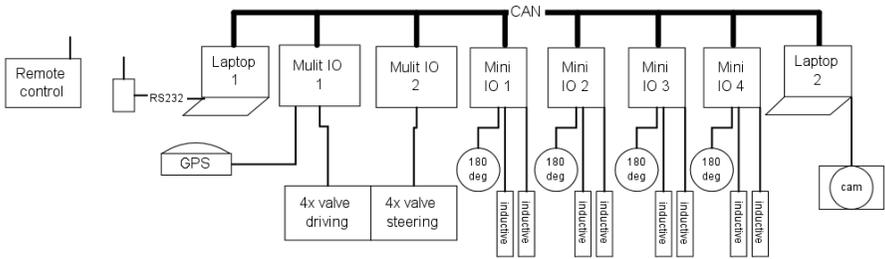


Fig. 5. Electronics architecture

using the ISO 11783 protocol. In figure 5 an schematic overview is given of this system with vehicle control related sensors and valves. Four micro controllers are located near the four wheels to measure the wheel speed and the wheel angle. Angles, wheel speeds and wheel direction are transmitted using the CAN bus. Via the CAN bus and two other micro controllers the hydraulic valve block is controlled. One laptop processes images supplied by the camera connected and returns the location of the crop rows in relation to the vehicle position in a CAN bus message. Another laptop does the vehicle control. It gathers wheel speed, wheel direction, crop row location data and GPS data and controls the vehicle by sending messages to the units connected to the valve block. This laptop is also the user interface of the weeding robot. A remote control is connected to this laptop via a radio modem for manual control of the weeding robot. Besides the sensors directly related to navigation and control, there are some more sensors connected to the modules. These sensors indicate oil filters functioning, oil temperature and oil level are also interfaced to the laptop. If a sensor indicates an emergency, the weeding robot will turn off automatically.

4 Navigation Along the Row

As explained in section 2.2 part of the navigation system of the weeding robot will consist of navigating along the row using machine vision. The machine vision algorithm was developed and tested on a sugar beet field prepared in a greenhouse. The area covered by one image was 2.5 meters long in row direction and 1.5 meters wide at the side closest to the camera. This means that three complete rows are visible in the image. The first step in the row recognition algorithm, is transforming the RGB image to a grey scale image with enhanced contrast between green plants and soil background. The next step is to correct the images for perspective by an inverse perspective transformation. In the corrected image three rectangular sections of crop row spacing are selected. The first section is selected in the middle of the image. The other two are selected on both sides of the first section. The sections are combined by summing up the grey values of the sections to a combined image. To the

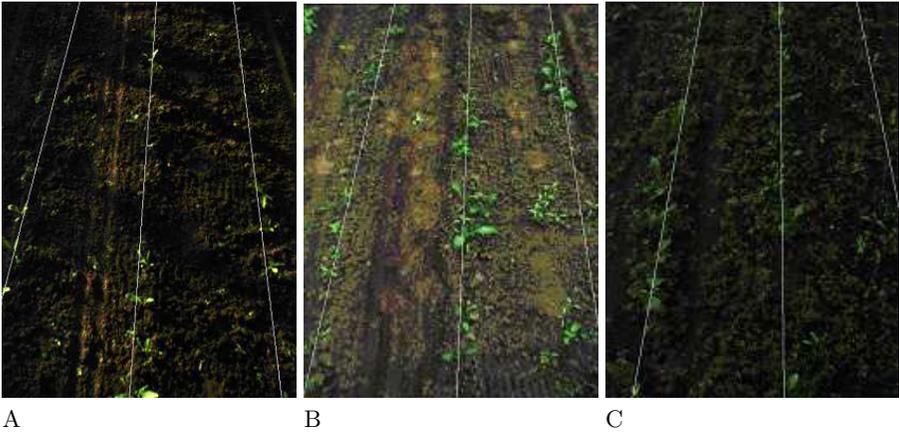


Fig. 6. Typical images together with estimated row position at different growth stage and weed pressure

resulting combined image grey scale Hough transform is applied.

Measurements show that the algorithm is able to find the row independent of the crop stage. Furthermore, the algorithm has found the crop row in images with a high weed density. Some typical results can be found in figure 6. The quality of the row position estimation is determined by comparing the lines found by the algorithm with lines positioned over the crop rows by hand in the original image. The average deviation between the estimated and real crop row varied from 6 to 223 mm. The higher deviations in this range can be explained by the number of plants visible in early crop stage, overexposure of the camera, and the presence of a lot of green algae due to our experimental setup. Ignoring the measurements under these extreme conditions, the algorithm was able to find the row with an average error of less than 25 mm. The measured processing time varied from 1 to 1.5 seconds per image. This variation can be explained by the varying amount of weed. The more light pixels there are, the more pixels have to be processed by the Hough transform. Details can be found in [1].

5 Conclusions

The advantage of using a structured design procedure is that it provides a good overview of the complete design. Also, the design method forces the designer to look at alternative solutions. Because of the structured sequence of design activities, it is easy to keep track of the progress of design. In a research context it is easy to identify alternative subjects that are worthwhile to investigate further. But in the mean time the main line of the research remains clear.

Applying the design procedure for the autonomous weeding robot resulted in a flexible research vehicle. The design consisting of diesel engine, hydraulic transmission, four wheel drive and 360 degrees four wheel steering is a good concept for an autonomous weeding robot in a research context. The available power and the stability of the vehicle does not limit the freedom regarding research to solutions for intra-row weed detection and weeding actuators.

From the established functions only for navigation along the row a new algorithm to detect sugar beet rows is discussed. The algorithm is able to find the row with an average error of less than 25 mm. Processing one image covering 2.5 meters row length takes less than 1.5 second. It is not expected that the attainable driving speed of about 1.5 m/s will be limiting. From earlier research it is expected that the actuator will limit the driving speed to 1 m/s or less. So it can be concluded that the results of the algorithm give good perspectives to navigate an autonomous vehicle along rows in a sugar beet field.

The planning for the current year is to finish the autonomous navigation and control of the weeding robot. This will be tested in a sugar beet field. Adding an intra-row weeding system is planned for next year. The ultimate test will then be to show that it is possible to weed a whole sugar beet field autonomously by a weeding robot.

References

- [1] T. Bakker, H. Wouters, C.J. van Asselt, J. Bontsema, J. Müller, G. van Straten, and L. Tang. A vision based row detection system for sugar beet. In *Computer-Bildanalyse in der Landwirtschaft. Workshop 2004*, Bornimer Agrartechnische Berichte, pages 42–55, Braunschweig, Germany, 2004. Institut für Agrartechnik Bornim e.V.
- [2] J. Bontsema, C.J. van Asselt, P.W.J. Lempens, and G. van Straten. Intra-row weed control: a mechatronics approach. In *1st IFAC Workshop on Control Applications and Ergonomics in Agriculture*, pages 93–97, Athens, Greece, 1998.
- [3] H.H. van den Kroonenberg and F.J. Siers. *Methodisch ontwerpen. Ontwerpmethoden, voorbeelden, cases, oefeningen*. Educatieve Partners Nederland BV, Houten, 1998.
- [4] R.Y. van der Weide, L.A.P. Lotz, P.O. Bleeker, and R.M.W. Groeneveld. Het spanningsveld tussen beheren en beheersen van onkruiden op biologische bedrijven. In F.G. Wijnands, J.J. Schroder, W. Sukkel, and R. Booij, editors, *Themaboek 303. Biologisch bedrijf onder de loep*, pages 129–138. Wageningen Universiteit, Wageningen, 2002.

V Shape Path Generation for Loading Operation by Wheel Loader

Shigeru Sarata¹, Yossewee Weeramhaeng², Akira Horiguchi³, and Takashi Tsubouchi²

¹ AIST, Namiki 1-2-1, Tsukuba, Ibaraki, JAPAN sarata-s@aist.go.jp

² University of Tsukuba, Ten-nohdai 1-1-1, Tsukuba, Ibaraki, JAPAN

³ Sogo Security Service Co., Saitama, JAPAN

Summary. In this paper, as a part of research work on the autonomous loading operation by wheel loader at surface mines or construction working places, a method of path generation for wheel loader will be described. V shape path connecting between the scooping position and the loading position consists of clothoid curves and straight lines. Each length of line segments are optimized in path generation procedure. The scooping direction is determined based on the estimation of resistance force applied on the bucket during scooping motion, by using simplified shape model of pile and bucket trajectory model. Proposed method is installed on the experimental model. Shape of the pile is measured by a stereo-vision system. For giving scooping position, scooping direction giving the least moment on the bucket is selected. By this method, appropriate path is generated.

Keywords: Path generation, wheel loader, loading operation, scooping

1 Introduction

One of the major fields for intelligent system applications is field of mining and construction. The working environment in mining or construction consists of irregular shaped material such as fragmented rock, sand or soil, and changes its shape with advancing the operation. Unmanned systems in these fields should be intelligent systems that can decide its actions based on the changing situation. Wheel loader (Front End Loader: "loader" hereinafter) is used for loading materials widely in these fields. It has large bucket at front end and four wheels. The main functions of the vehicle are to scoop with the bucket and freely maneuver with the wheels (Fig. 1). Our group has been researching an autonomous system for the loading operation of this vehicle [1, 2]. A method of path-generation for wheel loaders as part of our ongoing research is described in this paper. Several researches have been conducted on automatic operation system of wheel loader [3, 4]. These developed systems employed a guidance method or a teaching-playback method for traveling. Path generation

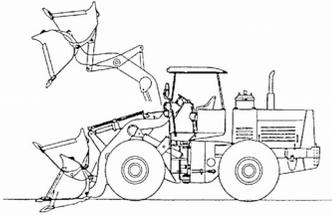


Fig. 1. Wheel Loader

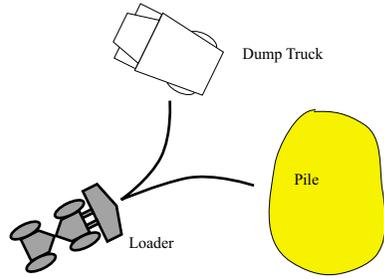


Fig. 2. V Shape Path

function was not included. As mentioned previously, path planning is one of the essential functions for autonomous operation system.

In the most common loading operation, loader travels on V shape path between scooping point and loading point on a dump truck including a switch back point as shown in Fig. 2. This paper describes a method to determine the scooping direction based on the shape of the pile and a method to generate V shape path. The proposed methods are applied in an experimental model and evaluated.

2 Path Planning

2.1 Steering System of Wheel Loader

Steering mechanism of wheel loader is articulate steering system as shown in Fig. 3. The body of the loader is separated into a front part and rear part connected by a centre pin. The angle around the centre pin is controlled by hydraulic cylinders. Because distances from centre pin to front axel and to rear axel are same, front wheels and rear wheels run on same trace and wheel loader has high mobility in off road such as muddy or soft soil.

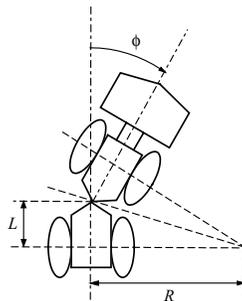


Fig. 3. Steering Mechanism

Let ϕ be the articulate angle, L be the distance from centre pin to the front axel or rear axel, R be the radius of curvature and k be the curvature. Relations between them are as follows:

$$L = R \tan(\phi/2) \tag{1}$$

$$k = 1/R = \frac{2 \tan(\phi/2)}{L} \tag{2}$$

When ϕ is small, curvature k is proportional to ϕ linearly. The operational range of the articulate angle is about 40 degrees. The relation between k and ϕ is almost linear in this range.

2.2 Path Planning with Path Elements

It is easy and effective to generate a path from the given initial position to the final position by combination of path elements such as straight line and curve. In the proposed method of path planning, clothoid is employed as curve element. Curvature of clothoid is proportional to the length of the path. Clothoid curve is described by following formulae.

$$x(s) = \int_{s_0}^s \cos(ks^2) ds \tag{3}$$

$$y(s) = \int_{s_0}^s \sin(ks^2) ds \tag{4}$$

Where s is the length of path and k is coefficient about sharpness of the path. If the curvature is changed as shown in Fig. 4(a), the path becomes the same clothoid joined in reverse direction and this curve is named "symmetrical clothoid" (Fig. 4(b)). At both of the end of symmetrical clothoid, curvature is zero therefore the curve can be connected to straight lines with continuously [5].

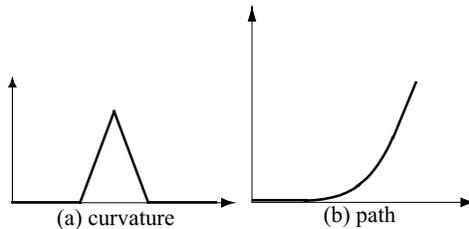


Fig. 4. Symmetrical Clothoid

V shape path as described in the previous section can be composed of two symmetrical clothoid and straight lines. For convenient of explanation, it is supposed that the initial position of loader is on the origin of coordinate system and direction of the loader is positive direction of x axis as shown in Fig. 5 . The grey zone in Fig. 5 is the area that can be reached via the combination of straight line l_1 , symmetrical clothoid c_1 , straight line l_2 , symmetrical clothoid c_2 , and straight line l_3 . The directions of straight lines l_1 , l_2 and l_3 are the directions of the loader at the initial position, at the end of c_1 , and at the final position, respectively. The loading operation with V shape path requires that the loader should move backwards on l_1 and c_1 , move forwards on c_2 and l_3 , therefore l_1 should be negative and l_3 should be positive value.

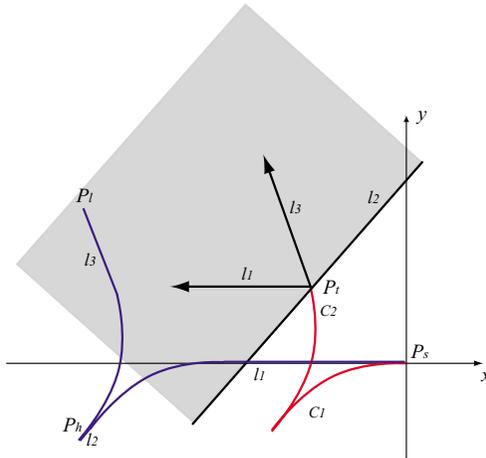


Fig. 5. Zone reachable via combination of clothoid and straight lines

When the end position and direction of the symmetrical clothoid is given, corresponding clothoid is obtained by solution of formulae (3) and (4). But they cannot be solved analytically because (3) and (4) are Fresnel integral. To simplify the procedure of numerical solution, the curvature of symmetrical clothoid is limited to several values. The path planning proceeds in the following steps. Let P_s be the scooping position, θ_s be direction at that position, P_l be the loading position, θ_l be direction at that position and θ_h be direction at P_h :the switch back point. θ_h takes value between θ_s and θ_l . The final path consists of $l_1 - c_1 - l_2 - c_2 - l_3$ in this order, however $c_1 - c_2$ is determined in the first step and l_1, l_2 and l_3 are determined in the second step. At the first step, set k at a certain value and calculate c_1 for $(\theta_h - \theta_s)$ at the end of c_1 . Next, calculate c_2 for $(\theta_e - \theta_h)$ in the same manner and connect c_1 and c_2 in direction θ_h . In the second step, generate the path $l_1 - l_2 - l_3$ between the end

point of $c_1 - c_2$: P_t and the scooping point P_l with optimization on length of $l_1 - l_2 - l_3$. Function Z for the optimization of the length $l_1 - l_2 - l_3$ is defined in (5),

$$Z = l_1^2 + l_2^2 + l_3^2 \tag{5}$$

Constraint conditions are

$$C_1 = (x_l - x_t) - (l_1 \cos \theta_s + l_2 \cos \theta_h + l_3 \cos \theta_l) = 0 \tag{6}$$

$$C_2 = (y_l - y_t) - (l_1 \sin \theta_s + l_2 \sin \theta_h + l_3 \sin \theta_l) = 0 \tag{7}$$

Where x_l, y_l, x_t and y_t are x and y elements of P_l and P_t respectively. Function F on the aforesaid is defined as follows,

$$F = Z + \mu_1 C_1 + \mu_2 C_2 \tag{8}$$

Where μ_1 and μ_2 are Lagrange’s multipliers. Let variation of F respect to l_1, l_2, l_3, μ_1 and μ_2 be zero, i.e.:

$$\frac{\partial F}{\partial l_1} = 0, \frac{\partial F}{\partial l_2} = 0, \frac{\partial F}{\partial l_3} = 0, \frac{\partial F}{\partial \mu_1} = 0, \frac{\partial F}{\partial \mu_2} = 0 \tag{9}$$

From these equations, following relation is obtained.

$$Tl = p \tag{10}$$

where $l = (l_1 \ l_2 \ l_3)^T$ and $p = (0 \ x_l - x_t \ y_l - y_t)^T$. T is the following matrix:

$$T = \begin{pmatrix} \sin(\theta_l - \theta_h) & \sin(\theta_s - \theta_l) & \sin(\theta_h - \theta_s) \\ \cos \theta_s & \cos \theta_h & \cos \theta_l \\ \sin \theta_s & \sin \theta_h & \sin \theta_l \end{pmatrix} \tag{11}$$

If p is given, l is obtained using inverse matrix of T .

$$l = T^{-1}p \tag{12}$$

The obtained V shape is the minimum length path at a certain angle of θ_h . The total length of V shape is changed with change of θ_h . In the final step of the path planning, V shape paths at several θ_h varied between θ_s to θ_l is calculated out and V shape with the minimum length is selected as the final result of the path planning. The entire procedure with example is illustrated in Fig. 6 and 7. Scooping point P_s is located on $(0 \ 0)^T$ and loading point P_l is located on $(-6L \ 6L)$. Where L is distance between the centre pin and front axel (or rear axel). Length and distance are normalized by L in following explanation. Scooping direction θ_s is 0 degree and loading direction θ_l is 80 degree. Curve segments c_1-c_2 at several angle of θ_t are shown in Fig. 6. The intermediate angle θ_h is set at 15, 30, 45, 60 degree. Fig. 7 shows result of the minimum length of the shape with angle $\theta_h = 58$ degree. Thick line represents V shape and thin line represents clothoid curve: c_1-c_2 .

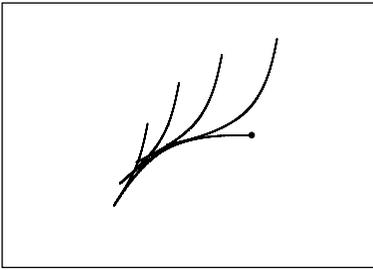


Fig. 6. c_1-c_2 at several θ_h

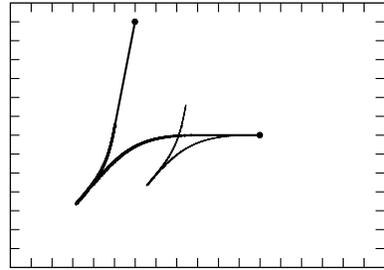


Fig. 7. V Shape Path

3 Determination of Loading and Scooping Direction

3.1 Resistance Force Applied on Bucket

Approach direction to the dump truck for loading is perpendicular to the vessel of the dump truck, however determination of the scooping direction is based on shape of the pile. If possible, the scooping direction should be perpendicular to slope of the pile (Fig.8). If the scooping direction is far from perpendicular, as shown in (b), the resistance force around the centre line of the bucket is asymmetrical, imposing undesirable stress on the bucket link mechanism.

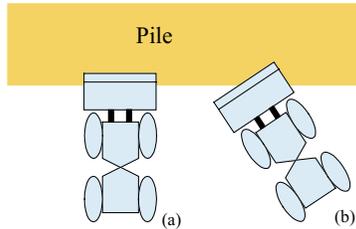


Fig. 8. Scooping angle

Large amount of resistance force from pile is applied on the bucket during scooping. Magnitude and direction of the resistance force are affected from shape of pile and trajectory of the bucket. The resistance force can be classified into several element of force. Dominant elements are penetrating force F_1 and force required moving material in front of the bucket F_2 . From results of basic experiments on resistance force between bucket and material of pile, F_1 and F_2 are formularized as follows [6];

$$F_1 = K_0 H S \gamma g \tag{13}$$

Where H is depth of pile at the tip of bucket, S is cross sectional area of the tip of bucket, γ is specific gravity of material of the pile and g is acceleration of gravity. K_0 is coefficient related penetration resistance for each material.

$$F_2 = \frac{\gamma H^2}{2} W \frac{1}{\sin^2 \alpha \sin(\alpha - \beta)} \frac{\sin^2(\phi + \alpha)}{\left(1 + \sqrt{\frac{\sin(\phi + \delta) \sin(\phi - \beta)}{\sin(\alpha - \delta) \sin(\alpha + \beta)}}\right)^2} \quad (14)$$

Where W is width of the bucket, ϕ is internal friction angle of the material. α , β and δ are angles on geometrical configuration of the bucket and the pile. F_1 and F_2 are proportional to H and H^2 respectively. The depth of the material at the tip of the bucket changes as the bucket advances during the scooping operation. The bucket trajectory of a standard scoop is composed of the three sections shown in Fig. 9. In the first section, the base of the bucket remains flat against the ground. As the loader advances on its wheels, the bucket penetrates the pile from points A to B (section 1). From points B to C (section 2), the bucket arm moves the tip of the bucket upward along a line or curve and increases the tilt. Once filled with the piled material, the bucket moves upward almost vertically from points C to D (section 3). The horizontal length of scooping, AB and BC', depends on the capacity of the bucket and shape of the pile. In a standard scoop, the total length of scooping AC' is about 1.5 times as long as the length of the bucket base.

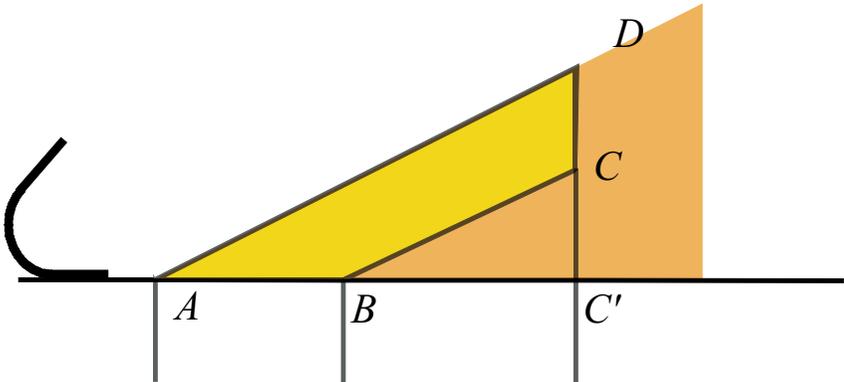


Fig. 9. Bucket trajectory model

3.2 Pile Model

The column model [1] is a useful system for estimating the interaction between the bucket and the piled material. Though simple in structure, the model can represent the shape of the pile, as well as changes in both shape and

volume. Fig. 10 shows the structure of the column model. The working area is tessellated into sections, each forming the base of a column. The height of each column represents the height of the pile at that position. The sizes of the sections and unit of height have no relation to the particle or fragment diameter of the material making up the pile.

3.3 Determination of Scooping Direction

Resistance force at each point on tip of the bucket is estimated using column model and bucket trajectory model(Fig. 9). Based on the estimation of resistance force, unbalance of the resistance force is also estimated. The scooping direction with the least unbalance of resistance force should be selected for proper scooping motion.

Discreet points are set on the scooping area as shown in Fig. 11. Let w_j be the distance from the centre of the bucket, h_{ij} be the average depth of material of the pile in the unit width at that point. The variable m_{ij} is defined as following,

$$m_{ij} = h_{ij} \cdot w_j \tag{15}$$

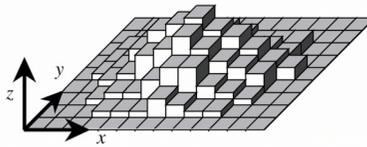


Fig. 10. Column model

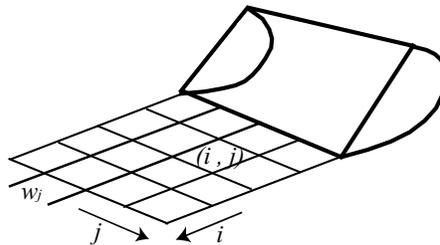


Fig. 11. Discrete point on the bucket

Thus m_{ij} is a value related to "moment" around the center of the bucket. Let m_{wj} be sum of all m_{ij} for setting points on tip of the bucket, as shown in Fig. 11.

$$m_{wi} = \sum_j m_{ij} \quad (16)$$

M is defined as the sum of m_{wi} with respect to interval point i in the advancing direction.

$$M = \sum_i m_{wi} \quad (17)$$

M is the sum of all values related to the moment applied on the bucket during one scooping cycle. M changes with changes in the scooping direction for the same initial scooping position. The smallest M is expected when the scooping is executed in a direction normal to the slope, but the value is also affected by the shape of the scooping area on the pile. The scooping direction with the smallest M is determined by comparison of the estimated M values in different directions around the normal vector in the neighboring area of the initial scoop position.

4 Experimental Result

The proposed method described above was installed and evaluated using experimental model called YAMAZUMI-2(YZ-2, Fig. 12) and the pile of fragmented granite. YZ-2 has same structure and function of a wheel loader. The length between the front axel and the rear axel is 270mm and width of bucket is 250mm. Size of the experimental pile is about 1000mm in width and 300mm in height. Particle size of the pile is 5mm. Two CCD cameras are attached to YZ-2 for shape measuring of the pile by a stereo-vision system. The length of the base line of the cameras is 100mm. The resolution of image is 640X480. Fig. 13 is image of the pile by CCD camera. The shape of the pile was obtained by applying the correlation method on the images. The column model of the experimental pile is shown in Fig. 14. Size of the basement of column is 5x5mm and the column model consists of 300x300 columns. The column model can represent 1500 x 1500 mm area in the experimental field.

Cross mark in Fig. 13 is the scooping position. In the first step of the procedure, the scooping direction is determined. For calculation of m_{ij} , 50 points at intervals of 5 mm were set on the tip of the bucket.

The advancing distance of the bucket B-C' in Fig. 9 was 150 mm, and 30 points for the calculation of m_{wj} are set on the tip of the bucket with 5 mm intervals. The normal vector of the slope in the section neighboring the scooping point on the pile was (0.889, 0.458). The temporary scooping direction was set at 27.26 degrees. The scooping direction for the estimation was changed between -50 and 50 degrees around the direction of the normal



Fig. 12. Experimental model YZ-2

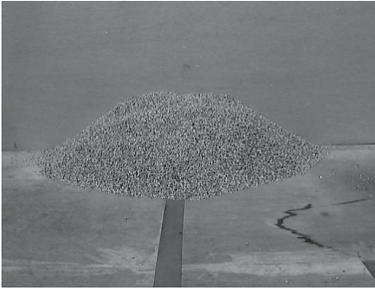


Fig. 13. Image of experimental pile

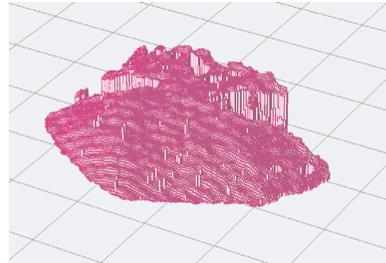


Fig. 14. Column model

vector to ascertain the minimum M . Fig. 15 shows the result. The least M was obtained at -22.24 degrees. The selected scooping direction differed slightly from the direction of the normal vector of the slope in the section neighboring the scooping point.

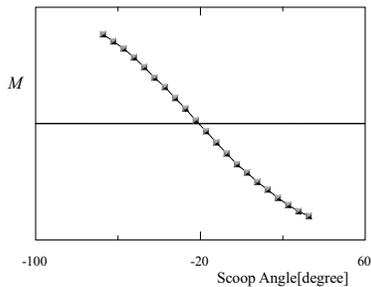


Fig. 15. Moment on the bucket

The angle at the loading position is set at 80 degree. Fig. 16 shows the length of V shape and path elements for θ_h between -22 to 80. Circle, triangle and square represent the total length of V shape, l_1 and l_3 respectively. At angle of 44 degree, the length of the V shape becomes the minimum. l_1 is negative and l_3 is positive. This fills the requirement for V shape path.

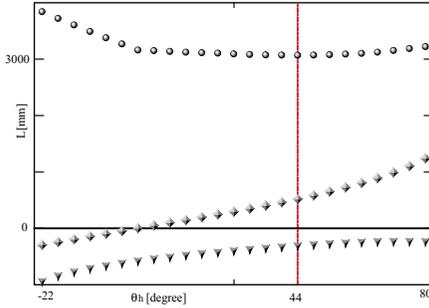


Fig. 16. Length of the path with angle θ_h

The final result of V shape generation is shown in Fig. 17 . Bold irregular shaped line represents the edge of the pile. The generated path connects the scooping position and the loading position properly.

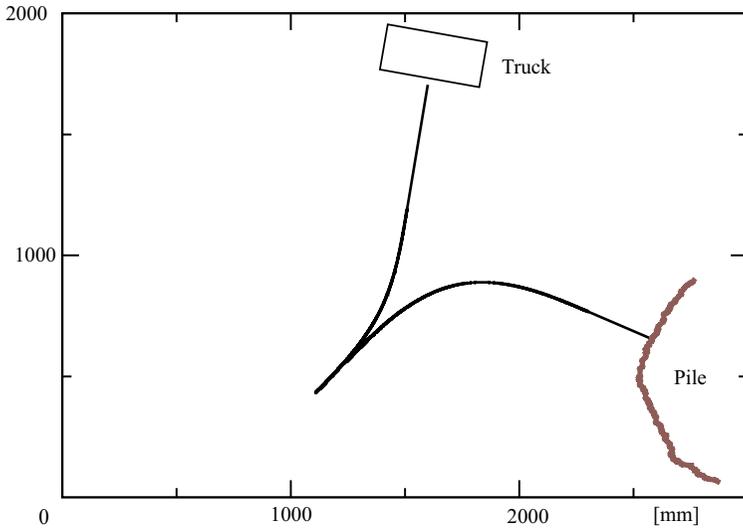


Fig. 17. V shape path

5 Conclusion

Outline of the proposed method is itemized below.

1. Irregular shape of pile is converted into numerical expression by column model. Unbalance of resistance force applied on bucket is estimated using the column model and bucket trajectory. Based on the estimation, appropriate scooping direction is determined.
2. Planned V shape path consists of two symmetrical clothoid curves and three line segments. To simplify the generation of clothoid, the curvature is limited in several value. The lengths of line segments are determined through optimization.
3. The total length of the V shape is affected by the angle at the switch back point: qh . The angle providing the minimum length of the path is obtained by comparison among these at different angle of qh between angles of the scooping position and the loading position.
4. The proposed method is installed on the experimental model. V Shape path for the given scooping position and loading position is generated. The loader can travel between the scooping position and loading position with smooth curvature. The results show that the proposed method works very well.

References

1. Sarata, S.: Model-based Task Planning for Loading Operation in Mining. Proc. of IROS, pp439-445, 2001
2. Sarata, S.: Research and Development on Unmanned Loading Operation by Wheel Loader, Proc. Rapid Mine Development, pp249-258, 2001
3. Gocho, T. et al: Autonomous Wheel-Loader in Asphalt Plant, Proc. 9th ISRAAC, 1992
4. Oshima, H. et al :Automation of Loading and Hauling Work in Mining and Quarry, Komatsu technical report, Vol.43, No.1, pp.27-39, 1997 (in Japanese)
5. Weeramhaen, Y. et al.: Path Generation for Articulated Steering Type Vehicle Using Symmetrical Clothoid, Proc. of ICIT, 2002
6. Sarata, S. et al.: Trajectory Arrangement based on Resistance Force and Shape of Pile at Scooping Motion, Proc. of ICRA, pp3488-3493, 2004

Development of an Autonomous Forest Machine for Path Tracking

Thomas Hellström, Thomas Johansson, and Ola Ringdahl

Department of Computing Science
Umeå University
Sweden
{thomash,thomasj,ringdahl}@cs.umu.se

Summary. In many respects traditional automation in the forest-machine industry has reached an upper limit, since the driver already has to deal with an excess of information and take too many decisions at a very high pace. To further automation still, introduction of semi-autonomous and autonomous functions are expected and considered necessary. This paper describes an ongoing project along these ideas. We describe the development of the hardware and software of an unmanned shuttle that shifts timber from the area of felling to the main roads for further transportation. A new path-tracking algorithm is introduced, and demonstrated as being superior to standard techniques, such as Follow the Carrot and Pure Pursuit. To facilitate the research and development, a comprehensive software architecture for sensor and actuator interfacing is developed. Obstacle avoidance is accomplished by a new kind of radar, developed for and by the automotive industry. Localization is accomplished by combining data from a Real-Time Kinematic Differential GPS/GLONASS and odometry. Tests conducted on a simulator and a small-scale robot show promising results. Tests on the real forest machine are ongoing.

1 Background and Introduction

This paper describes an ongoing project of the design and development of an autonomous path-tracking forest machine. This kind of product is part of a long-term vision in the forest industry [4], of developing an unmanned shuttle that transports timber from the felling area to the main roads for further transportation. The main advantages with such a solution are lower labor costs and less ground damages and emissions due to the lower weight of an unmanned vehicle (the cabin alone weighs several tons). The general requirements and conditions for the development of this kind of product are not addressed in this paper. It focuses instead on one of the necessary components: autonomous navigation, which involves sensing and moving safely along a user-defined path in a dynamic forest environment.

Unmanned vehicles in off-road use have been for long an active area of research and development. The mining company LKAB has been using vehicles in underground mines for many years, with reflective markers to aid the laser-based navigation system. Due to safety reasons combined with high demands on availability, these systems are no longer in full commercial operation. Localization techniques of autonomous forest machines based on a combination of odometry and artificial visual landmarks are described in [9]. Requirements and system design for a robot performing selective cleaning in young forest stands is described in [14].

The initial design and ideas underlying the project reported in this paper are described in [5]. More technical details are found in [3]. The resulting system has two modes of operation: *Path Recording*, in which the human operator drives or remote controls the vehicle along a selected path back and forth from the area of felling to the transportation road. In this phase, position, speed, heading and the operator's commands are recorded in the vehicle computer. When the vehicle has been loaded with timber (this subtask could also be done autonomously, but is not considered in this project) the operator activates *Path Tracking* mode, which means that the vehicle autonomously drives along the recorded path to the transportation road. In this paper, we describe the hardware and software developed and implemented for a pilot study on a Valmet 830 forest machine (forwarder) supplied by our industrial partner Komatsu Forest AB. The presented project started January 2003 and will end by December 2005. A continuation of the project is planned. Section 2 gives an overview of the developed hardware and software. The general design ideas behind a developed software system for sensor interfacing and actuator control are briefly described in Sect. 3. A novel algorithm for path tracking is described in Sect. 4. General experiences and directions for the future work finalizes the paper in Sect. 5.

2 System Overview

A block diagram of the developed system design is shown in Fig. 1. The major building blocks are described in this section. The system runs on two computers: the one placed on the vehicle is responsible for hardware interfacing and low-level data processing, such as data fusion in an occupancy grid. This computer communicates by a regular wireless local area network (WLAN) with the operator computer running the high level path tracking algorithms and the user interface. The two computers run with Windows XP at present, although it is designed so it can be moved to other OS'es, for instance UNIX or LINUX. This operating system independence comes from the choice of implementation language, Java and Matlab, as programs written in these languages are easily moved between different platforms. However, certain low-level drivers in C++ will have to be converted. The Java version used has varied from 1.1.8 up to 1.5, with little or no problems. The upgrading was needed partly

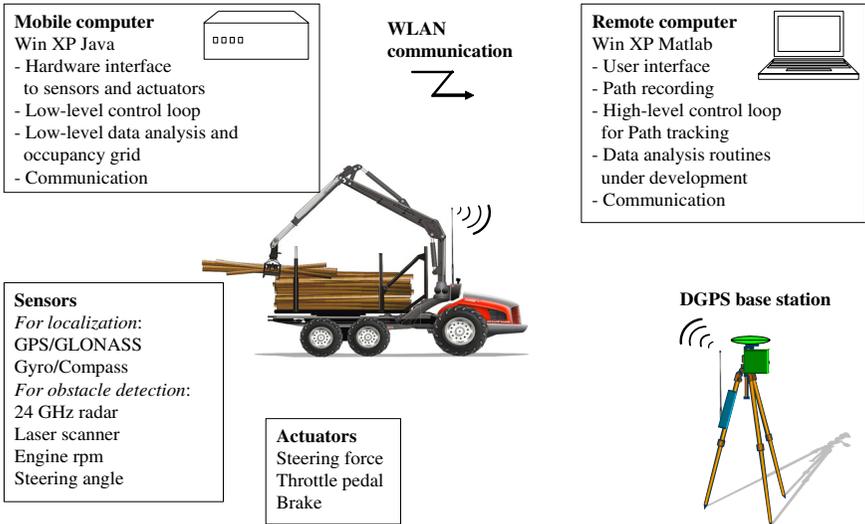


Fig. 1. Overview of the architecture of the developed system. The high- and low-level parts are split between two computers. The shown forest machine is a vision of what a future autonomous vehicle would look like.

for incorporating new features in Java, and partly to overcome compatibility problems with Matlab. In general, low-level processing and communication is implemented in Java, and high-level processing in Matlab. However, during development the Matlab environment has been used also for typical low-level operations, such as Kalman filtering. With a top-level cycle time of around 100 ms, it has not presented any problems, and it simplifies the research work considerably. In a final productified version most processing should be implemented on the mobile computer, with the operator computer in charge of the user interface only.

2.1 Communication

As mentioned in the previous section, the modules of the system may reside on different computers. The communication routines take care of the data routing, and make the actual location of each module transparent to the other modules. Communication between separate computers is done by Ethernet network, either directly through a cable or via a Wireless Local Area Network (WLAN). The WLAN equipment is the standard 10-to-54 Mbps hardware used for offices and homes. The WLAN is used for controlling the vehicle, but since the communication handling is transparent to the system, debug and

in-office tests can be done by either a cable or direct communication within the computer.

The network communication uses datagrams (by the Internet UDP protocol), i.e. small packets of data transmitted with no control over their arrival, and therefore no acknowledge of received packets is obtained. The amount of data that travels through the network is very small, currently the largest packets are about 1200 bytes, and are transmitted every 200 ms. Most packets contain only a few bytes of payload, usually one reading of a sensor. A conservative estimate on the communication bandwidth needed is about 200 Kbps. (20 different types of packets, 100 bytes per packet, 100 ms. period). This amounts to 2% of the bandwidth of a 10 Mbps. link. The delay in the network is difficult to measure, since the two participating computers usually do not have synchronized clocks, but the estimate from preliminary tests is of less than 10 ms.

2.2 Sensors and Actuators

Sensors are primarily required for localization of the vehicle and obstacle detection. The performance of various sensor types and analysis algorithms is closely tied to the physical vehicle, on which the sensors are mounted. This means that the approach with multiple target machines during development (further described in Sect. 2.3) is of limited value for evaluation and development of sensor hardware/software. E.g., the range of sensors is normally fixed, and can not be scaled up in the same way as path tracking for example.

One major sensor type for obstacle detection on the forest machine is radar. The advantage of a radar, compared to sensors based on ultrasound or light, is the ability to view obstacles reliably during bad weather conditions, like snow, fog or rain. One of the radar types used in the project is a pair of the Sequential Lobing Radar C1, produced by Tyco - M/A-COM, USA (primarily for the automotive industry). This radar works with 24 GHz frequency and is based on the monopulse theory [10]. Target range and bearing can be obtained by transmitting and receiving pulses. The range is estimated by matching the received pulse to an internally time-delayed one. An estimation of target-bearing can be obtained by using a receiver antenna with switchable lobe characteristics.

The occupancy grid is 2-dimensional (100x100 cells) and move along with the vehicle. Although the system is working under a flat-ground assumption, the vehicle and range sensors are not assumed to be parallel to the flat ground. Their poses are fully 3-dimensional with 6 degrees of freedom, and the subsystem is designed to handle any pose set to a sensor. Although the pose of each range sensor has 6 degrees of freedom, a range sensor is approximated to have a 2-dimensional sensing plane, aligned with the xy-plane of its pose, where the x-axis equals the line-of-sight.

An RTK DGPS (Real-Time Kinematics Differential GPS) satellite navigation system, Javad Maxor, is the primary sensor for the vehicle's position. A

stationary GPS receiver is connected by a radio link to a mobile GPS receiver (see Fig. 1). Correction signals for timing, ionospheric and tropospheric errors are transmitted by radio from the stationary to the mobile GPS, resulting in a centimeter accuracy under ideal conditions. The Javad receiver is capable of receiving signals from both American GPS system and Russian GLONASS system. While providing a lower accuracy than the GPS, GLONASS provides important backup, especially at high latitudes (64 degrees north), at which the work has been conducted. A second satellite receiver with the same accuracy enables computation of heading, by estimating the direction of the line between the two receiving antennas. The heading accuracy is better than 0.5 degree when both receivers operate in full Real-Time Kinematics mode. Under worse conditions, the heading and position is computed by odometry through wheel encoders or speed and turning angle sensors. The fusion of GPS and odometry sensors are done in a binary fashion based on status information from the GPS system.

The forest machine is entirely controlled via an industrial communications bus, a CAN bus. The forest machine, a Valmet 830 as shown in Fig. 2, is equipped with an articulated joint for steering. We have implemented a simple proportional integrating (PI) controller that takes care of controlling the steering angle by controlling the joint.

2.3 Development Strategy

Testing algorithms on the full-size forest machine is both impractical and inefficient. Therefore, the work has been conducted on four different target machines, each with increased complexity. As shown in Fig 2, the same main program can control any of the four target machines through a software switch board. Likewise, sensor data passes from the target machine to the main program. In this way, high-level routines like path tracking are easily developed and implemented by the use of a simple simulator [11]. The simulator implements no sophisticated sensor models, and has a simplified kinematics model for propulsion, but serves very well its purpose for debugging and testing the path-tracking algorithm described in Sect. 4. The user interface is also easily developed using the simulator as the target machine. The infrastructure for sensors or actuators (see Sect. 3), and the modules for communication between the two main computers are most conveniently developed on the small-size Pioneer AT2 robot [12]. Various types of sensors are also evaluated on this target machine. The third target machine, the Smart Pioneer AT2, has a dynamics (software determined) that more closely mimics a real forest machine, and is used for more realistic tests of the routines for turning, speed setting, and path tracking. In the current phase of the project, the system is moved to the real forest machine, and the routines for vehicle control are fine-tuned and tested. Also, reliable sensor tests are only possible using this final target machine.

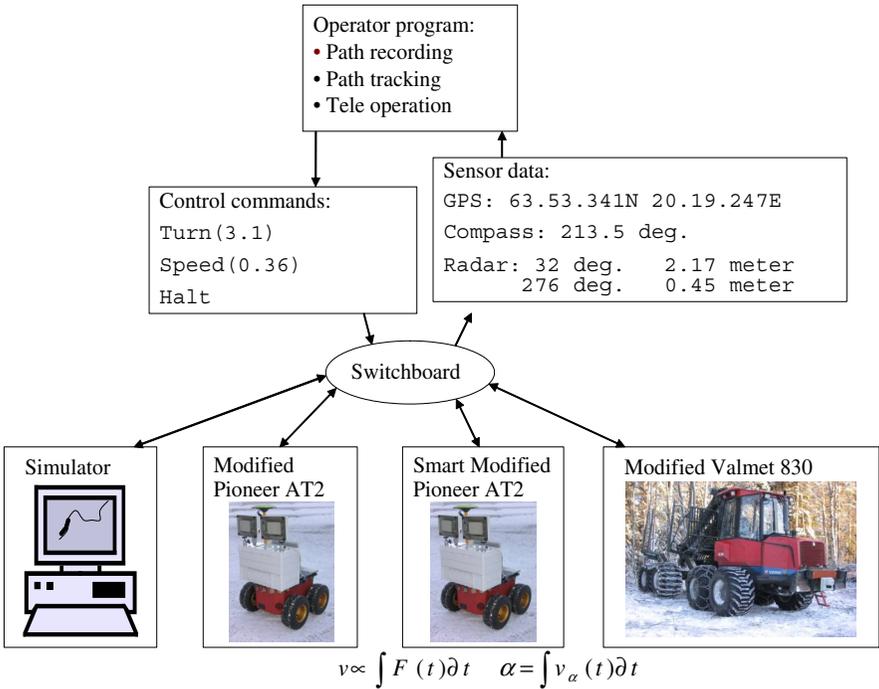


Fig. 2. The work has been conducted on four different target machines, each with increased complexity. This approach greatly simplifies the research and development of both hardware and software.

3 Infrastructure for Software Development

As part of the development work, a general framework for development of software for robots and autonomous vehicles has been constructed. A modular structure was designed with a set of requirements in mind:

- Not having to change any code in the system when replacing one sensor (e.g. a speed sensor that gets data from the wheel encoders) by another sensor (e.g. a speed sensor that gets its data from a GPS receiver).
- Not having to change any code when using different vehicles.
- Operating the system on one or more computers, connected by network.
- The analysis, design, and implementation should be object-oriented, to suit the modular design.

The system is made up of a number of software modules, some of which also represent real hardware devices. Each module can be independently loaded into the running system, and be logically connected to other modules. There are currently six major categories of modules:

Sensors (represent hardware units that deliver sensor data, e.g. speed, heading, position). Actuators - represent hardware units that control external equipment, e.g. throttle, steering angle, and brakes. Vehicle (several implementations of real and virtual vehicles. See Fig. 2). Control Panels (present data on the screen, or allow the vehicle to be manually controlled). Controllers (process sensor data and compute control signals for actuators). Proxies and Servers (facilitate transfer of sensor data and control commands over a network).

Each module is implemented as one or more classes. The modules are connected primarily by an event-driven system. There is no central control loop running. Instead the system reacts to changes in its environment. For example, a sensor signal arrives and sets up a series of method calls that ultimately leads to a change in the state of the system. Some sensor signals might just update an internal map of the surroundings, while others might stop the vehicle immediately. Other events are timed events, i.e. some action is performed repetitiously. Operator input for pure tele-operation of the vehicle is handled in the same way: a push of a button in the GUI (Graphical User Interface) or a real joystick sets up a chain of events. The Matlab program responsible for path tracking runs on another computer, and is not part of this event-driven system. Instead it polls the sensor system, in a traditional fashion, in its main control loop.

A module usually executes in a separate thread, i.e. all the modules run in parallel. Most sensing, control, user interface, and behavior are defined in modules. They communicate with other modules through connections, either by sending commands directly to another module or by listening on other modules. They also have other properties that depend on their actual type, for instance update interval for sensors, and network addresses for servers and proxies.

Every module has an associated configuration file with properties that control the module's behavior. For a sensor, this file may contain e.g. pose, network address, and filter constants. Most of these properties are set once and for all, while others are changed either by the user or by the module itself. The module can also save the changes so they take effect upon the next time the system is run. One example would be an experiment to find the most appropriate filter constant for a specific sensor; when a suitable value is found the sensor can store it in its configuration file. The next time the sensor is run, it automatically uses the saved value. The forest machine system contains more than 60 different initialization files, and to facilitate changes, as well as for providing an overview, a graphical configuration manager has been developed. The configuration manager gives the user a graphical picture of how every module is connected to other modules, and can also be used to modify individual properties for a module.

4 Path Tracking and Obstacle Avoidance

To navigate safely through the forest, a new path-tracking algorithm named *Follow-the-Past* has been developed. Traditional algorithms like Follow-the-Carrot [1] and Pure-Pursuit [2] use position information only, and sometimes run into problems that can be avoided by taking into account additional recorded information from a human driver. If the vehicle deviates from the recorded path, for example as a result of avoiding an obstacle, or because of noise in the positioning sensors, the Follow-the-Past algorithm steers like the driver, plus an additional angle, based on the distance to the path. The algorithm is described in the following section. More details and test results are found in [6] and [7].

4.1 Follow-the-Past Algorithm

While manually driving along the path, the orientation and steering angles are recorded together with the position at every moment. The recorded position (x', y') , the recorded orientation θ' and the recorded steering angle ϕ' are used by three independent behaviors:

- ϕ_β : Turn towards the recorded orientation θ'
- ϕ_γ : Mimic the recorded steering angle ϕ'
- ϕ_α : Move towards the path

Each behavior suggests a steering angle and is reactive, i.e. operates on the current input values; orientation, steering angle, and shortest distance to the path. ϕ_α uses recorded positions (x', y') and actual position (x, y) as inputs. ϕ_β uses recorded orientation θ' and actual orientation θ as inputs. ϕ_γ uses the recorded steering angle ϕ' as input. The three behaviors are fused into one action, the commanded steering angle ϕ_t , as shown in Fig. 3.

The three independent behaviors ϕ_α , ϕ_β , and ϕ_γ operate in the following fashion:

ϕ_β : *Turn towards the recorded orientation*

The angle θ' is defined as the recorded orientation at the closest point on the recorded path. This point is called the path point. ϕ_β is computed as the difference between the current orientation θ and the recorded orientation θ' :

$$\phi_\beta = \theta' - \theta. \quad (1)$$

ϕ_γ : *Mimic the recorded steering angle*

This behavior simply returns the recorded steering angle ϕ' at the path point:

$$\phi_\gamma = \phi'. \quad (2)$$

By using the recorded steering angle, the curvature of the path is automatically included in the final steering command. This is a great advantage compared to methods like Pure-Pursuit [2] and Follow-the-Carrot [1].

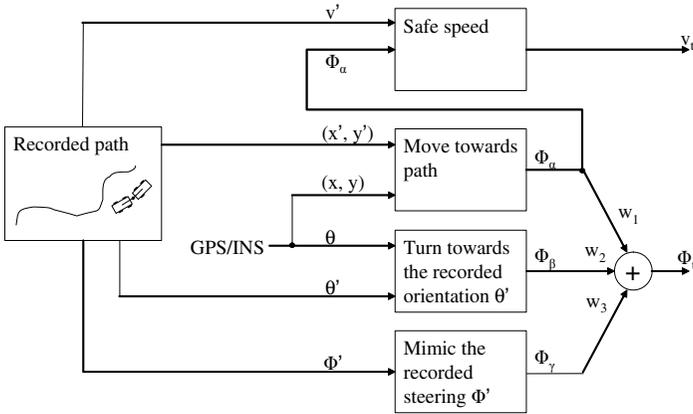


Fig. 3. Path tracking with reactive control of steering angle ϕ_t .

ϕ_α : Move towards the path

This behavior is responsible for bringing the vehicle back to the path, if the vehicle deviates for some reason from the path. Such a deviation can be caused by noise in the position signal, or by the obstacle-avoidance system. ϕ_α can be computed in many ways, e.g. by the following algorithm (refer to Fig. 4:

1. Determine the closest point on the recorded path (denoted Path Point).
2. Compute a Look Ahead Point at a Look Ahead Distance ℓ from the Path Point, in a direction δ , defined as the sum of the recorded orientation θ' and the recorded steering angle ϕ' at the Path Point, i.e.: $\delta = \phi' + \theta'$.
3. Calculate a Look Ahead Angle ψ , defined as the polar angular coordinate for the vector between the vehicle's current coordinates and the Look Ahead Point.
4. Compute ϕ_α as the difference between ψ and the angle δ , i.e.: $\phi_\alpha = \psi - \delta$.

An alternative method to compute ϕ_α can be found in [6].

Command Fusion

The three behaviors ϕ_α , ϕ_β , and ϕ_γ all return a suggested steering angle, aiming at fulfilling the goals of the respective behaviors. These three values are fused into one value ϕ_t by a weighted addition as shown in Fig. 3. In our tests, all weights have been set to 1. I.e.:

$$\phi_t = \phi_\beta + \phi_\gamma + \phi_\alpha. \quad (3)$$

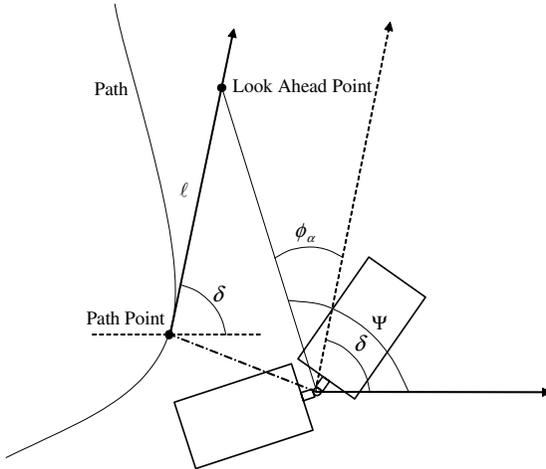


Fig. 4. Calculation of ϕ_α , which is responsible for keeping the vehicle on the track.

The expression for the fused ϕ_t is:

$$\begin{aligned}
 \phi_t &= \psi - \delta + \phi_\beta + \phi_\gamma \\
 &= \psi - (\phi' + \theta') + (\theta' - \theta) + \phi' \\
 &= \psi - \theta.
 \end{aligned}
 \tag{4}$$

Testing and Results

The developed algorithm has been tested both in a simulator for forest machines [11] and on a Pioneer robot. In this report we present only a test done with the Pioneer robot and compare the results to an implementation of the Pure-Pursuit [2] method. In this test, a Look-Ahead Distance $\ell = 1.2$ meter is used. Figure 5(a) shows results for path tracking with the Follow-the-Past method. The vehicle (thick line) is capable of following a recorded path (thin line) with almost no deviation from the path. As a reference, Fig. 5(b) shows how the vehicle behaves when using the Pure-Pursuit method under the same conditions. This path has some sharp turns, which makes it difficult for both Follow-the-Carrot and Pure-Pursuit, as they tend to “cut corners” instead of following a highly curved path. Under normal conditions, this problem is avoided by the Follow-the-Past algorithm.

To function in the forest machine application, the path-tracking behavior has been combined with VFH+ [13] for obstacle avoidance. The HIMM grid used in the original VFH+ algorithm is replaced by an occupancy grid with Bayesian updating.

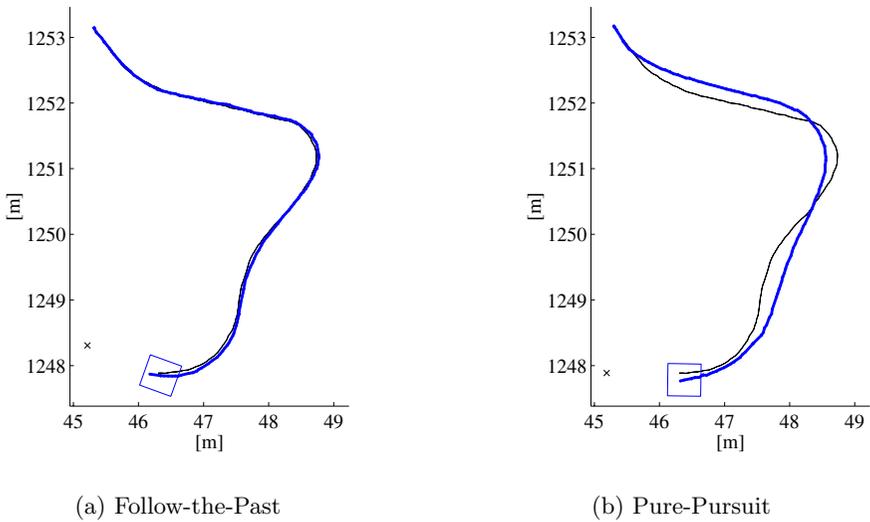


Fig. 5. Comparison between a) the Follow-the-Past and b) the Pure-Pursuit path tracking algorithms. Follow-the-Past is able to follow the path almost perfectly, while Pure-Pursuit tends to "cut corners". The examples above are from tests with the Pioneer robot.

5 Status, Experiences and Future Work

The system has been successfully implemented on the simulator and on the Pioneer robot. The developed algorithm for path tracking performs very well, and the general tools for robot software architectures have been shown to be both powerful and flexible. Sensor tests and system adjustments for the forest machine are in progress and planned to be completed during 2005.

The continuation of the project will deal with the sensing problems specific for forest environments. Techniques to distinguish obstacles from ground in uneven and hilly environment have to be developed. The specific problem with detection of human beings is also of the highest priority for a future productification of an unmanned forest machine. The future work will also involve development of ways to achieve accurate localization and estimation of heading without expensive GPS equipment.

Acknowledgements

This work was financed by The Kempe Foundations, VINNOVA, Land Systems Hägglunds, Carl Tryggers stiftelse, LKAB and Komatsu Forest AB. We acknowledge their support gratefully. The authors would also like to thank Ur-

ban Sandström for his implementation of the occupancy grid and Kalle Prorok for his work with the radar sensors.

References

1. M. J. Barton. *Controller Development and Implementation for Path Planning and Following in an Autonomous Urban Vehicle*. Undergraduate thesis, University of Sydney, Nov. 2001.
2. R. C. Coulter. Implementation of the pure pursuit path tracking algorithm. Technical Report CMU-RI-TR-92-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January 1992.
3. F. Georgsson, T. Hellström, T. Johansson, K. Prorok, O. Ringdahl, and U. Sandström. Development of an autonomous path tracking forest machine - a status report -. Technical Report UMINF 05.08, Department of Computing Science, Umeå University, 2005.
4. U. Hallonborg. Förarlösa skogsmaskiner kan bli lönsamma. *Skogforsk RESULTAT*, (9), 2003.
5. T. Hellström. Autonomous navigation for forest machines. Technical Report UMINF 02.13, Department of Computing Science, Umeå University, 2002.
6. T. Hellström and O. Ringdahl. Follow the past - a path tracking algorithm for autonomous forest vehicles. Technical Report UMINF 04.11, Department of Computing Science, Umeå University, 2004.
7. T. Hellström and O. Ringdahl. Autonomous path tracking using recorded orientation and steering commands. In *Proceedings of Towards Autonomous Robotic Systems 2005*, Imperial College London, England, 2005.
8. R. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82:35–45, 1960.
9. H. Mäkelä and K. Koskinen. Navigation of outdoor mobile robots using dead reckoning and visually detected landmarks. In *ICAR'91 Fifth International Conference on Advanced Robotics*, pages 1151 – 1156, June 1991.
10. P. Z. Peebles Jr. *Radar Principles*. John Wiley & Sons, 1998.
11. O. Ringdahl. Path tracking and obstacle avoidance for forest machines. Master's Thesis UMNAD 454/03, Department of Computing Science, Umeå University, 2003.
12. A. Robotics. Robots, AGV's & robotic sensing. <http://www.activmedia.com/>, 27 Jan. 2005.
13. I. Ulrich and J. Borenstein. VFH+: Reliable obstacle avoidance for fast mobile robots. *IEEE Int. Conf. on Robotics and Automation*, pages 1572–1577, May 1998.
14. K. Westlund and T. Hellström. Requirements and system design for a robot performing selective cleaning in young forest stands. *to be published in Journal of Terramechanics*, 2005.

Author Index

- Adachi, Tadashi 415
Althoefer, K. 517
Althoff, Heiko 567
Asama, Hajime 331
Asselt, Kees van 579
Aycard, Olivier 219, 305
- Bailey, Tim 167
Bakker, Tijmen 579
Barnes, Nick 43, 55, 79
Blair, Alan 67
Bontsema, Jan 579
Bouabdallah, Samir 429
Boudaba, Madjid 281
Broxvall, Mathias 129
Burgard, Wolfram 195
- Carsten, Joseph 117
Casals, Alicia 281
Chen, Alan 19
Christensen, Henrik I. 293, 367
Chugo, Daisuke 331
Coates, Adam 19
Cole, David T. 465
Corke, Peter 31, 231, 555
- Dankers, Andrew 79
Dissanayake, Gamini 155
Djugash, Joseph 231
Dunbabin, Matthew 31, 555
Durrant-Whyte, Hugh 257
- Einicke, Garry A. 355
Elkmann, Norbert 567
Engel, Walter 441
- Förell, Erik 93
Fraichard, Thierry 305
- Gheissari, Niloofar 43
Göktoğan, Ali Haydar 465
Grover, Richard 257
Gullstrand, Gunnar 93
- Hamner, Bradley 491
Hardwick-Jones, Rhys 465
- Hedström, Andreas 367
Hellström, Thomas 603
Hennessy, Ross 257
Horiguchi, Akira 591
Howard, Thomas M. 479
Huang, Shoudong 155
Hutangkabodee, S. 517
Hwangbo, Myung 491
- Inami, Masahiko 403
- Jensfelt, Patric 93, 293
Johansson, Thomas 603
- Kaestner, Ralf 179
Kaetsu, Hayato 331
Kawabata, Kuniaki 331
Kazerooni, H. 9
Kelly, Alonzo 269, 479, 541
Kiang, Kai-Ming 67
Kiyokawa, Kiyoshi 415
Konishi, Kaoru 505
Kotay, Keith 243
Kumar, Suresh 257
- Lamon, Pierre 529
Larsson, Johan 129
Laugier, Christian 207, 219, 305
Loy, Gareth 55
Lundberg, Carl 367
- Matsuno, Fumitoshi 403
Matsuoka, Masayoshi 19
Milford, Michael 143
Mishima, Taketoshi 331
Montemerlo, Michael 179
Müller, Joachim 579
- Nagai, Hirokazu 403
Nagatani, Keiji 389, 415
Nebot, Eduardo 167
Nelmes, Graeme 3
Ng, Andrew Y. 19
Nieto, Juan 167
Noth, André 441

- Öhman, Matti 377
Ohno, Hiroyuki 415
Okamoto, Shogo 505
Osswald, Dirk 281
- Pacchierotti, Elena 293
Peterson, Ron 243
Pfaff, Patrick 195
Pivtoraiko, Mihail 269, 541
Pradalier, Cédric 207
Prasser, David 143
- Rander, Peter 541
Reimann, Bert 567
Ringdahl, Ola 603
Roth, Stephan 491
Rus, Daniela 243
- Saffiotti, Alessandro 129
Saitoh, Hiroaki 415
Sarata, Shigeru 591
Scheding, Steve 257
Schulenburg, Erik 567
Seneviratne, L.D. 517
Shapiro, Amir 319
Shiroma, Naoji 403
Shoval, Shraga 319
Siegwart, Roland 429, 441, 529
Silver, David 117
Singh, Sanjiv 231, 491
Singh, Surya P.N. 19
Stone, R. Hugh 453, 465
Straten, Gerrit van 579
Sugimoto, Maki 403
- Sukkarieh, Salah 105, 465
Suomela, Jussi 343
- Tadokoro, Satoshi 505
Tanaka, Hiroyuki 415
Tanaka, Yutaka 389
Tay Meng Keat, Christopher 207
Thayer, Scott 117
Thompson, Paul 105
Thrun, Sebastian 19, 179
Tokuda, Kenichi 505
Tsubouchi, Takashi 591
- Usher, Kane 31
- Vasquez, Dizan 305
Visala, Arto 377
- Wang, Zhan 155
Weeramhaeng, Yossee 591
Wendt, Michael N. 355
Whalley, Matt 179
Willgoss, Richard 67
Woern, Heinz 281
Wyeth, Gordon 143
- Yagi, Yasushi 415
Yamada, Tomofumi 389
Yguel, Manuel 219
Ylikorpi, Tomi 343
Yoshida, Kazuya 415
- Zelinsky, Alex 79
Zweiri, Y.H. 517