

Springer Tracts in Advanced Robotics

Volume 28

Editors: Bruno Siciliano · Oussama Khatib · Frans Groen

Sebastian Thrun · Rodney Brooks · Hugh Durrant-Whyte (Eds.)

Robotics Research

Results of the 12th International Symposium ISRR

With 302 Figures



Springer

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. Sebastian Thrun

Stanford University

Department of Computer Science

CA 94305-9045 Stanford

USA

thrun@stanford.edu

Dr. Rodney Brooks

MIT Computer Science & Artificial

Intelligence Laboratory (CSAIL)

32 Vassar Street

MA 02139 Cambridge

USA

Dr. Hugh Durrant-Whyte

University of Sydney

Australian Centre for Field Robotics

2006 Sydney

Australia

ISSN print edition: 1610-7438

ISSN electronic edition: 1610-742X

ISBN-10 3-540-48110-9 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-48110-2 Springer Berlin Heidelberg New York

Library of Congress Control Number: 2006938801

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 2007

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: WMXDesign 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

STAR (Springer Tracts in Advanced Robotics) has been promoted under the auspices
of EURON (European Robotics Research Network)



Foreword

Robotics is undergoing a major transformation in scope and dimension. From a largely dominant industrial focus, robotics is rapidly expanding into human environments and vigorously engaged in its new challenges. Interacting with, assisting, serving, and exploring with humans, the emerging robots will increasingly touch people and their lives.

The *Springer Tracts in Advanced Robotics (STAR)* is devoted to bringing to the research community the latest advances in the robotics field on the basis of their significance and quality. Through a wide and timely dissemination of critical research developments in robotics, our objective with this series is to promote more exchanges and collaborations among the researchers in the community and contribute to further advancements in this rapidly growing field.

As one of robotics pioneering symposia, the International Symposium on Robotics Research (ISRR) has established over the past two decades some of the fields most fundamental and lasting contributions. Since the launching of STAR, ISRR and several other thematic symposia in robotics find an important platform for closer links and extended reach within the robotics community.

This twelfth edition of Robotics Research, edited by Sebastian Thrun, Rodney Brooks, and Hugh Durrant-Whyte, offers in its 14-part volume a collection of a broad range of topics in robotics. The content of these contributions provides a wide coverage of the current state of robotics research: the advances and challenges in its theoretical foundation and technology basis, and the developments in its traditional and novel areas of applications.

In addition to the collection of papers presented in the diverse technical areas, this volume reports on a panel discussion on the theme of robotics science, and on a major robotic exhibit that took place during the 2005 World Exposition (Expo 2005) in Aichi, Japan. The diversity and span of the unfolding work reveal the increased maturity and expanded scope of the

VIII Foreword

robotics field. This twelfth edition of ISRR culminates with this important reference on the current developments and new directions in the field of robotics – a true tribute to its contributors and organizers!

Stanford,
November 2006

Oussama Khatib

Preface

This volume contains a collection of papers presented at the 12th International Symposium of Robotics Research (ISRR). ISRR is the premiere meeting of the International Foundation of Robotics Research (IFRR) that covers all aspects of robotics. The 12th ISRR took place October 12-15, 2005, in San Francisco, near Fisherman's Wharf.

The 12th ISRR was attended by 70 researchers from all major geographic regions, representing many different research areas within robotics. The technical program featured 38 regular papers, which were carefully selected to cover some of the most important ongoing research in robotics. The presentations of these papers were organized into twelve thematic sessions, each of which was chaired by members of the Program Committee or officers of the IFRR. Five invited overview talks by Henrik Christensen, Hirohisa Hirukawa, Vijay Kumar, Bruno Siciliano, and Alex Zelinski informed the audience about special activities and events in the field. As is now tradition with ISRR, one evening was dedicated to an open video session chaired by Oussama Khatib, in which participants showed brief videos about their work. An open discussion on the topic of Robotics Science, organized by Ruzena Bajcsy, addressed important challenges for the field of robotics. The technical program of the 12th ISRR was complemented by a rich social program, which included a dinner cruise through the San Francisco Bay, and an excursion to nearby Alcatraz Island.

The scientific program was composed with the help of two committees: the Program Committee (PC) and the Selection Committee (SC). The PC consisted of Antonio Bicchi, Hirohisa Hirukawa, Andrew Howard, Hiroshi Ishiguro, Makoto Kaneko, Alonzo Kelly, Jean-Pierre Merlet, Paul Michael Newman, Nicholas Roy, Tomomasa Sato, Claire Jennifer Tomlin, Louis Whitcomb, and Alex Zelinsky. This committee recruited all reviewers and solicited submissions from key researchers in the field. It met at Stanford University on July 1, 2005, to make its selection from a pool of 79 submissions. The acceptance rate for submitted papers was approximately 25%. One day later, on July 2, the SC met at Stanford. This committee was comprised of Bob Bolles,

Raja Chatila, Hirochika Inoue, Oussama Khatib, and Bernie Roth. The SC finalized all selection decisions and added a number of invited speakers to the technical program. The result of this selection process was a truly outstanding technical program which, we believe, featured some of the very best work in the field.

The meeting would not have been possible without the diligent work of a great number of people, including the various committee members and technical reviewers. Special thanks go to Jennifer Hodges and Debbie Barros for assisting with the organization of the meeting and for running the registration desk. Also, Chieh-Chih (Bob) Wang organized the Web site and oversaw the submission process, which is gratefully acknowledged. Oussama Khatib, President of the IFRR, provided helpful advice all along, as did John Hollerbach and Raja Chatila. And finally, we thank all the participants of the 12th ISRR for making the meeting what it was: a premiere event in the field of robotics.

Stanford, Cambridge, and Sydney
August 2006

*Sebastian Thrun
Rodney A. Brooks
Hugh Durrant-Whyte*

Table of Contents

Physical Human Robot Interaction and Haptics

- Session Overview – Physical Human-Robot Integration and Haptics ... 3
Antonio Bicchi and Yoshihiko Nakamura

- A Unified Passivity Based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots 5
Alin Albu-Schäffer, Christian Ott and Gerd Hirzinger

- Wave Haptics: Encoderless Virtual Stiffnesses 22
Günter Niemeyer, Nicola Diolaiti and Neal Tanner

- Reality-Based Estimation of Needle and Soft-Tissue Interaction for Accurate Haptic Feedback in Prostate Brachytherapy Simulation. 34
James T. Hing, Ari D. Brooks, Jaydev P. Desai

- Haptic Virtual Fixtures for Robot-Assisted Manipulation 49
Jake J. Abbott, Panadda Marayong and Allison M. Okamura

Planning

- Session Overview – Planning 67
Nicholas Roy and Roland Siegwart

- POMDP Planning for Robust Robot Control 69
Joelle Pineau and Geoffrey J. Gordon

- On the Probabilistic Foundations of Probabilistic Roadmap Planning .. 83
David Hsu, Jean-Claude Latombe and Hanna Kurniawati

Humanoids

- Session Overview – Humanoids 101
Hirohisa Hirukawa

- Humanoid HRP2-DHRC for Autonomous and Interactive Behavior 103
S. Kagami, K. Nishiwaki, J. Kuffner, S. Thompson, J. Chestnutt, M. Stilman and P. Michel

- Android Science – Toward a New Cross-Interdisciplinary Framework... 118
Hiroshi Ishiguro

- Mimetic Communication Theory for Humanoid Robots Interacting with Humans 128
Yoshihiko Nakamura, Wataru Takano and Katsu Yamane

Mechanism and Design

Session Overview – Mechanism and Design	143
<i>Jean-Pierre Merlet</i>	
Design of a Compact 6-DOF Haptic Device to Use Parallel Mechanisms.....	145
<i>Masaru Uchiyama, Yuichi Tsumaki and Woo-Keun Yoon</i>	

Hybrid Nanorobotic Approaches to NEMS	163
<i>B.J. Nelson, L.X. Dong, A. Subramanian and D.J. Bell</i>	

Jacobian, Manipulability, Condition Number and Accuracy of Parallel Robots	175
<i>Jean-Pierre Merlet</i>	

SLAM

Session Overview – Simultaneous Localisation and Mapping	187
<i>Paul Newman and Henrik I. Christensen</i>	

Subjective Localization with Action Respecting Embedding	190
<i>Michael Bowling, Dana Wilkinson, Ali Ghodsi and Adam Milstein</i>	

D-SLAM: Decoupled Localization and Mapping for Autonomous Robots.....	203
<i>Zhan Wang, Shoudong Huang and Gamini Dissanayak</i>	

A Provably Consistent Method for Imposing Sparsity in Feature-Based SLAM Information Filters	214
<i>Matthew Walter, Ryan Eustice and John Leonard</i>	

Field Robots

Session Overview – Field Robotics	237
<i>Alonzo Kelly and Chuck Thorpe</i>	

Field D*: An Interpolation-Based Path Planner and Replanner	239
<i>Dave Ferguson and Anthony Stentz</i>	

Tradeoffs Between Directed and Autonomous Driving on the Mars Exploration Rovers.....	254
<i>Jeffrey J. Biesiadecki, Chris Leger, and Mark W. Maimone</i>	

Surface Mining: Main Research Issues for Autonomous Operations	268
<i>Eduardo M. Nebot</i>	

Robotic Vision

Session Overview – Robotic Vision	283
<i>Yoshiaki Shirai and Bob Bolles</i>	
Bias Reduction and Filter Convergence for Long Range Stereo	285
<i>Gabe Sibley, Larry Matthies and Gaurav Sukhatme</i>	
Fusion of Stereo, Colour and Contrast	295
<i>A. Blake, A. Criminisi, G. Cross, V. Kolmogorov and C. Rother</i>	
Automatic Single-Image 3d Reconstructions of Indoor Manhattan World Scenes	305
<i>Erick Delage, Honglak Lee and Andrew Y. Ng</i>	

Robot Design and Control

Session Overview – Robot Design and Control	325
<i>Claire J. Tomlin</i>	
One Is Enough!	327
<i>Tom Lauwers, George Kantor and Ralph Hollis</i>	
A Steerable, Untethered, $250 \times 60 \mu\text{m}$ MEMS Mobile Micro-Robot	337
<i>Bruce R. Donald, Christopher G. Levey, Craig D. McGray, Igor Paprotny and Daniela Rus</i>	
Some Issues in Humanoid Robot Design	357
<i>Atsuo Takanishi, Yu Ogura and Kazuko Itoh</i>	
That Which Does Not Stabilize, Will Only Make Us Stronger	373
<i>H. Kazerooni and R. Steger</i>	

Underwater Robotics

Session Overview – Underwater Robotics	399
<i>Louis L. Whitcomb and Hugh Durrant-Whyte</i>	
Improved Estimation of Target Velocity Using Multiple Model Estimation and a Dynamic Bayesian Network for a Robotic Tracker of Ocean Animals	402
<i>Aaron Plotnik and Stephen Rock</i>	
Techniques for Deep Sea Near Bottom Survey Using an Autonomous Underwater Vehicle	416
<i>Dana R. Yoerger, Michael Jakuba, Albert M. Bradley and Brian Bingham</i>	

Advances in High Resolution Imaging from Underwater Vehicles	430
<i>Hanumant Singh, Christopher Roman, Oscar Pizarro and Ryan Eustice</i>	

Learning and Adaptive Behavior

Session Overview – Learning and Adaptive Behavior	451
<i>Paolo Dario</i>	

Using AdaBoost for Place Labeling and Topological Map Building	453
<i>Óscar Martínez Mozos, Cyrill Stachniss, Axel Rottmann and Wolfram Burgard</i>	

Emergence, Exploration and Learning of Embodied Behavior	473
<i>Yasuo Kuniyoshi, Shinsuke Suzuki and Shinji Sangawa</i>	

Hierarchical Conditional Random Fields for GPS-Based Activity Recognition	487
<i>Lin Liao, Dieter Fox and Henry Kautz</i>	

Networked Robotics

Session Overview – Networked Robotics	509
<i>Tomomasa Sato and Ray Jarvis</i>	

Networked Robotic Cameras for Collaborative Observation of Natural Environments	510
<i>Dezhen Song and Ken Goldberg</i>	

Interfaces and Interaction

Session Overview – Interfaces and Interaction	523
<i>Makoto Kaneko and Hiroshi Ishiguro</i>	

Haptic Communication Between Humans and Robots	525
<i>Takahiro Miyashita, Taichi Tajika, Hiroshi Ishiguro, Kiyoshi Kogure and Norihiro Hagita</i>	

A Vestibular Interface for Natural Control of Steering in the Locomotion of Robotic Artifacts: Preliminary Experiments	537
<i>Cecilia Laschi, Eliseo Stefano Maini, Francesco Patane', Luca Ascari, Gaetano Ciaravella, Ulisse Bertocchi, Cesare Stefanini, Paolo Dario and Alain Berthoz</i>	

How Social Robots Will Help Us to Diagnose, Treat, and Understand Autism	552
<i>Brian Scassellati</i>	

Invited Overview Talk

Expo 2005 Robotics Project	567
<i>Hirohisa Hirukawa and Hirochika Inoue</i>	

Robotics Science (Panel Discussion)

Position Statement: Robotics Science	583
<i>Ruzena Bajcsy</i>	

Author Index	587
---------------------------	-----

Physical Human Robot Interaction and Haptics

Session Overview

Physical Human-Robot Integration and Haptics

Antonio Bicchi¹ and Yoshihiko Nakamura²

¹ Centro Interdipartimentale di Ricerca
“E. Piaggio”
Università di Pisa
<http://www.piaggio.ccii.unipi.it>

² Department of Mechano-Informatics
University of Tokyo <http://www.ynl.t.u-tokyo.ac.jp>

Machines and robots in the near future will share environments, and often come directly in touch with humans. This is to happen in several applications domains, including domestic applications (domotics), entertainment, assistance, cooperative manipulation tasks, teleoperation, human augmentation, haptic interfaces and exoskeletons. Physical Human-Robot Interaction (pHRI) poses many challenges, which can be summarized by the dichotomy *safety vs. performance*. The first and foremost concern, indeed, is that the robot must not hurt humans, directly nor indirectly, in regular operations nor in failures. Second, the machine is expected to perform swiftly and effectively its tasks in the service to humans.

As a consequence of this priority inversion, machines interacting with humans have different requirements than those currently met in industrial robots: while accuracy is less demanding, safety of operations is a must. Furthermore, the definition of *performance* is to be rethought, being sometimes the machine intended for quite different tasks than conventional industrial robots.

This session was designed to explore the safety and performance aspects of pHRI. The first paper, *A Unified Passivity Based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots* by Alin Albu-Schäffer, Christian Ott and Gerd Hirzinger, discusses how to effectively control a high-performance robot arm designed to minimize risks of impact with humans by reducing its inertia, and allowing joint compliance. Compare this with the traditional approach of controlling stiff and heavy arms so as to appear compliant, and consider the degree of fault tolerance of the two approaches.

The focus of three papers in this session was on design and control of high-performance haptic devices. Here, performance is to be intended at a broader, “system” level than conventionally done in robotics: indeed, the system includes both the machine and the human. The goal of haptics is that of stimulating tactile perceptions of the operator so as to provide a realistic and compelling sensation of being in touch (literally) with a remote or virtual environment.

In *Wave Haptics: Encoderless Virtual Stiffnesses*, by G. Niemeyer, N. Diolaiti, and N. Tanner, the difference between specifications of a haptic interface from those of classical servomechanisms is considered. Accordingly, a control scheme that exploits some physical behaviours of the actuation system, rather than counteracting them by imposing the controller authority, is shown to provide definite advantages. In *Reality-based Estimation of Needle and Soft Tissue Interaction for Accurate Haptic Feedback in Prostate Brachytherapy Simulation*, by J. T. Hing, A. D. Brooks, and Jaydev P. Desai, an interesting application to medical training is reported where the need for an objective interaction performance evaluation is preminent. Finally, in *Haptic Virtual Fixtures for Robot-Assisted Manipulation*, by J. J. Abbott, P. Marayong, and A. M. Okamura, it is shown how software-generated force and position signals are applied to human operators to improve the safety, accuracy, and speed of robot-assisted manipulation tasks.

A Unified Passivity Based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots

Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger

Institute of Robotics and Mechatronics, German Aerospace Center (DLR)
Alin.Albu-Schaeffer@dlr.de, Christian.Ott@dlr.de, Gerd.Hirzinger@dlr.de

Summary. In this paper we describe a general passivity based framework for the control of flexible joint robots. Herein the recent DLR results on torque-, position-, as well as impedance control of flexible joint robots are summarized, and the relations between the individual contributions are highlighted. It is shown that an inner torque feedback loop can be incorporated into a passivity based analysis by interpreting torque feedback in terms of shaping of the motor inertia. This result, which implicitly was already included in our earlier works on torque- and position control, can also be seized for the design of impedance controllers. For impedance control, furthermore, potential shaping is of special interest. It is shown how, based only on the motor angles, a potential function can be designed which simultaneously incorporates gravity compensation and a desired Cartesian stiffness relation for the link angles.

All the presented controllers were experimentally evaluated on the DLR light-weight robots and proved their performance and robustness with respect to uncertain model parameters. Herein, an impact experiment is presented briefly, and an overview of several applications is given in which the controllers have been applied.

1 Introduction

The currently growing research interest in application fields such as service robotics, health care, space robotics, or force feedback systems has led to an increasing demand for light robot arms with a load to weight ratio comparable to that of human arms. These manipulators should be able to perform compliant manipulation in contact with an unknown environment and guarantee the safety of humans interacting with them. A major problem which is specific to the implementation of light-weight robot concepts is the inherent flexibility introduced into the robot joints. Consequently, the success in the above mentioned robotics fields is strongly dependent on the design and implementation of adequate control strategies which can:

- compensate for the weakly damped elasticity in the robot joints in order to achieve high performance motion control,
- provide a desired Cartesian compliant behaviour of the manipulator,
- enable robust and fast manipulation in contact with unknown, passive environments,
- provide safety and dependability in interaction with humans.

It is commonly recognized that these control goals require measurement capabilities which clearly exceed the classical position sensing of industrial robots. The solution chosen in the case of the DLR light-weight robots (Fig. 1) was to provide the joints with torque sensors in addition to motor position sensors [12]. Additionally, a 6 dof force-torque sensor was mounted on the robot wrist.

The position control problem for flexible joint robots was extensively treated in the robot control literature [17, 19, 8, 10, 14]. However, the problem of compliant motion control for interaction with unknown environments and with humans is addressed only recently under consideration of robot flexibility. The relevance of the topics becomes clear by looking at latest hardware developments, where elasticity is deliberately introduced into the joints in order to increase the interaction performance and the safety of robots [18, 21, 7]. Due to the fact that the model structure is slightly more complex than for rigid robots, there was still a gap between theoretical solutions (which often require very accurate models and the measurement or estimation of high derivatives of the joint position) and the practical solutions commonly chosen, which are not always based on firm theoretical background.



Fig. 1. The DLR light-weight robot III

In this paper we give an overview of the controller structures for the DLR robots, sketch the passivity based theoretical framework on which the actual controllers are based, go into some detail with the Cartesian impedance controller, and shortly describe some typical applications.

2 Controller Overview

The first stage in the controller development was a joint state feedback controller with compensation of gravity and friction [2, 1]. The state vector contains the motor positions, the joint torques, as well as their first derivatives. By an appropriate parameterization of the feedback gains, the controller structure can be used to implement position, torque or impedance control. Based on this joint control structure, three different strategies for implementing Cartesian compliant motion have been realized: admittance control, which accesses the joint position interface through the inverse kinematics; Cartesian impedance control, which is based on the joint torque interface; and Cartesian stiffness control, which accesses the joint impedance controller (Fig.2).

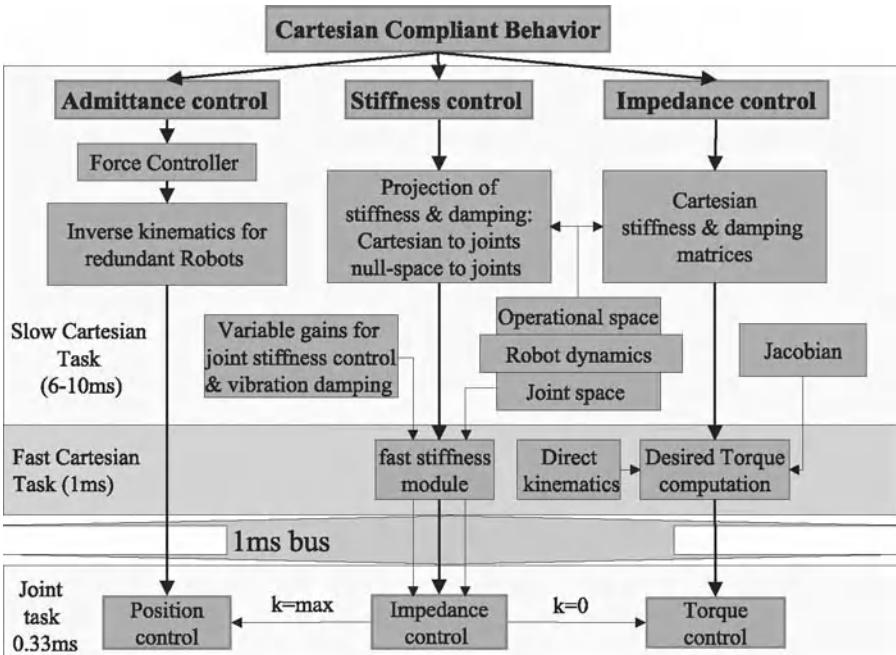


Fig. 2. Controller architecture for DLR's light-weight robots

The latest developments focused on strategies for impedance control based on a passivity approach under consideration of the joint flexibilities [16, 5, 4]. A physical interpretation of the joint torque feedback loop has been given as the shaping of the motor inertia, while the implementation of the desired stiffness can be regarded as shaping of potential energy. Therefore, the Cartesian impedance controller can be designed and analyzed within a passivity based framework in the same manner as the previously mentioned state feedback controller.

The following model structure based on [17] is assumed for the flexible joint robot:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{D}\mathbf{K}^{-1}\dot{\boldsymbol{\tau}} + \boldsymbol{\tau}_{\text{ext}} \quad (1)$$

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \boldsymbol{\tau} + \mathbf{D}\mathbf{K}^{-1}\dot{\boldsymbol{\tau}} = \boldsymbol{\tau}_m \quad (2)$$

$$\boldsymbol{\tau} = \mathbf{K}(\boldsymbol{\theta} - \mathbf{q}) \quad (3)$$

The vectors $\mathbf{q} \in \Re^n$ and $\boldsymbol{\theta} \in \Re^n$ contain the link and motor side positions respectively. $\mathbf{M}(\mathbf{q}) \in \Re^{n \times n}$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$, and $\mathbf{g}(\mathbf{q}) \in \Re^n$ are the components of the rigid body dynamics: inertia matrix, centripetal and Coriolis vector, and gravity vector. The vector $\boldsymbol{\tau} \in \Re^n$ represents the joint torques, $\boldsymbol{\tau}_{\text{ext}} \in \Re^n$ the external torques acting on the robot, and $\boldsymbol{\tau}_m \in \Re^n$ the motor torques. $\mathbf{K} = \text{diag}(K_i) \in \Re^{n \times n}$ and $\mathbf{B} = \text{diag}(B_i) \in \Re^{n \times n}$ are the diagonal, positive definite joint stiffness, and motor inertia matrices, respectively, and $\mathbf{D} = \text{diag}(D_i) \in \Re^{n \times n}$ is the diagonal positive semi-definite joint damping matrix.

3 Passivity Based Framework for Torque, Position and Impedance Control

In the following we summarize the approaches finally adopted for the DLR robots for torque, position, and impedance control and give a unified, passivity based view to these problems. Of course, the control literature for flexible joint robots contains various different other possible approaches to the problem. The best performance is theoretically given by decoupling based approaches, which provide a partially or even fully linearized closed loop system and ensure asymptotic stability also for the tracking case [17, 8, 14, 11, 15]. These controllers, however, require as a state vector the link side positions up to their third derivative and a very accurate robot model. For the DLR robots these approaches resulted in only moderate performance and robustness. The situation with back-stepping based controllers is similar to that of decoupling based approaches. On the other hand, singular perturbation based controllers are easy to implement, but their performance is theoretically and practically limited to the case of relatively high joint stiffness.

For the DLR light-weight robots, we preferred the passivity based approach described below, because it is based only on the available motor position and joint torque signals, as well as their first order derivatives and it provides a high degree of robustness to unmodeled robot dynamics and in contact with unknown environments. It provides a framework which is both theoretically sound and also practically feasible, as demonstrated by the various applications realized so far using these controllers.

3.1 Passivity Based Joint Position Control

The starting point in the control development was a joint state feedback controller given by

$$\begin{aligned}\tau_m = & -\mathbf{K}_P \tilde{\theta} - \mathbf{K}_D \dot{\theta} \\ & + \mathbf{K}_T(g(\mathbf{q}_s) - \tau) - \mathbf{K}_S \dot{\tau} + g(\mathbf{q}_s)\end{aligned}\quad (4)$$

with \mathbf{K}_P , \mathbf{K}_D , \mathbf{K}_T , and \mathbf{K}_S being positive definite diagonal matrices and with a gravity compensation $g(\mathbf{q}_s)$ based on the desired position. This constitutes an extension of the PD controllers from [19] to a full state feedback. Under some conditions related to the minimal eigenvalues of \mathbf{K}_P and \mathbf{K}_D [2, 1], the controller together with the motor side dynamics (2) can be shown to provide a passive subsystem, what in turn leads to passivity of the entire closed loop system¹, as sketched in Fig.3. In [1] it was exemplified that by ad-

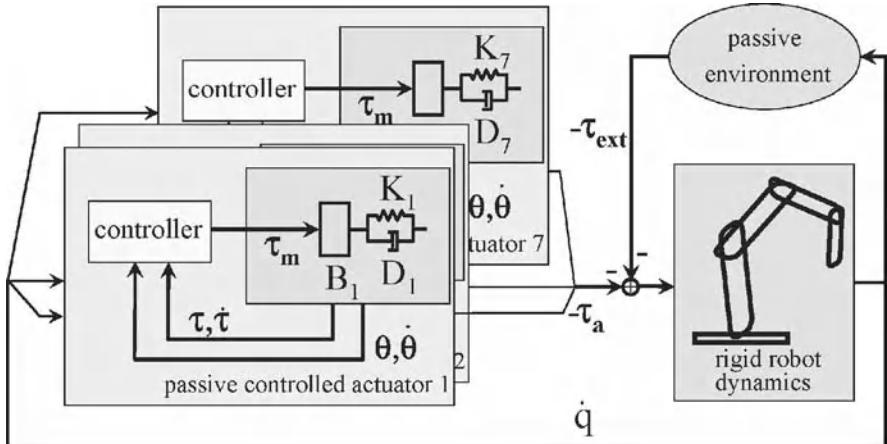


Fig. 3. Representation of the robot as a connection of passive blocks

¹ Passivity is given in this case, e.g. with respect to the variables $\{\tau_a, \dot{q}\}$, with $\tau_a = \tau + \mathbf{D}\mathbf{K}^{-1}\dot{\tau}$.

equately designing the controller gains \mathbf{K}_P , \mathbf{K}_D , \mathbf{K}_T , and \mathbf{K}_S , the structure can be used to implement a torque, position or impedance controller on joint level.

3.2 Joint Torque Control: Shaping the Actuator Kinetic Energy

In order to be able to generalize the joint level approach also to Cartesian coordinates, the idea of interpreting the joint torque feedback as the shaping of the motor inertia plays a central role [16]. It enables to directly use the torque feedback within the passivity framework and conceptually divides the controller design into two steps, one related to the torque feedback and the other to the position feedback. However, in contrast to singular perturbation approaches, the analysis does not require the two loops to have different time scales, which would imply very high bandwidth for the torque controller.

Consider a torque feedback of the form

$$\boldsymbol{\tau}_m = \mathbf{B}\mathbf{B}_\theta^{-1}\mathbf{u} + (\mathbf{I} - \mathbf{B}\mathbf{B}_\theta^{-1})(\boldsymbol{\tau} + \mathbf{D}\mathbf{K}^{-1}\dot{\boldsymbol{\tau}}). \quad (5)$$

Herein $\mathbf{u} \in \mathbb{R}^n$ is an intermediate control input. In [5] a more general form of this torque controller was presented, in which the feedback gain of $\dot{\boldsymbol{\tau}}$ is an additional independent design parameter, giving the possibility to optimize the performance and the vibration damping effect of the controller. Due to lack of space, the presentation will be restricted here to the simpler case given by (5). The torque controller leads together with (2) to

$$\mathbf{B}_\theta\ddot{\boldsymbol{\theta}} + \boldsymbol{\tau} + \mathbf{D}\mathbf{K}^{-1}\dot{\boldsymbol{\tau}} = \mathbf{u} \quad (6)$$

Comparing (2) with (6) it is clear that the effect of the torque controller is that of changing the motor inertia to \mathbf{B}_θ for the new subsystem with input \mathbf{u} .

3.3 Motor Position Based Feedback: Shaping the Potential Energy

First notice that for the joint control case, a controller of the form

$$\mathbf{u} = -\mathbf{K}_\theta\tilde{\boldsymbol{\theta}} - \mathbf{D}_\theta\dot{\boldsymbol{\theta}} + \mathbf{g}(\boldsymbol{\theta}_s) \quad (7)$$

with $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \boldsymbol{\theta}_s$ is passive with respect to the variables $(\dot{\boldsymbol{\theta}}, \mathbf{u})$. Taking into consideration the passivity of all other subsystems, this enables the conclusion of passivity for the entire closed loop system. Actually, the controller can be shown to be equivalent to the formulation (4), with $\mathbf{K}_P = \mathbf{B}\mathbf{B}_\theta^{-1}\mathbf{K}_\theta$, $\mathbf{K}_D = \mathbf{B}\mathbf{B}_\theta^{-1}\mathbf{D}_\theta$, $\mathbf{K}_T = \mathbf{B}\mathbf{B}_\theta^{-1} - \mathbf{I}$, and $\mathbf{K}_S = (\mathbf{B}\mathbf{B}_\theta^{-1} - \mathbf{I})\mathbf{D}\mathbf{K}^{-1}$. While the structure can be effectively used for position control, it has two major drawbacks when used for impedance control. First, as mentioned before, in

order to prove the asymptotic stability, some minimal values for \mathbf{K}_θ (or \mathbf{K}_P) have to be ensured. This is related to the fact that the gravity compensation is done based on the desired position. For impedance control, however, the desired stiffness may be arbitrary close to zero, making gravity compensation based on desired position not meaningful. Second, the desired stiffness relation is satisfied only locally by controllers of the type give by (7), due to additional variation of the gravity term and, in the Cartesian version, of the Jacobian. In the next subsection an approach is presented, which overcomes the mentioned shortcomings. The main idea is to design the outer loop by introducing a new control variable $\tilde{\mathbf{q}}$, which is a function of the collocated (motor) position $\boldsymbol{\theta}$ only, but is equal to the noncollocated position \mathbf{q} (link side) in every static configuration. An iterative computation method based on the contraction mapping theorem is used to calculate this variable. A passive outer loop controller can be designed in this way, while exactly fulfilling all the steady state requirements for the system. These include not only the desired equilibrium position, but also the exact stiffness relationship between the tip position and the external force. The approach can be interpreted as a shaping of the potential energy of the robot.

3.4 The Cartesian Case: Implementing Exact Desired Stiffness

In this section, the more general case of Cartesian impedance control is treated. The joint level impedance controller can be easily derived from it. In analogy to rigid robot impedance control [13], a first choice for the outer loop controller would be:

$$\mathbf{u} = -\mathbf{J}(\mathbf{q})^T (\mathbf{K}_x \tilde{\mathbf{x}}(\mathbf{q}) + \mathbf{D}_x \dot{\mathbf{x}}(\mathbf{q})) + \mathbf{g}(\mathbf{q}), \quad (8)$$

$$\tilde{\mathbf{x}}(\mathbf{q}) = \mathbf{f}(\mathbf{q}) - \mathbf{x}_s. \quad (9)$$

Herein, \mathbf{x}_s is the desired tip configuration and $\mathbf{x}(\mathbf{q}) = \mathbf{f}(\mathbf{q})$ is the tip configuration computed by the direct kinematics map \mathbf{f} . $\mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}}$ is the manipulator Jacobian. \mathbf{K}_x and \mathbf{D}_x are positive definite matrices of desired stiffness and damping. The equilibrium conditions² are then given by

$$\mathbf{K}(\boldsymbol{\theta}_0 - \mathbf{q}_0) = \mathbf{g}(\mathbf{q}_0) - \mathbf{J}(\mathbf{q}_0)^T \mathbf{F}_{\text{ext}} \quad (10)$$

$$\mathbf{K}(\boldsymbol{\theta}_0 - \mathbf{q}_0) + \mathbf{J}(\mathbf{q}_0)^T \mathbf{K}_x \tilde{\mathbf{x}}(\mathbf{q}_0) = \mathbf{g}(\mathbf{q}_0), \quad (11)$$

where the relation $\boldsymbol{\tau}_{\text{ext}} = \mathbf{J}(\mathbf{q}_0)^T \mathbf{F}_{\text{ext}}$ between the external torque and the external tip force \mathbf{F}_{ext} was used. Obviously, this leads to the desired stiffness relation $\mathbf{F}_{\text{ext}} = \mathbf{K}_x \tilde{\mathbf{x}}$ in any equilibrium position as long as $\mathbf{J}(\mathbf{q}_0)$ is invertible (what means that also \mathbf{f} is locally invertible). The following analysis is restricted to configurations in which this assumption is fulfilled.

²obtained for a constant $\boldsymbol{\tau}_{\text{ext}}$ from (1),(3),(6),(8) by setting all derivatives to zero .

It is well known that the system (1) is passive with respect to the input-output pair $\{\boldsymbol{\tau}_a + \boldsymbol{\tau}_{\text{ext}}, \dot{\mathbf{q}}\}$. This can be shown with the storage function $S_q = \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q})\dot{\mathbf{q}} + V_g(\mathbf{q})$, where $V_g(\mathbf{q})$ is a potential function for $\mathbf{g}(\mathbf{q})$. In order to ensure the passivity of the complete system, we are now looking for a control law for \mathbf{u} which determines (6) to be passive in $\{\dot{\mathbf{q}}, -\boldsymbol{\tau}_a\}$. Obviously, (8) does not satisfy the required passivity condition. It can be observed from [19, 2, 22, 16, 5] that it is possible to ensure the passivity in $\{\dot{\mathbf{q}}, -\boldsymbol{\tau}_a\}$ if \mathbf{u} is a function of $\boldsymbol{\theta}$ and its derivative only. The basic idea for the solution proposed in this paper uses the fact that, at equilibrium points, there is a one to one mapping $\boldsymbol{\theta}_0 = \mathbf{h}(\mathbf{q}_0)$ (in our case through (11)) between $\boldsymbol{\theta}_0$ and \mathbf{q}_0 :

$$\boldsymbol{\theta}_0 = \mathbf{h}(\mathbf{q}_0) = \mathbf{q}_0 + \mathbf{K}^{-1}\mathbf{l}(\mathbf{q}_0), \quad (12)$$

$$\text{with}^3 \mathbf{l}(\mathbf{q}_0) = -\mathbf{J}(\mathbf{q}_0)^T \mathbf{K}_x \tilde{\mathbf{x}}(\mathbf{q}_0) + \mathbf{g}(\mathbf{q}_0). \quad (13)$$

Furthermore, the inverse mapping \mathbf{h}^{-1} can be solved iteratively with arbitrary accuracy (see *Remark 1*).

The proposed solution consists in replacing \mathbf{q} in (8) with $\bar{\mathbf{q}}(\boldsymbol{\theta}) = \mathbf{h}^{-1}(\boldsymbol{\theta})$ and obtaining the following controller, which is *statically equivalent* to (8):

$$\mathbf{u} = -\mathbf{J}(\bar{\mathbf{q}})^T (\mathbf{K}_x \tilde{\mathbf{x}}(\bar{\mathbf{q}}) + \mathbf{D}_x \mathbf{J}(\bar{\mathbf{q}})\dot{\boldsymbol{\theta}}) + \mathbf{g}(\bar{\mathbf{q}}) \quad (14)$$

$$\tilde{\mathbf{x}}(\bar{\mathbf{q}}) = \mathbf{f}(\bar{\mathbf{q}}) - \mathbf{x}_s. \quad (15)$$

Since $\bar{\mathbf{q}}(\boldsymbol{\theta}_0) = \mathbf{q}_0$ holds at rest, it follows that the equilibrium (10),(11) and thus the desired static relation $\mathbf{F}_{\text{ext}} = \mathbf{K}_x \tilde{\mathbf{x}}(\mathbf{q}_0)$ is still valid for this new controller. This basic idea was introduced in [16, 5] for the case of gravity compensation only and was generalized in [4] in order to provide an exact link side Cartesian stiffness. The closed loop dynamics of the system results from (1), (6), and (14):

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_a + \boldsymbol{\tau}_{\text{ext}} \quad (16)$$

$$\mathbf{B}_{\boldsymbol{\theta}}\ddot{\boldsymbol{\theta}} - \mathbf{l}(\bar{\mathbf{q}}) + \mathbf{J}(\bar{\mathbf{q}})^T \mathbf{D}_x \mathbf{J}(\bar{\mathbf{q}})\dot{\boldsymbol{\theta}} + \boldsymbol{\tau}_a = \mathbf{0} \quad (17)$$

Remark 1. While in general the inverse function $\bar{\mathbf{q}} = \mathbf{h}^{-1}(\boldsymbol{\theta})$ can not be computed analytically, it is possible to approximate it with arbitrary accuracy by iteration in case that the mapping $\mathbf{T}(\mathbf{q}) := \boldsymbol{\theta} - \mathbf{K}^{-1}\mathbf{l}(\mathbf{q})$ is a contraction. The mapping $\mathbf{T}(\mathbf{q})$ has then an unique fixed-point $\mathbf{q}^* = \mathbf{T}(\mathbf{q}^*) = \bar{\mathbf{q}}$. The iteration

$$\hat{\mathbf{q}}_{n+1} = \mathbf{T}(\hat{\mathbf{q}}_n) \quad (18)$$

converges thus for every starting point (e.g. $\hat{\mathbf{q}}_0 = \boldsymbol{\theta}$) to this fixed-point, as follows from the contraction mapping theorem (see e.g. [20]):

$$\lim_{n \rightarrow \infty} \hat{\mathbf{q}}_n = \mathbf{q}^* = \bar{\mathbf{q}}. \quad (19)$$

In order for $\mathbf{T}(\mathbf{q})$ to be a contraction, it is sufficient to show that there exists an $\alpha \in \mathbb{R}$ satisfying:

³ In [16, 5], $\mathbf{l}(\mathbf{q}_0)$ is simply $\mathbf{l}(\mathbf{q}_0) = \mathbf{g}(\mathbf{q}_0)$.

$$\left\| \frac{\partial l(\mathbf{q})}{\partial \mathbf{q}} \right\| \leq \alpha < \frac{1}{\|\mathbf{K}^{-1}\|} \quad \forall \mathbf{q} \in \mathbb{R}^n. \quad (20)$$

This implies the following two inequalities:

$$\|l(\mathbf{q}_1) - l(\mathbf{q}_2)\| < \alpha \|\mathbf{q}_1 - \mathbf{q}_2\|, \quad \forall \mathbf{q}_1, \mathbf{q}_2 \in \mathbb{R}^n \quad (21)$$

$$\begin{aligned} |V_l(\mathbf{q}_1) - V_l(\mathbf{q}_2) - (\mathbf{q}_1 - \mathbf{q}_2)^T l(\mathbf{q}_2)| \\ < \alpha \|\mathbf{q}_1 - \mathbf{q}_2\|^2, \quad \forall \mathbf{q}_1, \mathbf{q}_2 \in \mathbb{R}^n \end{aligned} \quad (22)$$

with $V_l(\mathbf{q})$ being a potential function for $l(\mathbf{q})$. As a consequence of (21) it follows that

$$\begin{aligned} \|T_1(\mathbf{q}_1) - T_1(\mathbf{q}_2)\| &= \|\mathbf{K}^{-1}\| \|l(\mathbf{q}_1) - l(\mathbf{q}_2)\| \\ &< \|\mathbf{q}_1 - \mathbf{q}_2\| \end{aligned}$$

The condition (20) can always be fulfilled for a sufficiently small $\|\mathbf{K}_x\|$. A physical interpretation can be given as follows: ignoring gravity, the condition states that the desired Cartesian stiffness, transformed to joint space [9, 3] may not exceed the joint stiffness. On the other hand, in absence of external forces, the condition states that the joint stiffness should be high enough to sustain the robot in the gravity field. In the following it is therefore assumed that $\bar{\mathbf{q}}$ is known exactly. In practice, good results are obtained already by the first or second iteration step. In particular notice that by a first order approximation with $\hat{\mathbf{q}}_0 = \mathbf{q}_s$ one would obtain the second version of the controller from [22].

4 Passivity Analysis

The passivity of (17) with respect to $\{\dot{\mathbf{q}}, -\boldsymbol{\tau}_a\}$ can be shown using the following storage function:

$$S_\theta = \frac{1}{2} \dot{\boldsymbol{\theta}}^T \mathbf{B}_\theta \dot{\boldsymbol{\theta}} + \frac{1}{2} (\boldsymbol{\theta} - \mathbf{q})^T \mathbf{K} (\boldsymbol{\theta} - \mathbf{q}) - V_{\bar{l}}(\boldsymbol{\theta}), \quad (23)$$

where $V_{\bar{l}}(\boldsymbol{\theta})$ is a potential function for $\bar{l}(\boldsymbol{\theta}) = l(\bar{\mathbf{q}}(\boldsymbol{\theta}))$. It should be mentioned that a potential function for $l(\bar{\mathbf{q}}(\boldsymbol{\theta}))$ with $\boldsymbol{\theta}$ as an argument is required in (23), satisfying $\frac{\partial V_{\bar{l}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \bar{l}(\boldsymbol{\theta})^T = l(\bar{\mathbf{q}}(\boldsymbol{\theta}))^T$. A potential function $V_l(\bar{\mathbf{q}})$ in $\bar{\mathbf{q}}$, (with $\frac{\partial V_l(\bar{\mathbf{q}})}{\partial \bar{\mathbf{q}}} = l(\bar{\mathbf{q}})^T$) can easily be found:

$$V_l(\bar{\mathbf{q}}) = -\frac{1}{2} \tilde{\mathbf{x}}^T(\bar{\mathbf{q}}) \mathbf{K}_x \tilde{\mathbf{x}}(\bar{\mathbf{q}}) + V_g(\bar{\mathbf{q}}). \quad (24)$$

In [4] it is then shown that the required potential function $V_{\bar{l}}(\boldsymbol{\theta})$ is related to $V_l(\bar{\mathbf{q}})$ through

$$V_{\bar{l}}(\boldsymbol{\theta}) = V_l(\bar{\mathbf{q}}(\boldsymbol{\theta})) + \frac{1}{2} l^T(\bar{\mathbf{q}}(\boldsymbol{\theta})) \mathbf{K}^{-1} l(\bar{\mathbf{q}}(\boldsymbol{\theta})). \quad (25)$$

For robots with rotational joints, $-V_g(\bar{q})$ is lower bounded. By substituting (25) and (24) into (23), it follows that S_θ is bounded from below since all other terms are positive (quadratic). Thus S_θ represents an appropriate storage function.

The time derivative of (23) along the solutions of (17) is:

$$\begin{aligned}\dot{S}_\theta = & -\dot{\theta}^T \mathbf{J}^T(\bar{q}) \mathbf{D}_x \mathbf{J}(\bar{q}) \dot{\theta} - (\dot{\theta} - \dot{q})^T \mathbf{D}(\dot{\theta} - \dot{q}) \\ & - \dot{q}^T \boldsymbol{\tau}_a.\end{aligned}$$

The last term represents the exchanged power of the subsystem and the other terms are negative definite dissipation terms. This shows that the subsystem is indeed passive with respect to $\{\dot{q}, -\boldsymbol{\tau}_a\}$. If the robot is contacting an environment which is also passive (with respect to $\{\dot{q}, -\boldsymbol{\tau}_{\text{ext}}\}$), then the passivity of the entire system is given as a parallel and feedback interconnection of passive subsystems (Fig. 4).

As already mentioned before, the results of the passivity analysis have important implications for the robot interaction with the environment. Without going into details it should be mentioned that the storage functions from the passivity analysis can be used also as a Lyapunov function for the proof of asymptotic stability in the case of free motion [4].

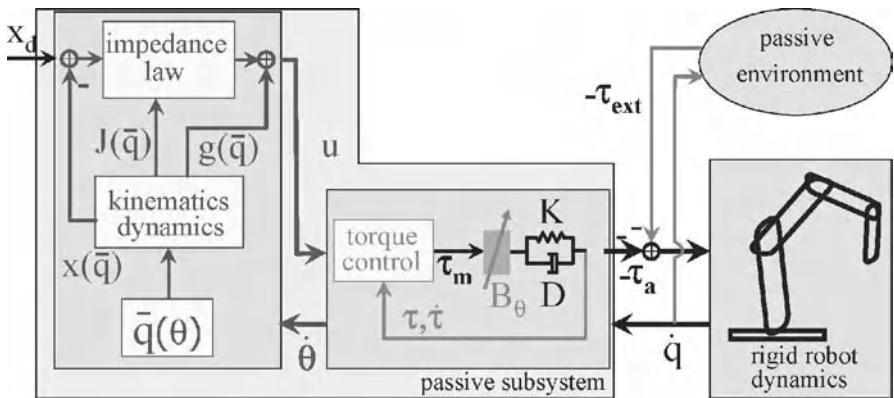


Fig. 4. Representation of the closed loop system as parallel and feedback interconnection of passive systems.

5 Experimental Evaluation

A typical impact experiment with the seven-dof DLR-light-weight-robot-II is shortly described in this section, in order to illustrate the controller perfor-

Table 1. Chosen values for the diagonal Cartesian stiffness matrix.

x	y	z	roll	pitch	yaw
4000	4000	4000	300	300	300
N m	N m	N m	Nm rad	Nm rad	Nm rad

mance. For the experiment a diagonal form of the Cartesian stiffness matrix \mathbf{K}_x , with the values of Table 1, was chosen. In the experiment a desired trajectory $z_d(t)$ along the vertical z -axis of the end-effector frame was commanded such that the robot hit a wooden surface. During this impact, the Cartesian contact force was measured by a six-dof force-torque-sensor⁴. The measurement of the external forces was done here only for the evaluation, but is not needed for the implementation of the controller. Furthermore, the end-effector coordinate $z(\mathbf{q})$ was computed from the link side angles $\mathbf{q} = \boldsymbol{\theta} + \mathbf{K}^{-1} \boldsymbol{\tau}$. The resulting motion $z(\mathbf{q})$ and the contact force F_z of the end-effector in z -direction are shown in Fig. 5. In order to evaluate the resulting impedance relationship,

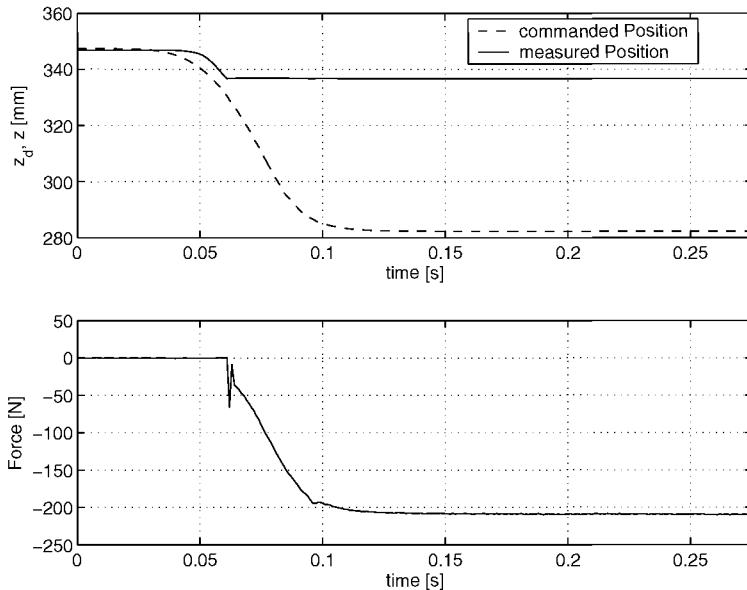


Fig. 5. The upper plot shows the desired and measured end-effector motion in z -direction during the impact experiment. In the lower plot the contact force in z -direction is displayed.

⁴ A JR3-sensor was used for this.

the ratio $\frac{F_z}{z_d - z(\mathbf{q})}$ was computed as an estimation of the stiffness⁵. This estimation is of course only valid in the steady state. The result is shown in Fig. 6. At the steady state the estimated stiffness reaches nearly the desired value of 4000N/m. The remaining difference lies in the range of known stiction effects for this robot.

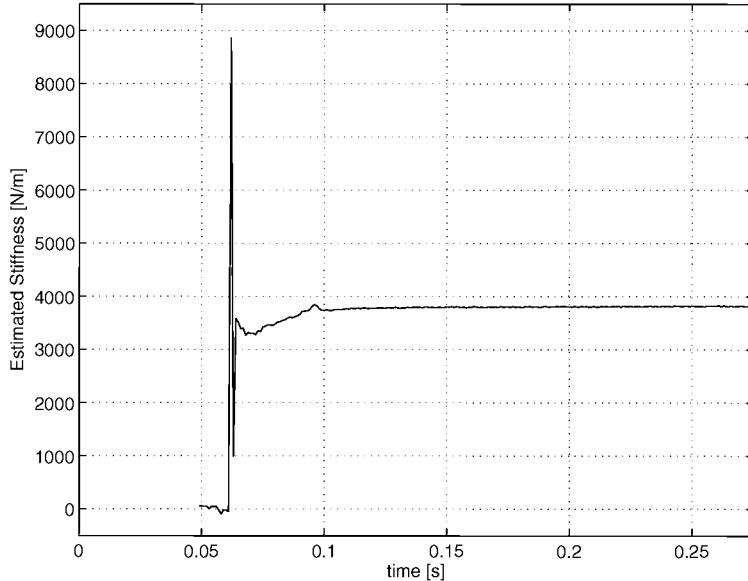


Fig. 6. Stiffness Estimation during the impact experiment.

6 Applications

In this section, some applications based on the presented controllers are shortly presented.

Piston insertion

Teaching by demonstration is a typical application for the impedance controller structure. A practical example was given with the task of teaching and automatic insertion of a piston into a motor block. Teaching is realized by guiding the robot with the human hand (Fig.7). It was initially known that the axes of the holes in the motor block were vertically oriented. In the teaching phase, high stiffness components for the orientations were commanded,

⁵ beginning at time 0.5s, when the robot movement started

while the translational stiffness was set to zero. This allowed only translational movements to be demonstrated by the human operator. In the second phase, the taught trajectory has been automatically reproduced by the robot. In this phase, high values were assigned for the translational stiffness, while the stiffness for the rotations was low. This enabled the robot to compensate for the remaining position errors. In this experiment, the assembly was executed automatically four times faster than by the human operator in the teaching phase. For two pistons, the total time for the assembly was 6s. The insertion task has been implemented before by using an industrial robot and a compliant force-torque sensor. Despite a well tuned Cartesian force controller, the insertion process had to be performed much slower, because of the well known control problems which occur in the case of hard contacts with conventional robots. Thus, the advantage of a compliant manipulator with stiffness control in assembly tasks became obvious.

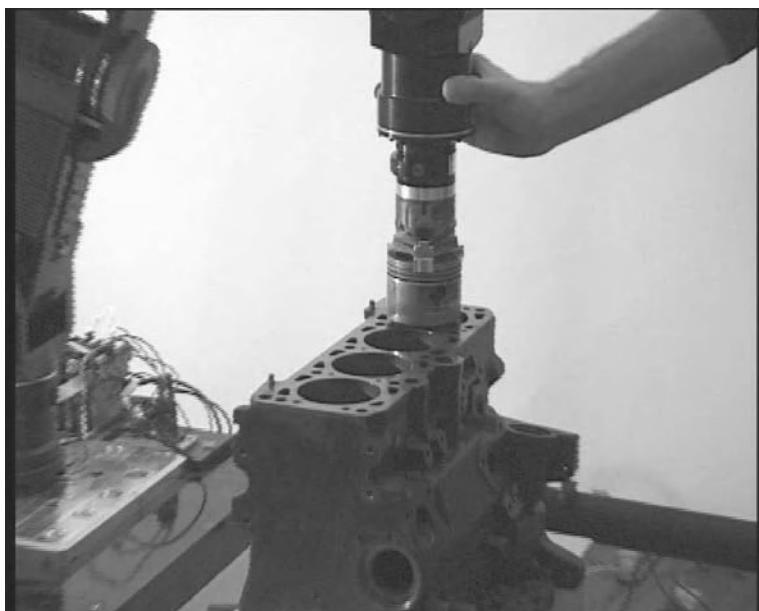


Fig. 7. Teaching phase for the automatic piston insertion using the light-weight robot II.

Wiping the table

Here the demand for a compliant behaviour of the robot also arise from reasons of safety for humans interacting with it, while the contact to the environment (table) was quite soft due to the cloth and hence not as challenging as in the

case of piston insertion. The whole task was split up into similar guiding and impedance control phases as in the piston insertion application. Fig. 8 shows a demo at the Hanover fair where the robot's elbow is deflected within its null space, while the robot continues wiping the table and applying a constant force in vertical direction.



Fig. 8. Table wiping with null space movement.

Opening a Door

In another service robotics application we used the Cartesian impedance control of the DLR light-weight robot II in order to open a door. Here the arm was used in combination with a mobile platform and the DLR-hand-II, Fig. 9.

In this application, first, the door handle was manipulated by a sequence of impedance controlled movements in order to open the door. During these motions the measurements of the joint torques provided an estimate of the contact force and thus of the current contact state.

When the mobile platform finally moved through the door hinge, the door was kept at a distance by impedance control of the arm. Therefore, instead of using the stiffness term from Section 3.4, in this stage the desired impedance was based on an appropriate potential function which has its minimum all



Fig. 9. DLR light-weight robot II while opening a door.

along a circularly shaped path with respect to a platform fixed frame. Additionally, the rotational stiffness was set to zero such that the end-effector orientation automatically adjusted.

7 Conclusions

In this paper, a unified, passivity based perspective was given to the problem of position, torque and impedance control of flexible joint robots, both on joint and Cartesian level. These methods are especially relevant for light-weight, compliant robots designed for service applications or for human-robot interaction. A physical interpretation was given for the torque controller and an energy shaping method was designed, which is based only on motor position (collocated controller), but which satisfies the static requirements formulated in terms of the robot tip. Without going into details, it is worth noting that the proposed energy shaping method can be generalized to a broader class of underactuated Euler-Lagrange systems [6], namely to such systems which can be stabilized by shaping of the potential energy only. An important advantage of these passivity-based controllers is the robustness with respect to uncertainties of the robot or load parameters, as well as to contact situations with unknown but passive environments. These properties were validated during numerous applications with the DLR light-weight robots.

References

1. A. Albu-Schäffer. *Regelung von Robotern mit elastischen Gelenken am Beispiel der DLR-Leichtbauarme*. PhD thesis, Technical University Munich, april 2002.
2. A. Albu-Schäffer and G. Hirzinger. A globally stable state-feedback controller for flexible joint robots. *Journal of Advanced Robotics, Special Issue: Selected Papers from IROS 2000*, 15(8):799–814, 2001.
3. A. Albu-Schäffer and G. Hirzinger. Cartesian impedance control techniques for torque controlled light-weight robots. *IEEE International Conference of Robotics and Automation*, pages 657–663, 2002.
4. A. Albu-Schäffer, C. Ott, and G. Hirzinger. Passivity based cartesian impedance control for flexible joint manipulators. *Proc. 6-th IFAC-Symposium on Nonlinear Control Systems, Stuttgart*, 2:111, 2004.
5. A. Albu-Schäffer, C. Ott, and G. Hirzinger. A passivity based cartesian impedance controller for flexible joint robots - part ii:full state feedback, impedance design and experiments. *ICRA*, pages pp. 2666–2673, 2004.
6. A. Albu-Schäffer, C. Ott, and G. Hirzinger. Constructive energy shaping based impedance control for a class of underactuated euler-lagrange systems. *ICRA*, pages 1399–1405, 2005.
7. A. Bicchi, G. Tonietti and M. Bavaro, and M. Piccigallo. Variable stiffness actuators for fast and safe motion control. *11th International Symposium of Robotics Research (ISRR)*, oct. 2003.
8. B. Brogliato, R. Ortega, and R. Lozano. Global tracking controllers for flexible-joint manipulators: a comparative study. *Automatica*, 31(7):941–956, 1995.
9. S. Chen and I. Kao. Simulation of conservative congruence transformation conservative properties in the joint and cartesian spaces. *IEEE International Conference of Robotics and Automation*, pages 1283–1288, 2000.
10. A. DeLuca. Feedforward/feedback laws for the control of flexible robots. *IEEE International Conference of Robotics and Automation*, pages 233–240, 2000.
11. A. DeLuca and P. Lucibello. A general algorithm for dynamic feedback linearization of robots with elastic joints. *IEEE International Conference of Robotics and Automation*, pages 504–510, 1998.
12. G. Hirzinger, A. Albu-Schäffer, M. Hähnle, I. Schaefer, and N. Sporer. On a new generation of torque controlled light-weight robots. *IEEE International Conference of Robotics and Automation*, pages 3356–3363, 2001.
13. N. Hogan. Impedance control: An approach to manipulation, part I - theory, part II - implementation, part III - applications. *Journ. of Dyn. Systems, Measurement and Control*, 107:1–24, 1985.
14. T. Lin and A.A. Goldenberg. Robust adaptive control of flexible joint robots with joint torque feedback. *IEEE International Conference of Robotics and Automation*, RA-3(4):1229–1234, 1995.
15. C. Ott, A. Albu-Schäffer, and G. Hirzinger. Comparison of adaptive and non-adaptive tracking control laws for a flexible joint manipulator. *IROS*, 2002.
16. C. Ott, A. Albu-Schäffer, and G. Hirzinger. A passivity based cartesian impedance controller for flexible joint robots - part i:torque feedback and gravity compensation. *ICRA*, pages pp. 2659–2665, 2004.
17. M. Spong. Modeling and control of elastic joint robots. *IEEE Journal of Robotics and Automation*, RA-3(4):291–300, 1987.
18. S. Sugano. Human-robot symbiosis. *Workshop on Human-Robot Interaction, ICRA*, 2002.

19. P. Tomei. A simple PD controller for robots with elastic joints. *IEEE Transactions on Automatic Control*, 36(10):1208–1213, 1991.
20. M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice-Hall, 1978.
21. M. Zinn, O. Khatib, B. Roth, and J.K. Salisbury. A new actuation approach for human friendly robot design. *Int. Symp. on Experimental Robotics, Ischia*, 2002.
22. L. Zollo, B. Siciliano, A. De Luca, E. Guglielmelli, and P. Dario. Compliance control for a robot with elastic joints. *Proceedings of the 11th International Conference on Advanced Robotics, Coimbra, Portugal*, pages pp. 1411–1416, june 2003.

Wave Haptics: Encoderless Virtual Stiffnesses

Günter Niemeyer¹, Nicola Diolaiti², and Neal Tanner¹

¹ Stanford Telerobotics Lab, USA {gunter.niemeyer, tanner}@stanford.edu

² Stanford AI-Robotics Lab, USA and DEIS, University of Bologna, Italy
ndiolaiti@deis.unibo.it

Summary. Haptic rendering commonly implements virtual springs using DC motors with current amplifiers and encoder-based position feedback. In these schemes, quantization, discretization, and delays all impose performance limits. Meanwhile the amplifiers try to cancel the electrical motor dynamics, which are actually beneficial to the haptic display.

We present an alternate approach that fully embraces and utilizes all electrical dynamics, following two insights: First, the electrical inductance L can serve as a stiffness, providing a natural sensor-less coupling between the virtual environment and the user. Second, the electrical resistance R can serve as part of a wave transformation. Implementing virtual objects in a wave domain provides complete robustness to servo delays or discretization.

The resulting system requires only a simple voltage drive circuit. Built upon the physical behaviors, it can outperform traditional approaches achieving higher virtual stiffness. Encoder feedback is only required for absolute position information, with damping and velocity information inherently available from back-EMF effects. A prototype system has been implemented and confirms the promise of this novel paradigm.

1 Introduction

Stable implementation of stiff virtual environments remains a challenge for kinesthetic force feedback devices with impedance causality. In particular, the traditional approach consists of a digital control loop using discretized and quantized position readings, as seen in Fig. 1. Force is actuated by means of a DC motor controlled by a current amplifier, in turn fed by a constant force command during each servo cycle.

It has been recognized that the maximum achievable stiffness with such an approach is limited by the lack of information to the controller caused by time discretization [1, 2] and position quantization [3, 4] related to the use of encoders as position sensors. Therefore the intrinsic friction of the device and possibly the damping added by user's grasp become essential in

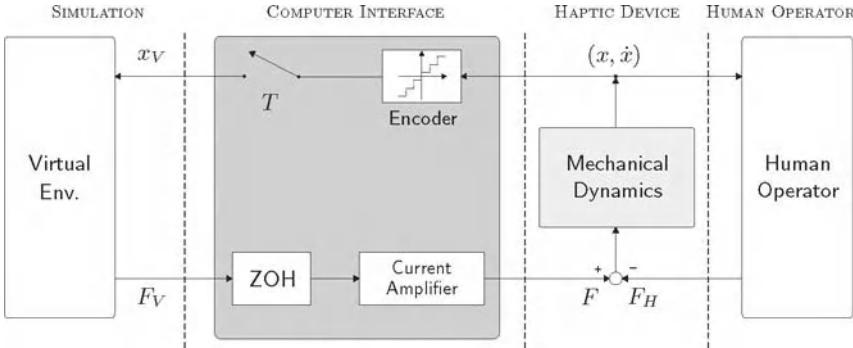


Fig. 1. The traditional implementation of stiff virtual environments

stabilizing the haptic rendering. In effect, the virtual environment can only be rendered for a limited frequency range. Alternatively, the use of analog position measurements and time continuous feedback has been explored in [5]. The electrical current amplifiers include their own internal feedback to regulate the motor current. They aim to reject back-EMF while speeding up the L-R dynamics.

In the following we adopt a different perspective. We use the electrical resistance and back-EMF to implicitly obtain velocity information and enable appropriate viscous damping. We also use the electrical inductance to create a stiffness. Built out of natural dynamics, these effects are always available at high frequency together with the controlled lower frequencies, creating performance beyond traditional approaches. Furthermore, using a wave variable description borrowed from telerobotics, the implementation is entirely insensitive to servo delays.

2 Exploiting the Motor Dynamics

Though generally ignored under the assumptions of an ideal actuator and perfect current amplification, the electrical motor dynamics are well known to be:

$$\begin{aligned} e_A(t) &= Ri(t) + L \frac{di(t)}{dt} + e_{BE}(t) \\ e_{BE}(t) &= k_T \dot{x}(t) \end{aligned} \tag{1}$$

where e_A is the applied voltage to the armature circuit consisting of the resistance R , the inductance L and the back-EMF voltage e_{BE} . The mechanical dynamics are given as:

$$\begin{aligned} m\ddot{x}(t) &= F(t) - c(\dot{x}(t)) - F_H(t) \\ F(t) &= k_T i(t) \end{aligned} \tag{2}$$

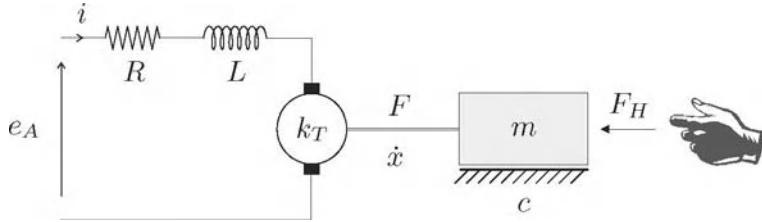


Fig. 2. The electrical and mechanical DC motor dynamics

where m is the rotor inertia, c is the (nonlinear) friction and F_H is the user torque *opposing* the motion \dot{x} of the rotor. The torque constant and back-EMF constant are the same physical parameter and are both denoted by k_T . The equations are illustrated in Fig.2 and represented in block diagram form in Fig. 3.

The actuator converts electrical into mechanical energy and thus the elements R and L can be easily mapped into the mechanical domain. For example, it is well known that the back-EMF voltage together with the resistance can be used to increase the apparent viscous friction [6] or to obtain an accurate measurement of the velocity. The resistance R maps into a mechanical viscous damper

$$B_R = \frac{k_T^2}{R} \quad (3)$$

Similarly, the inductance L provides energy storage and can be interpreted as a spring of stiffness:

$$K_L = \frac{k_T^2}{L} \quad (4)$$

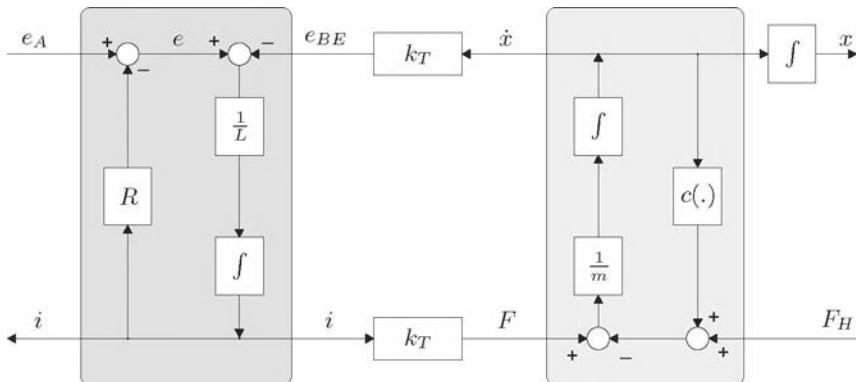


Fig. 3. Block diagram of the DC motor dynamics

The equivalent spring and damper are connected in series between the rotor inertia and the energy conversion element, as seen in Fig. 4. Because of the series connection, the damper dominates the low frequency behavior and the stiffness is often overlooked.

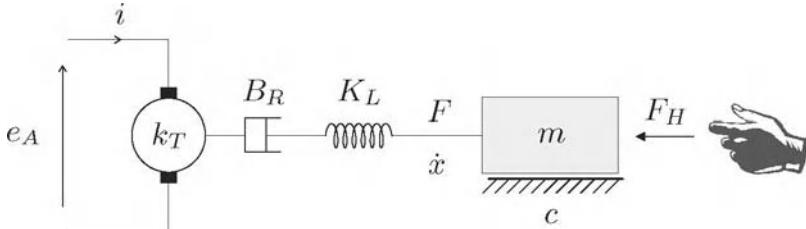


Fig. 4. Rotor inductance L and resistance R can be interpreted as series connection of a spring ($K_L = k_T^2/L$) and a viscous damper ($B_R = k_T^2/R$)

With low inductance motors commonly used in haptics, K_L creates a very high stiffness. For example, for the Maxon RE 25 motors found in the PHAN-ToM 1.0 with values of $k_T = 43.8$ mNm/A and $L = 0.83$ mH, we have $K_L = 2.31$ Nm/rad. With an approximate 8 : 1 gear ratio and lever arm of 14 cm the corresponding tip stiffness reaches 7500 N/m. For comparison, the maximum stable value of a virtual spring implemented according to the scheme of Fig. 1 has been found to be approximately 1100 N/m [4].

Therefore, it is worthy to develop a control scheme that takes advantage of the *built-in spring* present in each motor for the haptic rendering of stiff virtual environments. Since K_L is a *physical* element of the system, it is not affected by the non-idealities of the digital control loop that cause energetic inconsistencies and lead the system to instability. Moreover the force feedback it provides does not require any position sensing at all and operates at high frequencies.

3 Wave Variables for Haptic Rendering

For the purpose of designing a controller we consider the inductor as a series spring, retaining the resistance in the electrical domain as seen in Fig. 5. The controller imposes motion

$$\dot{x}_d(t) = \frac{1}{k_T} e \quad (5)$$

on the spring corresponding to a voltage e . The controller represents a voltage drive and the current i indicates a measurement of the torque F .

We interpret the dissipative element as part of a natural wave transform [7, 8], as shown in Fig. 6. A wave transform encodes the normal power variables of

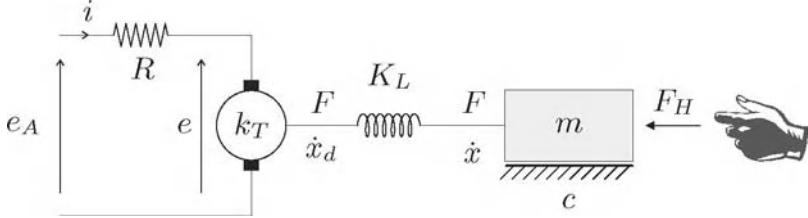


Fig. 5. Motor dynamics interpretation used in the controller design

velocity \dot{x} and force F into wave variables u and v , without loss of information or change in power flow. The wave quantities inherently describe both signal and power flow and are thus unaffected by delays or lags. In this context, according to the notation defined in [8], the wave variables are given as:

$$\begin{aligned} v(t) &:= \frac{e - Ri}{\sqrt{2R}} = \frac{B_R \dot{x}_d - F}{\sqrt{2B_R}} \\ u(t) &:= \frac{e + Ri}{\sqrt{2R}} = \frac{B_R \dot{x}_d + F}{\sqrt{2B_R}} \end{aligned} \quad (6)$$

where $\dot{x}_d(t)$ is the desired spring motion and F denotes the spring force. The equivalent viscous damping B_R serves as the wave impedance. The overall instantaneous power $P(t)$ flowing from the virtual environment to the motor is:

$$P(t) = e(t)i(t) = \dot{x}_d(t)F(t) = \frac{1}{2}u^2(t) - \frac{1}{2}v^2(t) \quad (7)$$

so that a wave variable has units of square root of Watt.

To complete the wave transform, the dark shaded area of Fig. 6 showing the two $\sqrt{2R}$ gains and the summing junction is realized by means of an *analog circuitry*. This implements:

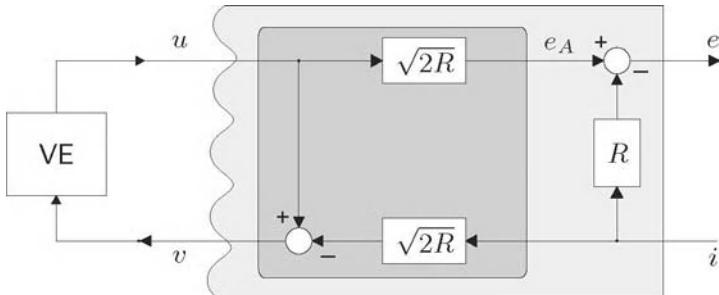


Fig. 6. Wave Transform connecting Virtual Environment VE to electrical domain

$$\begin{aligned} e_a &= \sqrt{2R} u \\ v &= u - \sqrt{2R} i \end{aligned} \quad (8)$$

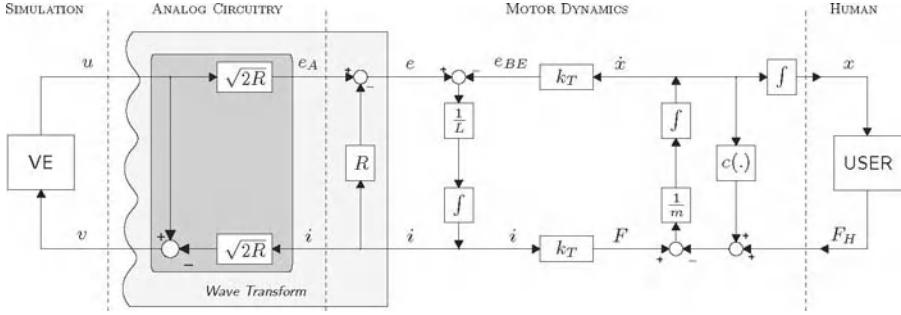


Fig. 7. Block diagram of a DC motor with the wave controller

The complete system is depicted in Fig. 7. The wave variables $u(t)$ and $v(t)$ encode the power exchanged with the motor by the simulated virtual environment. Since each wave variable carries its own power, the passivity of the interconnection is guaranteed as long as the modulus of the transfer function

$$D(s) = \frac{U(s)}{V(s)} \quad (9)$$

representing the virtual environment in wave space is at most the unity.

Because the wave variable $u(t)$ and $v(t)$ exist as real signals in the circuit, the virtual environment can be implemented in several ways. Simple transfer functions $D(s)$ can be realized in analog hardware. Alternatively, $v(t)$ and $u(t)$ can be digitized and the virtual environment implemented on a computer either in wave space or in traditional power variables by use of a second decoding digital wave transformation. In either case, any time delays or phase lags due to the discretization are guaranteed not to affect the stability of the overall system.

4 Interpretation

As discussed in detail in [7], wave variables can be used to describe an interconnection of elements. This gives us the ability to implement any passive virtual environment, with which the user interacts through the natural dynamics of the mechanical device and equivalent spring K_L . The latter can therefore be interpreted as a coupling element, resembling the virtual coupling concept of [9]. This coupling has the advantage of being a physical element and is not

affected by the stability issues of a digital implementation. This interpretation leads to the conceptual scheme of Fig. 8.

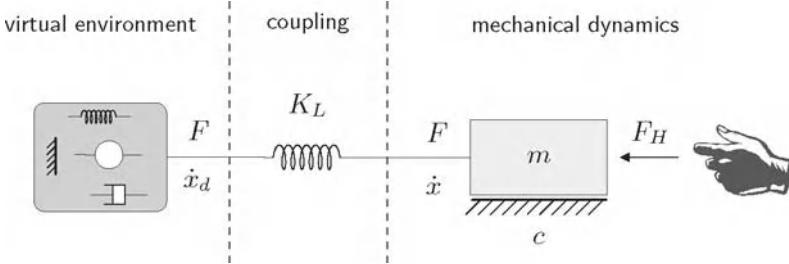


Fig. 8. The inductance acts as *physical* coupling with the virtual environment

The two most extreme passive environments are free motion and rigid contact. Both imply an infinite frequency response, as motion occurs immediately for any force in the former and forces are immediately created for any motion in the latter. Causality of an impedance device clearly favors free motion and challenges rendition of rigid contact.

In the wave domain, both of these environments are easily expressed. Free motion avoids all forces ($F = 0$) and reflects all power carried by the incoming wave $v(t)$ back by means of $u(t)$ as:

$$u(t) = v(t) \quad \iff \quad i = 0 \quad \forall e \quad \text{or} \quad F = 0 \quad \forall \dot{x}_d \quad (10)$$

where (6) converts the wave relation into the power variable description.

Dually, a rigid contact also reflects all power by suppressing any motion ($\dot{x}_d = 0$) as:

$$u(t) = -v(t) \quad \iff \quad e = 0 \quad \forall i \quad \text{or} \quad \dot{x}_d = 0 \quad \forall F \quad (11)$$

Note this does not hold the applied voltage e_A at zero, but only cancels the voltage across the inductor and back-EMF. It does not short the motor, instead effectively sets $e_A = Ri$. Illustrated in Fig. 9, it implies that K_L is the maximum stiffness that can be rendered to the user.

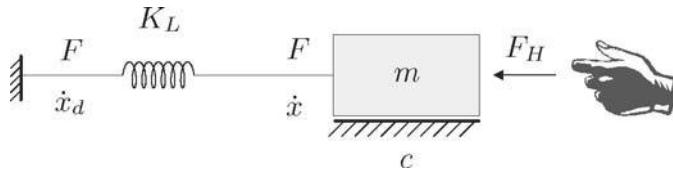


Fig. 9. Haptic interaction with a rigid virtual wall

A haptic simulation where the user experiences interactions with stiff bodies or unconstrained motion can be implemented as:

$$u(t) = \eta v(t) \quad \text{where} \quad \eta = \begin{cases} -1 & \text{in contact} \\ +1 & \text{in free motion} \end{cases} \quad (12)$$

To switch between these two values, a collision detection algorithm should take advantage of direct position measurements. An encoder is thus required to detect the collision with a unilateral constraint, but is not used to compute the force fed back to the user.

5 Implementation

Incorporating and utilizing the electrical dynamics, K_L is the maximum stiffness that can be rendered by means of a passive wave-haptic approach. Disturbances may stem from the unmodeled high frequency dynamics of the power transistors used to achieve the desired voltage input e_A , as well as from other sources of electrical noise in the control loop. These phenomena occur at much higher frequency (several hundred kilohertz) than the perceptual bandwidth of the human operator (about one kilohertz). To address these limitations, a high frequency low-pass wave filter

$$H(s) = \frac{\lambda}{s + \lambda} \quad (13)$$

can then be included in series with the virtual environment without significantly affecting the perceived transients [10]. Indeed such a filter retains passivity and adds a further series stiffness of

$$K_{filter} = \frac{k_T^2}{R} \lambda \quad (14)$$

which is significantly stiffer than the existing coupling of K_L . The transfer function $D(s)$ of the overall controller dynamics in the wave domain becomes:

$$D(s) = \eta \frac{\lambda}{s + \lambda} W(s) \quad (15)$$

where the magnitude of η can be further tuned in the interval $-1 \leq \eta \leq 1$ to remove power and introduce damping. Finally $W(s)$ may incorporate other dynamics into the simulated virtual environment.

We have implemented a simple prototype system using (12) and (15) with $\lambda = 10,000$ rad/sec, $\eta = \pm 1$, $W(s) = 1$. In particular, we use a Maxon RE25-118743 motor that features $K_L = 2.2815$ Nm/rad with the control electronics realized by means of analog circuitry. Power amplification is performed by a couple of complementary MOSFET transistors in a push-pull configuration

and an analog switch alternates between free space ($\eta=1$) and rigid contact ($\eta=-1$). Power supply limits the current provided to the motor to 1 A., Data is collected at 5 kHz using A/D conversion and a high resolution encoder (10^4 counts per revolution) to measure the motor shaft position.

Consider contact against a stiff virtual wall located at $x_W = 0$. From the encoder reading, we derive a simple binary signal indicating penetration into the wall. This logic signal commands the analog switch and selects the appropriate behavior. Preliminary results are shown in Fig. 10 and 11. In Fig. 10 we see the circuit correctly renders the behavior of free space, since when $x < 0$ the feedback torque F is zero. The voltage e_A adjusts slightly to counter the back-EMF voltage. As contact is experienced, current and torque quickly rise to their maximum value before saturation intervenes at about $F \simeq 0.021$ Nm. Better viewed in Fig. 11, the stiffness rendered during the compression phase is approximately $K \simeq 1.9$ Nm/rad, in good agreement with the expectations from the previous analysis.

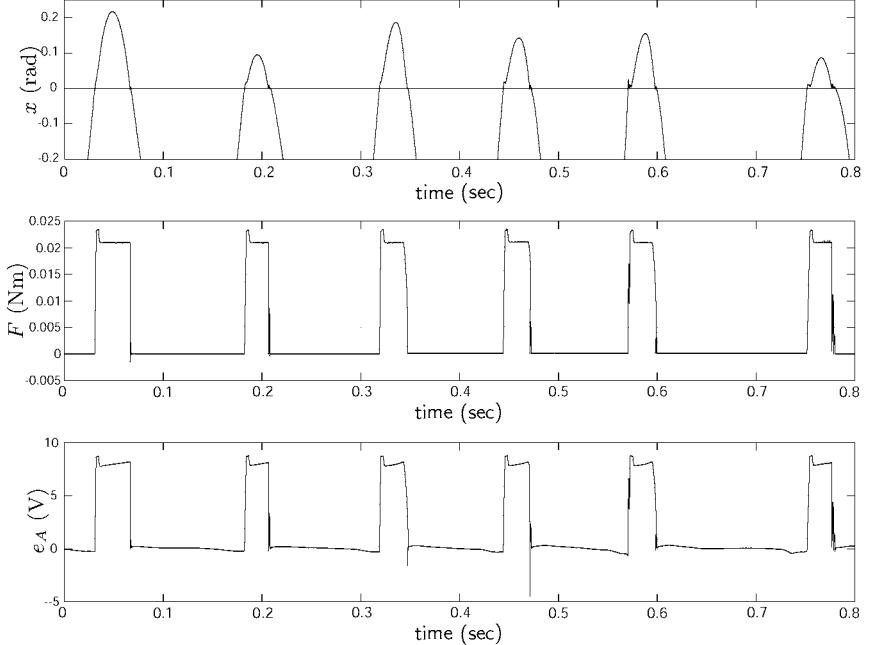


Fig. 10. Repeated contacts with the virtual wall: position, force and voltage diagrams

We note that the compression and the restitution phases appear asymmetric. This behavior is a direct consequence of voltage saturation, which is not yet included in the simple dynamics (1). When the drive voltage hits a fixed saturation limit, the back-EMF effects can not be properly canceled and the

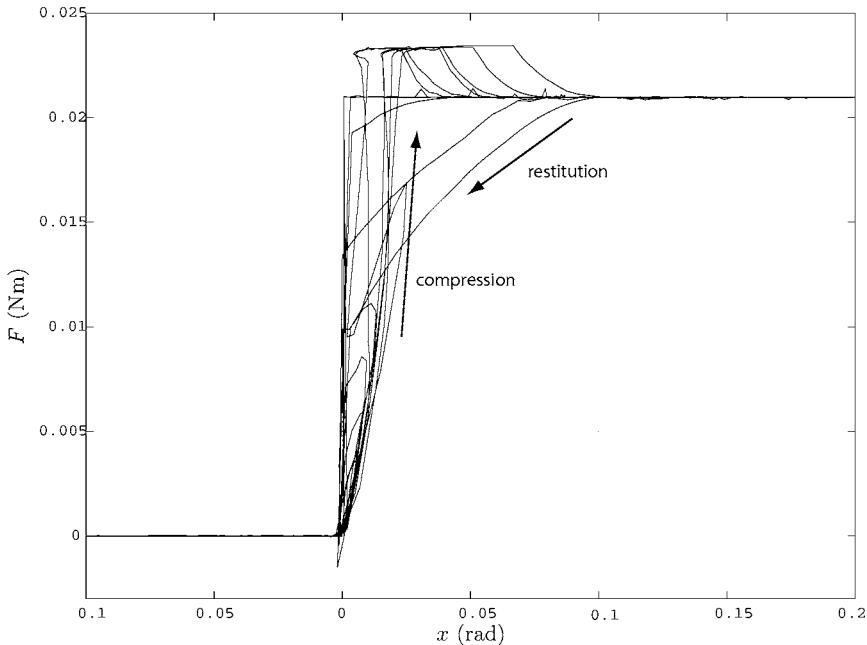


Fig. 11. Behavior of the feedback torque F during contact with the virtual wall

current is affected by motion. As the compression slows, the resisting force drops to its steady state value and, as motion begins during restitution, the restoring force drops accordingly. Effectively the spring forces are overlayed with the back-EMF's viscous damping.

As a second effect, the voltage modification caused by the saturation also shifts the endpoint x_d of the spring K_L . Recall from (5) that voltage implies motion, such that the desired behavior depicted in Fig. 9 reverts back to Fig. 4. Fortunately, as contact is broken, the behavioral switch via η resets the system for the next collision.

Finally, we find two issues that may require study for future implementations: First, knowledge of the motor resistance R is necessary to implement (8) and create the wave transformation (6) in a passive fashion. Inaccuracies lead to errors in the command voltage e_A and, as above, to drift. In particular, the resistance varies with temperature and adjustable circuitry may be required to compensate for this effect. Fortunately thermal dynamics are much slower than the dynamics involved in haptic rendering, so their influence on the stability of the overall scheme is marginal and drift may be negated by encoder feedback at a higher level. Finally, commutation in brushed DC motors introduces discontinuities into the simple dynamics (1) and may be perceived by the user. Application to brushless motors promises smoother operation.

6 Conclusions

In this paper we propose to exploit the electrical dynamics of the DC motor used to render the force feedback in virtual reality applications. For the motors commonly used in this context, the equivalent stiffness of the motor inductance is higher than the stiffness that can be achieved by means of a classical digital control loop.

We propose to take advantage of this physical spring to render stiff virtual objects, avoiding the problems related to position quantization and time discretization. In this way, the usually neglected electrical dynamics are effectively used to improve performance. Realization by means of a reliable analog circuitry is possible entirely within the electrical domain. The required components include only a sense resistor to acquire i , two gain stages, and a summation stage depicted in Fig. 6. The voltage command e_A is applied directly to the motor via a power stage, replacing the more complex current amplifiers typically used.

The virtual environment is interfaced to the motor by means of wave variables and in this domain the time delays and phase lags caused by a discrete-time implementation do not affect the energy balance and therefore the stability of the overall system.

This approach is very appealing with its intrinsic simplicity and the better use it makes of the physical components of the haptic device. It does not require assumptions on the mechanical friction to obtain stability and passivity. Conversely, the passivity is obtained constructively and the effects of non-idealities are confined behind the wave variables transform, guaranteeing intrinsic robustness to servo delay.

Acknowledgments

N. Diolaiti gratefully acknowledges support for this research which was provided, in part, by NIH Grant R33 LM 007295 and by the AGI Corporation.

References

1. J. Colgate and G. Schenkel, "Passivity of a class of sampled-data systems: Application to haptic interfaces," in *American Control Conference*, Baltimore, Maryland, June 1994, pp. 3236–3240.
2. B. Gillespie and M. Cutkosky, "Stable user-specific rendering of the virtual wall," in *ASME IMECE*, vol. DSC-Vol. 58, Atlanta, GA, November 1996, pp. 397–406.
3. J. J. Abbott and A. M. Okamura, "A sufficient condition for passive virtual walls with quantization effects," in *ASME IMECE International Symposium on Advances in Robot Dynamics and Control*, Anaheim, CA, USA, Nov. 2004, pp. 1065–1073.

4. N. Diolaiti, G. Niemeyer, F. Barbagli, J. K. Salisbury, and C. Melchiorri, "The effect of quantization and coulomb friction on the stability of haptic rendering," in *WHC '05: First WorldHaptics Conference*. Pisa, Italy: IEEE Computer Society, March 2005, pp. 237–246.
5. M. Kawai and T. Yoshikawa, "Haptic display with an interface device capable of continuous-time impedance display within a sampling period," *IEEE/ASME Transactions on Mechatronics*, vol. 9, no. 1, pp. 58–64, March 2004.
6. J. S. Mehling, J. E. Colgate, and M. A. Peshkin, "Increasing the impedance range of a haptic display by adding electrical damping," in *WHC '05: First WorldHaptics Conference*. Pisa, Italy: IEEE Computer Society, March 2005, pp. 257–262.
7. G. Niemeyer and J. Slotine, "Using wave variables for system analysis and robot control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, Albuquerque, New Mexico, April 1997, pp. 1619–1625.
8. G. Niemeyer and J. Slotine, "Telemanipulation with time delays," *International Journal of Robotics Research*, vol. 23, no. 9, pp. 873–890, September 2004.
9. B. Miller, J. Colgate, and R. Freeman, "Environment delay in haptic systems," in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 2000, pp. 2434–2439.
10. N. Tanner and G. Niemeyer, "Practical limitations of wave variable controllers in teleoperation," in *IEEE Conference on Robotics, Automation, and Mechatronics*, Singapore, 1–3 December 2004.

Reality-Based Estimation of Needle and Soft-Tissue Interaction for Accurate Haptic Feedback in Prostate Brachytherapy Simulation*

James T. Hing[†], Ari D. Brooks^{††}(MD), Jaydev P. Desai^{‡a} (PhD)

[†] Program for Robotics, Intelligent Sensing, and Mechatronics
(PRISM) Laboratory, Drexel University, Philadelphia, PA

^{††} Drexel University College of Medicine, Philadelphia, PA

Abstract – Prostate Brachytherapy is the implantation of radioactive seeds into the prostate as a treatment for prostate cancer. The success rate of the procedure is directly related to the physician's level of experience. In addition, minor deviations in seed alignment caused by gland compression/retraction, gland edema (swelling) and needle deflections can create significant areas of over or under dosage to the gland and/or injury to surrounding nerves and organs, leading to increased morbidity. Therefore, reductions in brachytherapy complication rates will be dependent on improving the tools physicians use for training to improve the accuracy of needle guidance and deployment of 'seeds' within the prostate gland. Through our novel approach of using two C-ARM fluoroscopes, we propose a reality-based approach for estimating needle and soft tissue

* This work was supported in part by National Science Foundation CAREER award IIS 0133471, NSF ITR award 0312709, and Beukenkamp Foundation.

Mr. James Hing is a graduate student in the PRISM laboratory (e-mail: jth23@coe.drexel.edu).

Dr. Ari D. Brooks is a General Surgeon specializing in Surgical Oncology at Drexel University College of Medicine (email: Ari.Brooks@DrexelMed.edu).

Dr. Jaydev P. Desai is the Director of the PRISM Laboratory, Drexel University, Philadelphia, PA 19104, USA (corresponding author, phone: 215-895-1738; fax: 215-895-1940; e-mail: desai@coe.drexel.edu).

^a Corresponding author.

interaction for the purpose of eventually developing an accurate seed placement training simulator with haptic feedback for prostate brachytherapy. By recording implanted fiducial movement and needle-soft tissue interaction forces, we can: extract the local effective modulus during puncture events, quantify tissue deformation, obtain an approximate cutting force, and build a finite element model to provide accurate haptic feedback in the training simulator for needle insertion tasks.

Index Terms – *Surgical Simulation, Soft-tissue Modeling, Prostate Brachytherapy, Local Effective Modulus.*

1 Introduction

Prostate cancer is the most common cancer in men in the United States and it is the second leading cause of cancer deaths in men (Jemal 2004). The prevalence is about 10 to 20 times the incidence because the vast majority of affected men will not die of prostate cancer. This identifies an important issue regarding the treatment of prostate cancer: the quality of life for what amounts to a normal remaining life span becomes extremely important (Hall 2003). Patients have three main choices for treatment of prostate cancer: surgical excision, radiation therapy, and expectant management/hormone therapy (Leak 2002), and each approach is associated with acceptable long-term survival. The difference between these therapies lies in the associated outcomes of treatment, i.e. acute and chronic complications/side effects that can limit one's enjoyment of life and even lead to chronic conditions requiring medical intervention, medications, surgical procedures, hospitalizations, and even death (Hall 2003). Since there are so many prostate cancer survivors at risk for chronic disabilities and decreased quality of life, reduction of complications and avoidance of adverse outcomes becomes a national health issue. Among the three main treatment options for localized prostate therapy, prostate brachytherapy has emerged as an excellent alternative for patients who meet specific criteria because it offers the benefits of a higher gland specific dose of radiation therapy without the side effects of external beam therapy (Potters 2003). The procedure is completed in one session either on an outpatient basis, or requiring an overnight hospital stay. There is no significant blood loss, making this an attractive alternative to surgery. Recent studies have indicated a disease specific survival equal to prostatectomy (Langley and Laing 2004, Sharkey 2000). All of these factors point to a potential for equal survival with lower morbidity than surgical excision or external beam

therapy. For prostate brachytherapy, the success rate of the procedure is directly related to the clinician's level of experience. Therefore, improvements in brachytherapy complication rates will be dependent on improving the tools clinicians use for training to improve the accuracy of needle guidance and deployment of 'seeds' within the prostate gland.

From the surgical simulation viewpoint, most tissue response modeling efforts in the literature are targeted towards assuming mechanical properties and developing methods to efficiently solve the tissue simulation problem for robot-assisted surgery/training. Several simulations have developed very sophisticated virtual environments that allow for plastic deformations of the material and interactions in multiple dimensions (Forest, et al. 2002, Picinbono, et al. 2001). However, it has been difficult to populate these models with data from real tissues. Simulation and modeling of needle insertions have been conducted by a number of researchers (Alterovitz, et al. 2003, Brett, et al. 2000, DiMaio and Salcudean 2002, Hong, et al. 2004, Kataoka, et al. 2002, Magill, et al. 2004, Nienhuys and van der Stappen 2003, Simone and Okamura 2002, Smith, et al. 2001, Stoianovici, et al. 1998). However, most assume linear elastic properties, homogenous tissues, and no needle deflection. Only a few groups have modeled and studied the measurement of forces during needle insertion into soft tissue and the effects of needle geometry on the deflection during needle insertions into homogenous tissues (Kataoka, Washio, Chinzei, Mizuhara, Simone and Okamura 2002, O'Leary, et al. 2003). Needle deflection is an important part of our study because it has been observed during surgical procedures of prostate brachytherapy that the needle can deflect from the initial insertion point as it is being inserted through the body by more than 10mm (Cormack, et al. 2000).

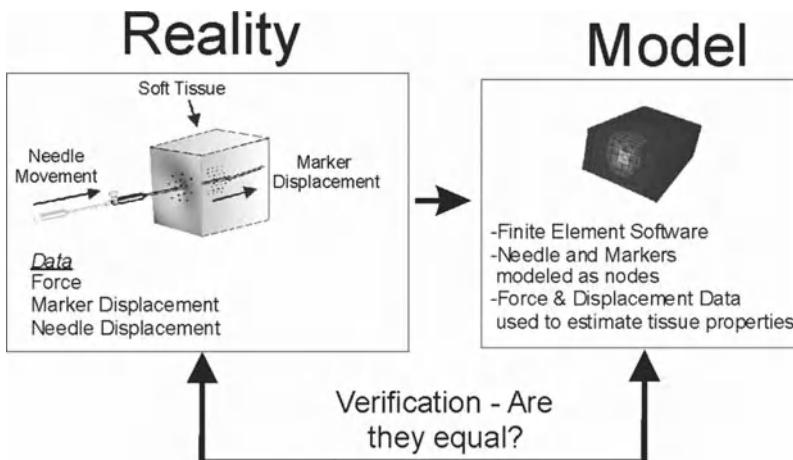


Fig. 1. Schematic of the proposed reality-based modeling approach for accurate needle and soft-tissue interaction in a training simulator for prostate brachytherapy.

To the author's knowledge, there has been no work on measuring in real time the 3-D movement of fiducials (beads) in non-homogeneous soft tissue during needle insertion. Our method of tracking beads using two C-ARMS facilitates the extraction of necessary parameters for accurate estimation of needle and soft tissue interaction. *This type of reality based modeling is critical for providing accurate haptic feedback in surgical simulation.* The findings in this study will be used to further the development of an accurate haptic feedback simulator for prostate brachytherapy training. Figure 1 shows a schematic of the proposed reality-based modeling approach.

2 Materials and Methods

Table 1 demonstrates our proposed approach for modeling needle and soft-tissue interaction during a needle insertion and withdrawal task in prostate brachytherapy. Each task is broken down into the experimental tools needed, data acquired during the experiment and computational tools used for analysis. This paper presents: a) the computation model for task 1 to estimate the local tissue stiffness prior to puncture and b) reality-based estimation of needle and soft tissue interaction for tasks 2 and 3.

Needle insertion device: The needle apparatus was designed to measure the forces on a surgical needle during insertion into soft tissue.

The insertion and withdrawal speeds varied from 1.14 mm/sec to 25.4 mm/s. The needle insertion device consisted of a geared DC motor, an incremental encoder and a JR3 precision 6 axis force/torque sensor. The JR3 sampled the force at 1000 Hz. For our experiments, we used an 18-gauge prostate seeding needle (Mick Radio Nuclear Instruments, Inc.) of length 20 cm. This is consistent with the type of needle typically used by surgeons when performing prostate brachytherapy (Figure 2).

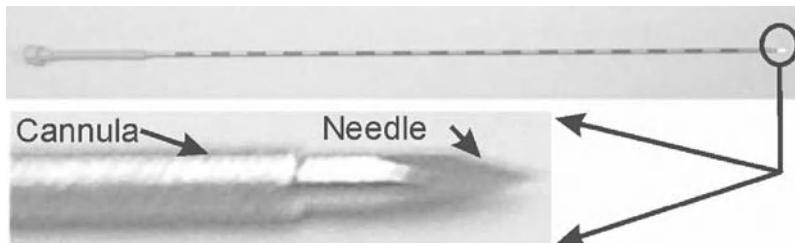
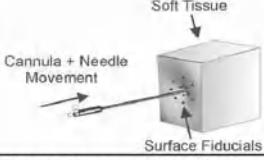
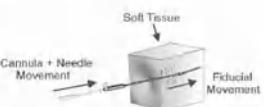
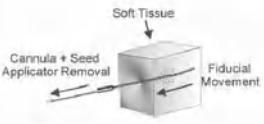


Fig. 2. Various components of a typical prostate brachytherapy seeding needle.

Soft Tissue Markers: To view the internal tissue movement during needle insertion, forty 1mm diameter stainless steel beads were inserted

into the soft tissue. These beads were chosen because their size was small enough to not affect the properties of the soft tissue or impede the needle insertion path. They also show up well under X-ray imaging. The beads are placed in a grid pattern spaced approximately 5mm apart from one another. The grid was meticulously placed to avoid occlusion between beads during imaging. Each bead was inserted perpendicular to the experiment needle path using an 18 gauge needle to an approximate depth of 10 to 20 mm from the tissue surface.

Table 1. Modeling needle and soft-tissue interaction during needle insertion and withdrawal

Tasks	Tools	Data Acquired
1. Needle Puncture 	Exp. Tools -External Vision Sys. -JR3 -Fiducials on perineum -DSpace 1103 Comp. Tools -ABAQUS -MATLAB	-Fiducial movement on skin surface -Force vs. displacement prior to skin puncture
2. Soft-tissue and needle interaction (needle insertion) 	Exp. Tools -2 C-ARMS -JR3 -Fiducials inside tissue -DSpace 1103 Comp. Tools -ABAQUS -MATLAB	-Force vs. displacement -Fiducial movement -Local tissue motion -Global tissue movement
3. Tissue relaxation 	Exp. Tools -2 C-ARMS -JR3 -Fiducials inside tissue -DSpace 1103 Comp. Tools -ABAQUS -MATLAB	-Force decay over time -Fiducial movement -Local tissue movement -Global tissue movement
4. Cannula + Seed applicator removal -Seed recoil -Cannula-tissue friction force 	Exp. Tools -2 C-ARMS -JR3 -Fiducials inside tissue -Seed insert -DSpace 1103 Comp. Tools -ABAQUS -MATLAB	-Force vs. disp. -Fiducial / seed movement -Local fiducial movement vs. relaxed state -Global tissue movement -Cannula and needle friction force

Dual C-arm Fluoroscopes for bead tracking: Two C-ARM Fluoroscopes were used to image the fiducial markers and the needle during insertion. C-ARM Fluoroscopy allows for real time X-ray imaging where X-rays are generated at the transmitter and photographed at the receiver. The C-ARMS were positioned so that their imaging planes were orthogonal to each other, allowing for real time imaging of the side and top views of the soft tissue fiducial markers and needle during insertion (Figure 3a). The video images of each C-ARM were captured onto a hard disk using a video capture device (Pinnacle Systems) at 30 frames per second at a resolution of 720 x 480 pixels.

Marker Registration: Once the soft tissue was in place for the experiment, a 1mm graduation radiopaque ruler (Lightek Corporation) was imaged in side view and top view to obtain the conversion for image length in pixels to length in millimeters. Using the two C-ARM configuration, we were able to correlate the beads in the top view with the beads in the side view because of the C-ARM ability to continually image as it is rotated from 90 to 180 degrees. Each bead could then be tracked as it moved in the image from the top view to the side view. This was done before inserting the needle for each soft tissue sample.

After bead registration, the needle was moved into place and inserted approximately 90 mm into the soft tissue at three different speeds, namely: 1.14, 12.7, and 25.4 mm/s. During the insertion, the JR3 force sensor captured the forces acting on the needle while the side view C-ARM and top view C-ARM continually recorded X-ray images of the needle position and the movement of beads inside the tissue (Figure 3b). After each insertion, the needle was moved to a different position in the soft tissue to minimize the chance of following a previous insertion path.

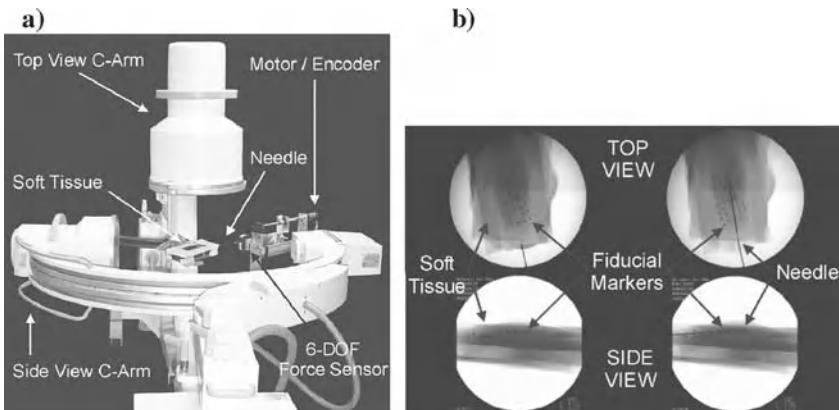


Fig. 3. a) C-ARM fluoroscope setup and b) Images acquired from both C-ARM's during needle insertion and withdrawal.

MATLAB image processing toolbox combined with standard kinematic transformations was used to extract the bead and needle coordinates in the global frame from the top and side view X-ray images. The videos from the top and side views for each insertion were loaded separately into MATLAB as .avi files. An image difference algorithm was then applied from an image of the soft tissue with no beads to the frame being analyzed. The new difference image highlighted the bead and needle movement between the frames.

3 Results and Discussion

3.1 Needle-Soft Tissue Interaction Forces

Based on needle and soft-tissue interaction during puncture events, the three graphs in Figure 4a represent insertion of the needle into 3 different soft tissue samples. Needle insertion consists of 4 events as shown in Table 1, namely: puncture, insertion, relaxation, and withdrawal. The forces acting on the needle are: the force at the tip of the needle required for cutting the tissue, the friction force of the tissue sliding along the needle shaft, and the clamping force of the tissue on the needle (Kataoka, Washio, Chinzei, Mizuhara, Simone and Okamura 2002). As the needle inserts farther into the soft tissue, it undergoes a series of micro punctures where the force rises a small amount and then drops down. Once the needle is inside the tissue, the force increases relatively linearly; with the exception of a few major puncture events along its path resulting from significant change in tissue stiffness due to its non-homogeneity. A puncture event comprises of initial deformation (leading to a rise in the force reading in the force sensor) followed by puncture (sudden drop in the force reading). The force increases linearly during insertion due to the increased surface area of the needle inside the tissue (friction force along the cannula length) and clamping force of the tissue around the needle. Based on our experimental observations, we hypothesize that in a typical puncture event the tissue is deformed at the same rate as the velocity of the needle tip. This causes a quick increase in force until puncture occurs. Based on this rationale, Figure 4b illustrates ten major puncture events for a sample of soft tissue.

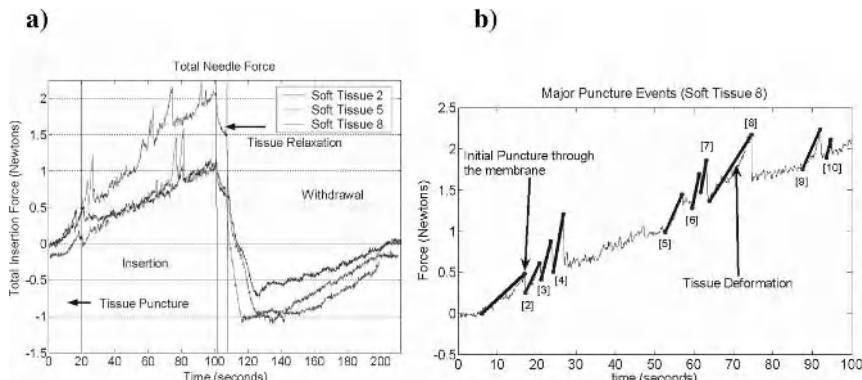


Fig. 4a) Total needle forces during insertion and withdrawal and **b)** Major puncture events

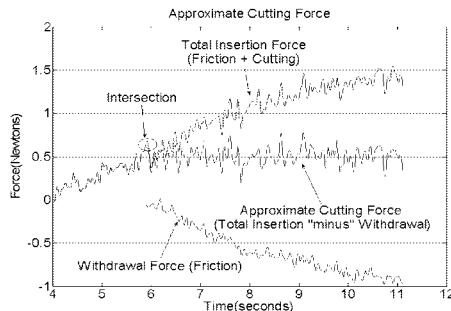


Fig. 5. Plot of the approximate cutting force during needle insertion

during the insertion part of the experiment. Figure 5 shows the approximate cutting force based on this approach.

We found that separating the cutting tip force from the friction force on the needle is a challenging experimental task. Our approach for estimating the needle-tissue interaction force purely due to cutting was to subtract the force data during the withdrawal portion of the experiment from the force data

3.2 Estimating Needle Trajectory, Bead Movement, and Tissue Relaxation

Figure 6 shows the trajectory of the needle through the soft tissue during insertion and withdrawal. Deflection of the needle tip from the straight line trajectory was observed during the experiment. Needle deflection is important to measure and predict for training radiation oncologists to place seeds accurately in the prostate. In a typical prostate

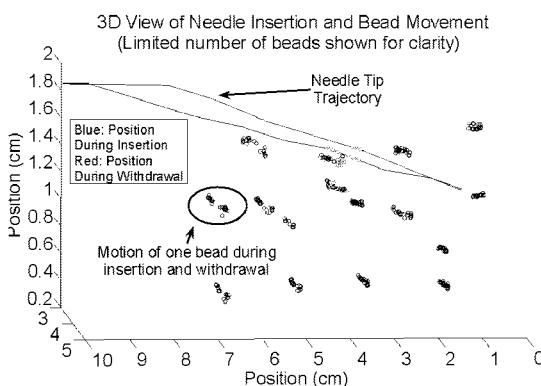


Fig. 6. 3-D bead movement during needle insertion and withdrawal

brachytherapy task the needle can deflect as much as 10 mm from the initial insertion point (Cormack, Tempany and D'Amico 2000) which requires recomputation of the dosage information for radioactive seed placement. In our studies, the needle was shown during some insertions to deflect

approximately 9.75 mm away from the straight line trajectory. Most of the deflection was caused during the initial puncture due to the force produced by tissue deformation although deflection was also caused by the

inhomogeneity of the soft tissue, generating asymmetric forces on the needle.

Figure 6 also illustrates the movement of the beads during a typical needle insertion and withdrawal in soft tissue. Each cluster in Figure 6 represents the area covered by the movement of a bead with the surrounding tissue. For clarity, we have only shown a small subset of beads actually used in the experiment. The needle was inserted at 12.7 mm/sec to a depth of approximately 9.5 cm. Each bead has a corresponding blue color for its position during insertion and red color for its position during withdrawal. Beads closest to the needle path showed the largest range of movement while the movement of beads farther away from the needle path was less. The estimated movement of the beads is used to validate a finite element model to predict soft tissue deformation during needle insertion and withdrawal task.

Tissue relaxation is a very important parameter to understand when simulating seed placement for prostate brachytherapy. Relaxation can cause a seed to be placed in an inaccurate location. Tissue relaxation can be observed by analyzing the force data as the needle is held in its full insertion position. Figure 7a illustrates the relaxation of the tissue occurring based on the force data. Figure 7b represents the top view of the tissue sample for the movement of one bead inside the tissue close to the needle path during both needle insertion and withdrawal task. Both needle insertion and withdrawal contribute to tissue relaxation, namely relaxation after the needle reaches its final position in the tissue and relaxation after the needle is completely withdrawn from the tissue. Tissue relaxation based on movement can be seen from the difference in the position of the bead after full insertion and the position of the bead after tissue relaxation has occurred while holding the needle in place.

Based on our experimental data, we can initially model the tissue relaxation process at the end of an insertion task as an exponentially decaying curve given by:

$$y = 0.5689e^{-0.7276x} + 1.5149 \quad (1)$$

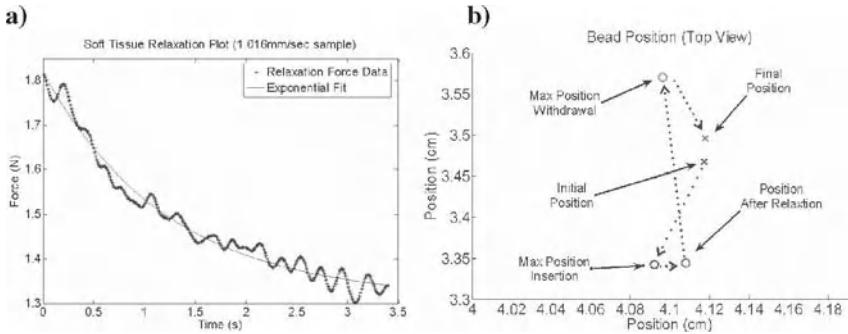


Fig. 7 a) Tissue relaxation after needle insertion and b) Bead movement during relaxation of the tissue after needle insertion and withdrawal.

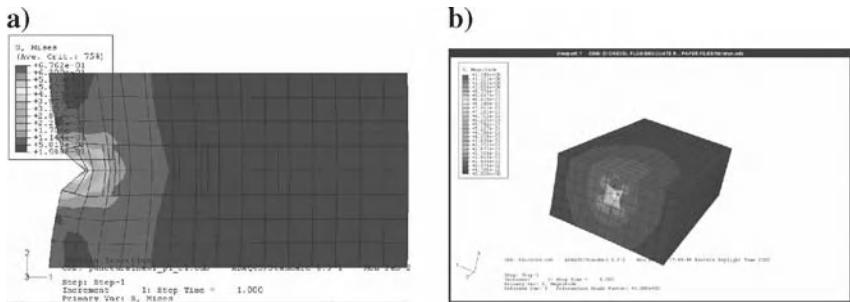


Fig. 8 a) 2D FEM model of initial puncture through the perineum and b) 3D FEM model of initial puncture.

3.3 Modeling Needle Puncture

Based on our experimental data, we have recently modeled step 1 of Table 1, namely tissue puncture. A plane stress finite element model, using 4 node quadrilateral elements was built using the ABAQUS software (Version 6.3) as shown in Figure 8a. Figure 8b shows a 3D model of tissue deformation prior to puncture.

We conducted a linear elastic FEM analysis with a Poisson's ratio of 0.3 and an initial local effective modulus of arbitrary magnitude E_1 . The tip of the needle was modeled as a node located at points corresponding to the position of the puncture events shown in Figure 6. For each event, the node was given an experimentally measured displacement ΔU^{exp} . The computed ΔF^{FEM} from the displacement at that node was compared with the ΔF^{EXP} measured experimentally. Using the initial effective modulus E_1 , we

performed iterations to obtain the ΔF^{FEM} equal to ΔF^{EXP} . The final E value determined is the local effect modulus, $E^{\text{effective}}$, of the tissue during the puncture event (see Table 2). Figure 9 illustrates this computation procedure.

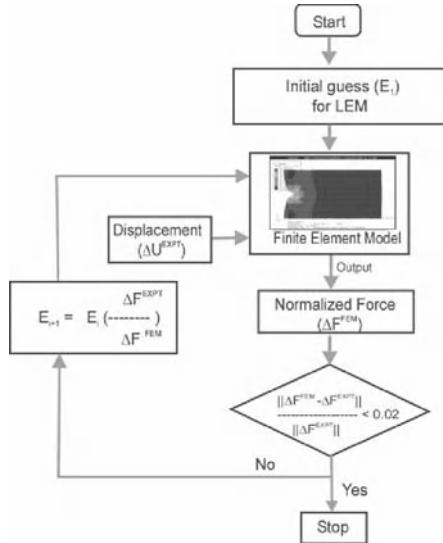


Table 2. Local Effective Modulus of Puncture Events

Puncture Event	LEM*10 ³ [N/m ²]
Initial Puncture	9.575
[2]	13.975
[3]	23.833
[4]	34.263
[5]	11.798
[6]	23.325
[7]	30.459
[8]	8.291
[9]	11.541
[10]	21.422

Fig. 9. Flow chart for determination of the local effective modulus (LEM).

The 3D FEM model has been developed and will also be used to predict soft tissue movement and forces under needle insertion and will be verified with that of experimental data. Figure 8b is an initial three dimensional model with a global mesh of $0.5 \times 0.5 \times 0.5 \text{ cm}^3$, 8-node, solid linear brick element with incompatible modes. Nodes of elements closest to fiducial marker locations are placed at the coordinates of the fiducial markers. The tip of the needle was modeled as a node similar to that of the two dimensional model.

4 Conclusion and Future Work

This study demonstrates a unique approach for estimating needle and soft tissue interaction for the simulation of accurate seed placement in prostate brachytherapy. We have shown through the use of 2 C-ARM Fluoroscopes that we can obtain in real time 3D needle trajectory and internal global and local tissue deformation during needle insertion into soft

tissue. From this we can extract important parameters for modeling tissue puncture and tissue relaxation. The internal tissue movement can also be used to verify predictions of soft tissue finite element models. Also, we have shown by using the force and displacement data from puncture events that we can quantify the local resistance of the soft tissue to puncture, through the computation of the LEM. Additionally, by subtracting the friction and clamping force during needle withdrawal from the total needle force during insertion, we can obtain the tissue cutting force during needle insertion.

We propose to develop a three dimensional finite element model for simulating the needle insertion and withdrawal task in prostate brachytherapy. Based on 3D fiducial movement estimated from our current work, we can compute the “strain field” for each fiducial in the image and determine the local effective modulus of the tissue during a needle insertion and withdrawal task.

The work presented in this paper is to our knowledge the first of its kind for modeling needle deflection and soft tissue movement during needle insertion and withdrawal task in prostate brachytherapy. This work will be the basis for developing a reality-based training simulator for training radiation oncologists in prostate brachytherapy.

Acknowledgements. We would like to thank Dr. Waqus Anjum and Sajeel Shiromani for their valuable help in conducting dual C-ARM experiments.

References

- R. Alterovitz, R. Pouliot, R. Taschereau, I. Hsu and K. Goldberg (2003) Simulating Needle Insertion and Radioactive Seed Implantation for Prostate Brachytherapy. Medicine Meets Virtual Reality 11, 19-25
- P. N. Brett, A. J. Harrison and T. A. Thomas (2000) Schemes for the Identification of Tissue Types and Boundaries at the Tool Point for Surgical Needles.IEEE Transactions on Information Technology in Biomedicine, 30-36
- R. A. Cormack, C. M. Tempny and A. V. D'Amico (2000) Optimizing Target Coverage by Dosimetric Feedback During Prostate Brachytherapy.International Journal of Radiation Oncology Biology and Physics, 48 4: 1245-1249

- S. P. DiMaio and S. Salcudean (2002) Needle insertion modeling and simulation. IEEE International Conference on Robotics and Automation, 2098-2105
- C. Forest, H. Delingette and N. Ayache (2002) Cutting Simulation of Manifold Volumetric Meshes. Proceedings of the Fifth International Conference on Medical Image Computing and Computer Assisted Intervention, 2 235-244
- J. D. Hall (2003) Why patients choose prostatectomy or brachytherapy for localized prostate cancer: results of a descriptive survey. Urology, 61 402-407
- J. Hong, T. Dohi, M. Hashizume, K. Konishi and N. Hata (2004) An Ultrasound-driven needle-insertion robot for percutaneous cholecystostomy. Physics in Medicine and Biology, 49 441-455
- A. Jemal (2004) Cancer Statistics. CA Cancer J. Clin, 54 8-29
- H. Kataoka, T. Washio, K. Chinzei, K. Mizuhara, C. Simone and A. Okamura (2002) Measurement of Tip and Friction Force Acting on a Needle During Penetration. Proceedings of the Fifth International Conference on Medical Image Computing and Computer Assisted Intervention, 216-223
- S. E. Langley and R. W. Laing (2004) Iodine seed prostate brachytherapy: an alternative first-line choice for early prostate cancer. Prostate Cancer Prostatic. Dis., 7 201-207
- B. Leak (2002) Relavent patient and tumor considerations for early prostate cancer treatment. 39-44
- J. Magill, B. Anderson, G. Anderson, P. Hess and S. Pratt (2004) Multi-Axis Mechanical Simulator for Epidural Needle Insertion. International Symposium on Medical Simulation,
- H. Nienhuys and F. van der Stappen (2003) Interactive needle insertions in 3D nonlinear material.
- M. D. O'Leary, C. Simone, T. Washio, K. Yoshinaka and A. M. Okamura (2003) Robotic Needle Insertion: Effects of Friction and Needle Geometry. IEEE International Conference on Robotics and Automation,
- G. Picinbono, H. Delingette and N. Ayache (2001) Nonlinear and anisotropic elastic soft tissue models for medical simulation. IEEE International Conference on Robotics and Automation, 1370-1375
- L. Potters (2003) Permanent prostate brachytherapy in men with clinically localized prostate cancer. Clinical Oncology, 15 301-315
- J. Sharkey (2000) Minimally invasive treatment for localized adenocarcinoma of the prostate: review of 1048 patients treated with ultrasound-guided palladium-103 brachytherapy. Journal of Endourology, 14 343-350
- C. Simone and A. Okamura (2002) Modeling of Needle Insertion Forces for Robot-Assisted Percutaneous Therapy. IEEE International Conference on Robotics and Automation, 2085-2091

- W. L. Smith, K. Surry, G. Mills, D. Downy and A. Fenster (2001) Three Dimensional Ultrasound-Guided Core Needle Breast Biopsy. *Ultrasound in Medicine and Biology*, 27 8: 1025-1034
- D. Stoianovici, L. Whitcomb, J. Anderson, R. Taylor and L. Kavoussi (1998) A Modular Surgical Robotic System for Image Guided Percutaneous Procedures. *MICCAI*, 404-410

Planning

Haptic Virtual Fixtures for Robot-Assisted Manipulation

Jake J. Abbott*, Panadda Marayong, and Allison M. Okamura†

Department of Mechanical Engineering, The Johns Hopkins University
Baltimore, Maryland, 21218 USA
jabott@ethz.ch, pmarayong@jhu.edu, aokamura@jhu.edu

Summary. Haptic virtual fixtures are software-generated force and position signals applied to human operators in order to improve the safety, accuracy, and speed of robot-assisted manipulation tasks. Virtual fixtures are effective and intuitive because they capitalize on both the accuracy of robotic systems and the intelligence of human operators. In this paper, we discuss the design, analysis, and implementation of two categories of virtual fixtures: guidance virtual fixtures, which assist the user in moving the manipulator along desired paths or surfaces in the workspace, and forbidden-region virtual fixtures, which prevent the manipulator from entering into forbidden regions of the workspace. Virtual fixtures are analyzed in the context of both cooperative manipulation and telemanipulation systems, considering issues related to stability, passivity, human modeling, and applications.

1 Introduction

Haptic virtual fixtures are software-generated force and position signals applied to human operators via robotic devices. Virtual fixtures help humans perform robot-assisted manipulation tasks by limiting movement into restricted regions and/or influencing movement along desired paths. By capitalizing on the accuracy of robotic systems, while maintaining a degree of operator control, human-machine systems with virtual fixtures can achieve safer and faster operation. To visualize the benefits of virtual fixtures, consider a common physical fixture: a ruler. A straight line drawn by a human with the help of a ruler is drawn faster and straighter than a line drawn free-hand. Similarly, a robot can apply forces or positions to a human operator to help him or her draw a straight line. However, a robot (or haptic device) has the additional ability to provide assistance of varying type, level, and geometry.

* Jake J. Abbott is now with the Institute of Robotics and Intelligent Systems, ETH Zurich, Switzerland.

† This work is supported by National Science Foundation grants #ITR-0205318 and #IIS-0347464.

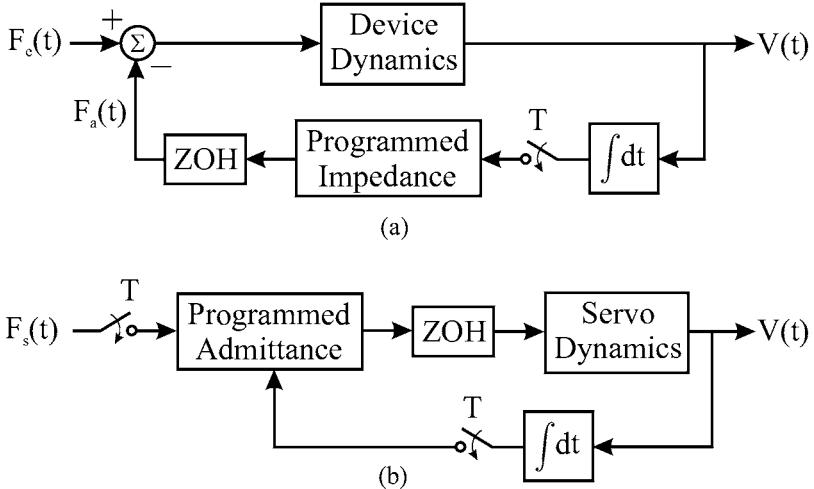


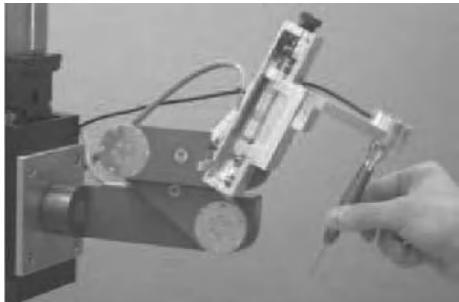
Fig. 1. Models of robots of the (a) impedance and (b) admittance types. For the impedance-type robot, F_a is the actuator force and F_e is the sum of all externally applied forces. For the admittance-type robot, F_s is the component of the externally applied force that is sensed. V is the robot velocity, and T is sampling period of the control system.

Virtual fixtures show great promise for tasks that require better-than-human levels of accuracy and precision, but also require the intelligence provided by a human directly in the control loop. Human-machine manipulation systems make up for many of the shortcomings of autonomous robots (e.g., limitations in artificial intelligence, sensor-data interpretation, and environment modeling), but the performance of such systems is still fundamentally constrained by human capabilities. Virtual fixtures, on the other hand, provide an excellent balance between autonomy and direct human control. Virtual fixtures can act as safety constraints by keeping the manipulator from entering into potentially dangerous regions of the workspace, or as macros that assist a human user in carrying out a structured task. Applications for virtual fixtures include robot-assisted surgery, difficult assembly tasks, and inspection and manipulation tasks in dangerous environments.

Virtual fixtures can be applied to two types of human-machine robotic manipulation systems: *cooperative manipulators* and *telemanipulators*. In cooperative manipulation, the human uses a robotic device to directly manipulate an environment. In telemanipulation, a human operator manipulates a master robotic device, and a remote slave robot manipulates an environment while following the commands of the master. In general, the robots used in these systems can be of the *impedance* or the *admittance* type [6]; basic models for these robot types are shown in Fig. 1.

Robots of the impedance type, such as typical haptic devices, are backdrivable with low friction and inertia, and have force-source actuators. An example of an impedance-type robot familiar to many is the PHANTOM® from SensAble Technologies, Inc. [32]. Robots of the admittance type, such as typical industrial robots, are modeled as being nonbackdrivable with velocity-source actuators. This is due to either large inertia and joint friction from gear reduction in electric-motor systems, or valves and incompressible fluid in hydraulic systems. The velocity is controlled with a high-bandwidth low-level controller, and is assumed to be independent of applied external forces. This model loses validity when the admittance-type robot interacts with a very stiff environment.

Figure 2(a) shows the Johns Hopkins University Steady-Hand Robot [33], an admittance-type cooperative manipulator designed for microsurgical procedures. Figure 2(b) shows the da Vinci® Surgical System (Intuitive Surgical, Inc.) [12, 13], an impedance-type telemanipulator designed for minimally invasive surgical procedures. The virtual fixtures created and studied in our lab are designed explicitly for systems such as these.



(a)



(b)

Fig. 2. (a) The Johns Hopkins University Steady-Hand Robot [33]. (b) The da Vinci® Surgical System [12, 13] (image used with the permission of Intuitive Surgical, Inc.).

2 Prior Work on Virtual Fixtures

“Virtual fixtures” [1, 17, 26–29] (also appearing under the name of “synthetic fixtures” [31], “virtual mechanisms” [15, 24], “virtual tools” [14], “virtual paths and surfaces” [25], and “haptically augmented teleoperation” [34]) have been applied to robotic manipulators using a variety of methods, though they can generally be categorized as either *guidance virtual fixtures* or *forbidden-region*



Fig. 3. (a) Guidance virtual fixtures assist in guiding the robot along desired paths. (b) Forbidden-region virtual fixtures help keep the robot out of forbidden regions.

virtual fixtures. As their name implies, guidance virtual fixtures (GVFs) help keep the manipulator on desired paths or surfaces. Alternatively, forbidden-region virtual fixtures (FRVFs) [28] help keep the manipulator out of forbidden regions of the workspace. These virtual fixture types are illustrated in Fig. 3.

The majority of prior work on virtual fixtures has been applied to telemanipulation. Rosenberg [29] implemented FRVFs as impedance surfaces on the master device to assist in peg-in-hole tasks. Joly et al. [15] introduced a proxy-based [36] GVF method where the proxy is constrained to move on the virtual fixture, and the master and slave both servo to the proxy position and affect its movement along the virtual fixture. Micaelli et al. [24] extended this method to allow for multiple proxies, each on its own virtual fixture and with its own dynamics. Itoh et al. [14] developed a task-assistance tool that connects admittance-type robots to virtual fixtures with impedance control methods. Park et al. [26] implemented FRVFs on the remote slave by rejecting master commands into the forbidden region. In their method, the slave manipulator servos to a proxy, and the proxy follows the master when outside the FRVF, but will not follow the master into the forbidden region. Turro et al. [34] implemented GVFs on a system with an impedance-type master and admittance-type slave. The master is bound to a proxy, which is constrained to move on the virtual fixture, and the slave then tracks either the master or the proxy, depending on the desired level of user control. Payandeh and Stanisic [28] implemented virtual fixtures on both the master and slave manipulators, using a variety of geometries, to help guide the remote manipulator in a predetermined task. Kuang et al. [17] then applied this research to difficult assembly tasks. The virtual fixtures above were implemented with penalty-based or potential-field methods. These are impedance-type virtual fixtures that act in an active way, in that stored potential energy in the virtual fixture may potentially be released in an undesirable fashion.

Virtual fixtures have also been implemented on passive cooperative manipulation systems known as Cobots [25]. Park et al. [27] extended these methods to telemanipulation systems where the master device is a Cobot, for assistance in nuclear deactivation and decommissioning tasks. These virtual fixtures act in a passive way in the sense that the virtual fixtures are only able to restrict, and not generate, motion. These so-called passive virtual fixtures work much like methods developed for autonomous robots, such as “passive velocity field control” [21]. It is also possible to implement passive virtual fixtures

using admittance-type systems. Since these nonbackdrivable robots move in a highly-controlled fashion, one can passively restrict movement in any given direction by simply not commanding any movement in that direction. This type of virtual fixture has been implemented on the Johns Hopkins University Steady-Hand Robot [33] by Bettini et al. [7] and Li et al. [18]. In [18], an optimization-based approach is used to construct motion constraints from known task geometries and instantaneous robot kinematics that can be applied independent of the manipulator type (cooperative manipulation or telemanipulation, admittance or impedance type). Research on this type of virtual fixture has also been recently been extended to admittance-type telemanipulators by Aarno et al. [1].

Prior work on virtual fixtures has been largely ad hoc, with significant reliance on particular applications. Thus, in this paper, we attempt to unify the past and present research in the field by considering the design, analysis, and application of virtual fixtures to various system types. In Sections 3 and 4, we discuss how guidance virtual fixtures and forbidden-region virtual fixtures, respectively, can be used for task assistance in both cooperative manipulation and telemanipulation. Then, in Section 5, we discuss in detail the issues involved with safe and functional implementation of virtual fixtures. Finally, in Section 6, we present a set of interesting topics for future work in this field of research.

3 Guidance Virtual Fixtures

Guidance virtual fixtures (GVFs) assist the user in moving the robot manipulator along desired paths or surfaces in the workspace. GVF_s can be of either the impedance or admittance type [6]. Impedance-type GVF_s act as potential fields, actively influencing the movement of the robotic manipulator. These impedance methods can lead to unexpected and undesirable movements of the manipulator, so we have chosen to focus on GVF_s of the admittance type.

Admittance control typically takes the form $\mathbf{v} = K_a \mathbf{f}$, where \mathbf{f} is the user's applied force vector, K_a is an admittance gain matrix, and \mathbf{v} is the output velocity vector. This control scheme is sometimes referred to as proportional-velocity control. Admittance control has the desirable property that the velocity of the manipulator is proportional to the applied force, so the manipulator does not move if the user does not apply a force. In addition, slow robot movement is achieved with a soft touch. Admittance-type GVF_s are very natural with admittance-type cooperative systems, but can also be implemented on impedance-type telemanipulation systems with a novel Pseudo-admittance control law [2, 4].

3.1 GVF_s for Cooperative Manipulation

In an admittance-type cooperative manipulation system, the robot motion is proportional to the user's applied force, which is measured by a force sensor.

To create GVF_s, an instantaneous preferred direction is defined based on the position of the robot relative to the desired path or surface. The applied force is then decomposed into components in the preferred direction and in other, non-preferred directions. By eliminating the commanded motion due to the applied force in the non-preferred directions, we create a passive guidance along the preferred direction. Implementing GVF_s in this fashion essentially makes the admittance gain matrix K_a both state and input dependent. Details of this GVF method can be found in [7].

Varying the response to the non-preferred force component creates different levels of guidance. *Hard* guidance refers to GVF_s where none or almost none of the non-preferred force component is permitted, leaving the user with no or little freedom to deviate from the preferred path. Alternatively, *soft* GVF_s give the user the freedom to move away from the path by allowing some motion in the non-preferred directions. We conducted an experiment with the JHU Steady-Hand Robot to evaluate the effect of GVF admittance on user performance, including accuracy and execution time [23]. Three tasks (Path Following, Off-path Targeting, and Avoidance) were selected to represent a broader class of motions that can occur in a real task execution. GVF_s were used with varying admittance to keep the user on the preferred path, in this case a sine curve on a horizontal plane.

Figure 4 shows the robot trajectories during the Off-path Targeting and Avoidance tasks, with three levels of guidance. In the Targeting task, the users were instructed to reach the target located on the perimeter of the circle outlined in gray. In the Avoidance task, the users avoided the area by trying to follow along the circle perimeter. Robot trajectories in the Path Following task were similar to the portions seen outside the circular area in the two off-path tasks shown in Fig. 4. In the Path Following task, the users performed

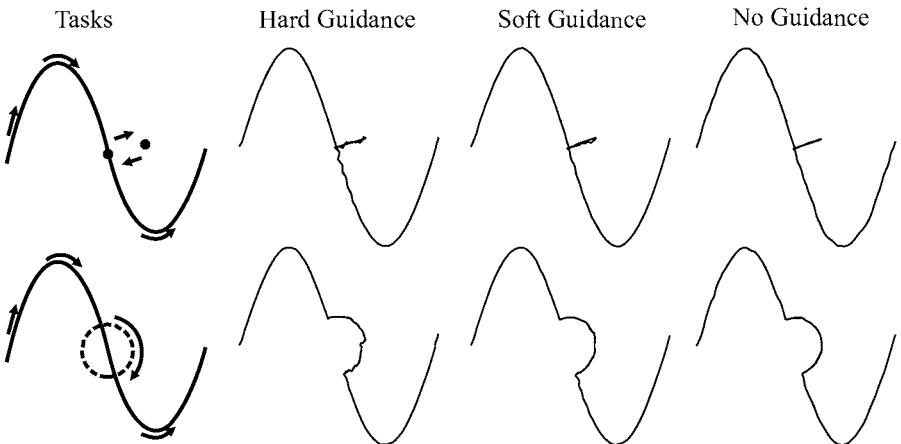


Fig. 4. Robot trajectories in the Targeting task (top) and the Avoidance task (bottom) with JHU Steady-Hand Robot.

the task more accurately (with statistical significance) with GVF_s, though not significantly faster. In the off-path tasks, the users had to fight against the GVF guidance to complete the desired motion. This represents situations where the virtual fixture is incorrectly placed and the user wishes to override the guidance. As expected, users take significantly longer to perform off-path tasks with increased guidance. Error also increases slightly. The experiment shows that GVF_s can improve both time and accuracy simultaneously, while still allowing some independent user motion. More detailed descriptions of the experiment and the results can be found in [23]. GVF implementation for tasks in 3-D were also explored in Dewan et al. [9], where the tool was guided along a user-defined desired surface. In this experiment, stereo cameras were used to reconstruct the workspace and track the tool position and orientation.

3.2 GVF_s for Telemanipulation

In telemanipulation, good position correspondence between the master and slave robots is desirable to create a sense of telepresence for the user. However, it is actually the slave manipulator that we wish to guide using GVF_s, and master movements in its corresponding workspace are somewhat less important.

The GVF_s developed for admittance-type cooperative manipulators could trivially be extended to telemanipulation systems where both the master and slave are of the admittance type. However, unlike cooperative manipulation systems, telemanipulation systems are typically designed as impedance-type systems (that is, the master is an impedance-type haptic device, while the slave manipulator can be of either the impedance or admittance type). For these systems, we do not control the velocity of the system directly (due to force-source actuation), so we cannot implement admittance control directly. We have developed a novel telemanipulation control algorithm called Pseudo-admittance control [2, 4] that mimics admittance control on impedance-type telemanipulators, and extends the GVF_s described in Section 3.1 and [7] to telemanipulation. Pseudo-admittance makes use of a proxy [36], which exists only in software, that can be commanded to move under admittance control.

Under Pseudo-admittance control, the master servos to the slave position, while the slave servos to the proxy position, as illustrated in Fig. 5. The proxy moves under admittance control, using the force of the master’s servo controller as its input force. GVF_s are then implemented by attenuating the commanded velocity in non-preferred directions, as described in Section 3.1. Figure 5 shows the experimental results from two PHANToM® robots [32] configured for Pseudo-admittance control. Using different levels of guidance (i.e., modifying the calculation of the preferred direction and the attenuation of velocities in the non-preferred directions), the slave is guided to a preferred plane in the workspace, but the user retains ultimate control to move the slave anywhere in the workspace.

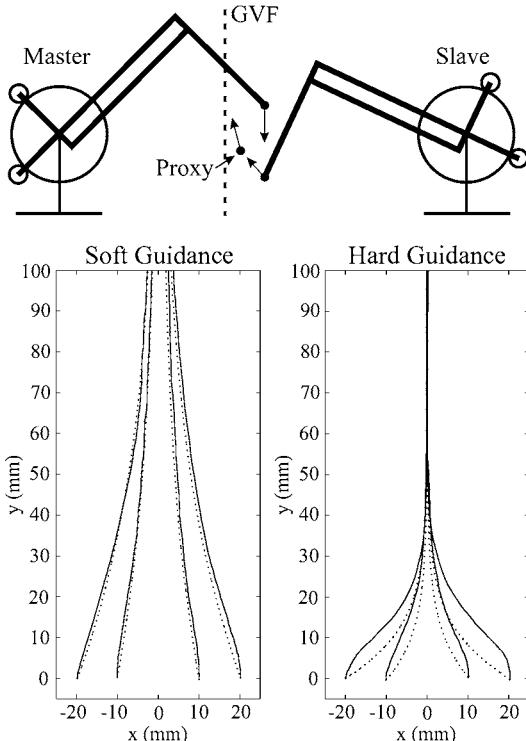


Fig. 5. Guidance virtual fixtures implemented on two PHANToM® robots (**top**) configured for Pseudo-admittance Bilateral Telemanipulation [2, 4]. The robots are shown superimposed on the same workspace to aid in visualization. Experimental data (**bottom**), with master (—) and proxy (···) trajectories, are shown for two levels of guidance. The slave servos to the proxy. The GVF is on the plane $x = 0$. The user applies a force approximately in the positive y direction, and the manipulator is guided by the GVF.

4 Forbidden-Region Virtual Fixtures

Forbidden-region virtual fixtures (FRVFs) prevent the robot manipulator from entering into forbidden regions of the workspace. They have an on/off nature, such that they have no effect on the robot when it is outside of the forbidden region. As with GVF, FRVFs can be of either the impedance or admittance type. Impedance-type FRVFs take the form of “virtual walls,” which are commonly employed and studied for haptic virtual environments, and are typically implemented as simple spring-damper surfaces. These are penalty-based methods, so the force generated by the FRVF is proportional to the manipulator’s penetration of the FRVF (i.e., some penetration is necessary to engage the FRVF). Admittance-type FRVFs are simply implemented by not commanding any manipulator motion into the forbidden region.

4.1 FRVFs for Cooperative Manipulation

FRVFs can be viewed as a subclass of GVF for an admittance-controlled cooperative manipulator. The FRVFs are trivial to implement, by simply eliminating any commanded motion into the forbidden region. Inherently, the forbidden region is the non-preferred direction defined in the GVFs.

Examples of FRVFs in cooperative systems are highlighted in [9] and [20]. In Dewan et al. [9], the virtual fixtures constrained the user to move along the shortest path between the current tool position and a predefined target on the surface. The robot admittance gain was turned to zero once the target was reached. Li and Taylor [20] combined both GVF_s and FRVFs in creating anatomy-based motion constraints for a path-following task in a constrained workspace. The algorithm uses the robot kinematics, the user's force input, and a 3-D geometric model of the workspace to generate virtual fixtures and an optimal set of joint displacements to guide the tool tip along a path while preventing the tool shaft from entering into forbidden regions.

The user may want the option to intentionally move past the FRVF if it is deemed necessary. The GVFs implemented in Section 3.1 left the user with ultimate control to move the manipulator away from the desired path, but it is not clear if it makes sense to create admittance-type FRVFs that allow some motion into the forbidden region. In one sense, an admittance-type FRVF that acts in this way is not a FRVF at all. It may be possible though, through state-and-input-dependent adaptation of the admittance-gain matrix, to implement FRVFs that allow some penetration into the forbidden region while retaining their functional purpose.

4.2 FRVFs for Telemanipulation

As with the GVFs of Section 3.2, in telemanipulation we are only really concerned with penetration of the slave manipulator into the forbidden region. Penetration of the master device into the corresponding region of its workspace is somewhat inconsequential.

Impedance-type FRVFs can be implemented on telemanipulators by overlaying a penalty-based virtual wall on the existing telemanipulation controller. It is possible to implement the virtual wall on either the master or the slave side (or both simultaneously). Both have the effect of reducing movement of the slave into the forbidden region. However, each presents a different haptic experience for the user, depending on the underlying telemanipulation controller, and each provides different levels of disturbance rejection, depending on the location of the disturbance. In [2], we found that slave-side FRVFs are most effective at rejecting disturbances on the slave, while maintaining a sense of telepresence for the user (i.e., minimizing position error between the master and the slave). However, we found that master-side FRVFs are most effective at rejecting (un)intentional user commands into the forbidden region, while maintaining a sense of telepresence. The choice of FRVF architecture is likely to be task dependent.

It is also possible to implement admittance-type FRVFs through the use of a proxy. If the slave manipulator servos to a proxy, rather than directly servoing to the master, then we can influence slave movement in forbidden regions by adapting the dynamic properties of the proxy. When the master is not interacting with the FRVF, the proxy is made to follow the master exactly.

When the master moves beyond the FRVF, we attenuate the movement of the proxy past the FRVF (including removing the penetration completely).

Both types of FRVF act by attenuating slave movement into the forbidden region, while allowing the user to move the slave into the forbidden region if desired. The amount of attenuation, and consequently user control, is governed by system gains. Admittance-type FRVFs implemented on admittance-type slaves can be made to be infinitely stiff. The stiffness of an admittance-type FRVF with an impedance-type slave is limited by the stability of the virtual coupling between the slave and the proxy [6]; however, this FRVF can still be made to appear infinitely stiff to the user commands. The performance of an impedance-type FRVF is also ultimately limited by stability constraints. The stability of impedance-type FRVFs, under stability and passivity considerations, is explored in detail in [2].

5 Virtual Fixture Design Considerations

Prior work in virtual fixtures has focused primarily on application-specific virtual-fixture geometries and user performance of specific tasks. This section highlights a number of additional design considerations that are important for progress in this field; researchers have only recently begun to examine these issues.

One fundamental design problem is to determine the best type of underlying system for a virtual-fixture application. Cooperative manipulation systems are intuitive to use, due to the natural hand-eye coordination that comes from directly manipulating the tool. The sense of telepresence felt with a telemanipulator is limited by the position error in the system, as well as the quality of the visual and haptic feedback provided to the user. Admittance-type cooperative systems also have desirable “steady-hand” properties; the user’s hand is literally steadied by holding onto the rigid, slow-moving robot. This behavior must be mimicked on an impedance-type telemanipulator; the slave manipulator can be controlled to move slowly, but a backdrivable master device is not as capable of steadyng the hand of the user. However, telemanipulators provide not only the ability to manipulate distant environments, but also the ability to provide scaling in both position and force. Force scaling is also possible with cooperative manipulation [30], although an additional force sensor or accurate environmental model is needed to obtain the contact force. It is important, in general, to consider whether force sensing is necessary and practical in terms of size, cost, and environment compatibility.

System performance also depends on the accuracy of the task geometry definition. For example, a computer vision system can be used to reconstruct the workspace and define the geometry of the virtual fixtures. The accuracy of the virtual fixtures defined depends on the resolution of the vision system, calibrations, and the accuracy of the tracking algorithm, which can be sensitive to changing light conditions and occlusions. The designer of a virtual fixture

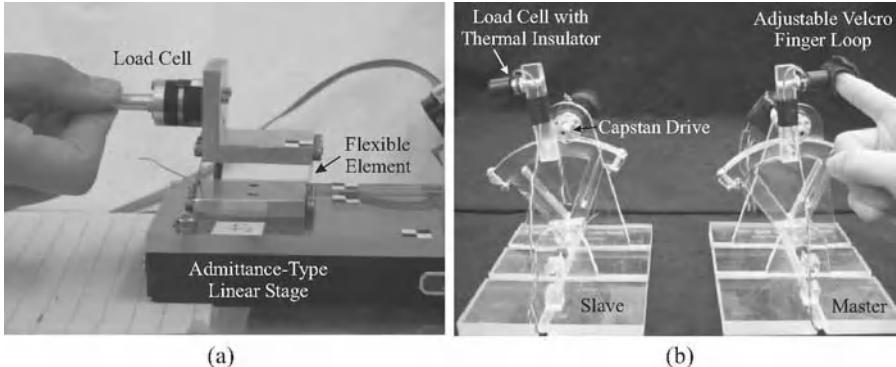


Fig. 6. 1-DOF experimental systems. (a) Admittance-type cooperative manipulator for the study of the effect of link compliance on virtual-fixture performance. (b) Impedance-type telemanipulator for the study of FRVF stability.

must be able to predict the sensitivity of system performance to inaccuracies in virtual-fixture geometry definition and develop mechanisms to correct for errors. It may be necessary to build in enough user control to compensate for inaccuracies in the virtual-fixture geometry, as was discussed in Section 3.1.

In cooperative systems, unmodeled robot dynamics, such as joint and link compliance, can introduce significant tool positioning error, especially for micro-scale tasks. Joint and link flexibility add unactuated degrees of freedom to the robot. A human actively and directly manipulating the tool exacerbates the difficulty of error correction. A hand dynamic model could be added to better predict the system response near a virtual fixture, and adjust the controller appropriately to compensate for the error. This issue is being investigated on a 1-DOF admittance-type system (Fig. 6(a)) where the FRVF was implemented as a virtual wall. Joint compliance was simulated with a physical spring added between the tool and the stage. Two methods were proposed to create a dynamic virtual fixture, with its location determined based on the system dynamics, that prevents the user from entering the true forbidden region. The experimental results shown in Fig. 7 indicate that accounting for both the dynamic properties of the hand and the effects of robot momentum are effective in preventing FRVF penetration. The description of the methods and the complete experimental results can be found in [22].

Another major concern in the design of virtual fixtures for impedance-type telemanipulators is stability. Because of their backdrivable force-source actuators, these systems are prone to instability if the control-system gains are too high. This makes stable and effective virtual fixtures conflicting goals. We have investigated the stability of FRVFs, considering effects of friction, sampling, and quantization, using both equilibrium stability analysis [2, 5] and passivity analysis [3]. We used a 1-DOF system, shown in Fig. 6(b), for this purpose. It is possible to design a FRVF to be passive, with the

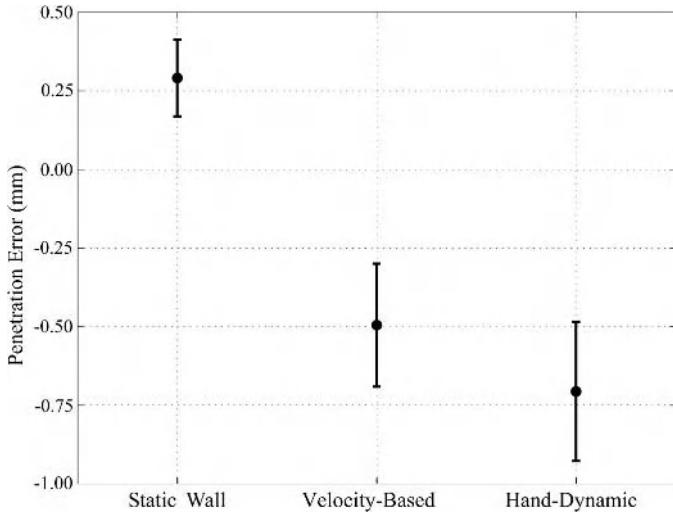


Fig. 7. Experimental results indicating that the effects of robot compliance on FRVF functionality can be mitigated through dynamic modeling of the robot and the human hand [22]. Mean values of the penetration into the forbidden region and standard deviation bars collected from eight users are shown. Negative error indicates no penetration into the forbidden region.

additional assumption of human passivity being sufficient for system stability. However, as shown in Fig. 8, we found that including an explicit model of potential human users can lead to stability predictions that are significantly less conservative than simply requiring passivity of the FRVF. The description of the methods and the complete experimental results can be found in [2, 5].

It is tempting to model the human user as an exogenous input to the system, for the purpose of stability analysis, but in general, the dynamics of the human user are part of the closed-loop feedback system. However, it is also reasonable to assume that for certain slow-moving systems, the human user is essentially unaffected by the movement of the system. An initial study in our lab shows that, for an admittance-type cooperative manipulator, it is the velocity of the robot, and not the admittance gain, that directly affects human force control precision [35]. Thus, by restricting the velocity of the manipulator, it may be possible to consider the human user as an exogenous input, greatly simplifying system stability analysis. More research is needed to better understand the role of the human user in the total system response.

As illustrated above, it is not always obvious when dynamic modeling of the human user is necessary or desirable in virtual-fixture design and analysis. Most of the prior work on virtual fixtures has excluded modeling of the human user. In addition to mechanical modeling, experimental results of GVF in cooperative systems suggest that human intent and psychophysics may also affect GVF performance. Selecting an appropriate level of guidance

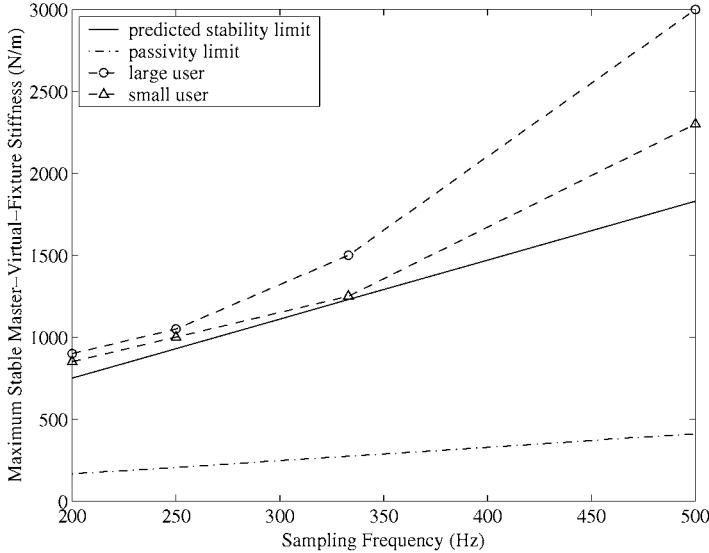


Fig. 8. Experimental stability limits on master FRVF stiffness on a unilateral telemanipulator, for large- and small-handed malicious users, compared to predicted stability limits based on models of the “worst-case” user [2, 5]. Passivity of the FRVF based on [8] is also shown. For each data point, the users found the stiffest virtual fixture for which instability could not be experimentally generated.

is required for optimal performance, and the selection is task dependent. Having a high level of guidance increases error and time for tasks that require off-path motions, though it significantly improves both time and error during path-following. An optimal GVF selection was explored in [23]. Artificial intelligence can also be added to adjust the GVF based on the user’s intent. For example, Li and Okamura [19] and used Hidden Markov Models to recognize user motions and provide appropriate GVF assistance in a combined curve-following and object-avoidance task in cooperative manipulation. Aarno et al. [1] took a similar approach with telemanipulation. Kragic et al. [16] broke a complex microsurgical task into subtasks, each of which benefited from different types of virtual-fixture assistance.

6 Summary and Future Work

This paper described methods for design and implementation of haptic virtual fixtures on a number of different underlying platforms. Through analysis and experiments, we show that virtual fixtures can improve human-machine performance, while allowing the user to maintain ultimate control over the task execution.

There are a number of critical questions that provide important topics for future research in this field. For example, what is the best virtual-fixture geometry for a given task? How does the human user interpret the combination of haptic cues coming from the manipulated environment and the virtual fixture? Does this lead to haptic confusion, affecting the user's sense of immersion in the task? If the virtual-fixture geometry and/or gains vary in time, not only could it lead to confusion on the part of the user, but it also complicates stability analysis. Can virtual fixtures be used as training devices for complicated tasks, and then eventually be removed, much like training wheels on a bicycle [10, 11]? To what extent does the human need to be included in the analysis of these systems? It is desirable to say as much as possible about the robotic system itself, without needing to consider human dynamics. Is it possible to apply what we have learned thus far to the design of force virtual fixtures, which assist the user in applying the proper force to the manipulated environment?

It is important that we generalize the research in this field across systems and tasks, so that knowledge gained in individual research efforts can advance the field as a whole. Virtual fixtures will no doubt facilitate robot-assisted tasks that were previously impossible, but this nascent field is still rich with interesting research topics that must be explored before human-machine systems can capitalize on the full benefit of virtual fixtures.

Acknowledgements

The authors would like to thank all those researchers at Johns Hopkins University who have contributed to the work presented in this paper, namely, Dr. Gregory Hager, Dr. Russell Taylor, Dr. Ming Li, and Maneesh Dewan.

References

1. D. Aarno, S. Ekvall, and D. Kragić. Adaptive virtual fixtures for machine-assisted teleoperation tasks. In *Proc. IEEE Int'l. Conf. on Robotics and Automation*, pages 1151–1156, 2005.
2. J. J. Abbott. *Virtual Fixtures for Bilateral Telemanipulation*. PhD thesis, Department of Mechanical Engineering, The Johns Hopkins University, 2005.
3. J. J. Abbott and A. M. Okamura. Effects of position quantization and sampling rate on virtual-wall passivity. *IEEE Trans. Robotics*, 21(5):952–964, 2005.
4. J. J. Abbott and A. M. Okamura. Pseudo-admittance bilateral telemanipulation with guidance virtual fixtures. In *Proc. Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2006.
5. J. J. Abbott and A. M. Okamura. Stable forbidden-region virtual fixtures for bilateral telemanipulation. *ASME J. Dynamic Systems, Measurement, and Control*, In Press.

6. R. J. Adams and B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Trans. Robotics and Automation*, 15(3):465–474, 1999.
7. A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hager. Vision-assisted control for manipulation using virtual fixtures. *IEEE Trans. Robotics*, 20(6):953–966, 2004.
8. J. E. Colgate and G. G. Schenkel. Passivity of a class of sampled-data systems: Application to haptic interfaces. *J. Robotic Systems*, 14(1):37–47, 1997.
9. M. Dewan, P. Marayong, A. M. Okamura, and G. D. Hager. Vision-based assistance for ophthalmic micro-surgery. In *Proc. Int'l. Conf. on Medical Image Computing and Computer-Assisted Intervention*, pages 49–57, 2004.
10. D. Feygin, M. Keehner, and F. Tendick. Haptic guidance: Experimental evaluation of a haptic training method for a perceptual motor skill. In *Proc. Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, pages 40–47, 2002.
11. R. B. Gillespie, M. S. O'Modhrain, P. Tang, D. Zaretzky, and C. Pham. The virtual teacher. In *Proc. ASME Int'l. Mechanical Engineering Congress and Exposition*, 1998.
12. G. S. Guthart and J. K. Salisbury. The IntuitiveTM telesurgery system: Overview and application. In *Proc. IEEE Int'l. Conf. on Robotics and Automation*, pages 618–621, 2000.
13. Intuitive Surgical. <http://www.intuitivesurgical.com>.
14. T. Itoh, K. Kosuge, and T. Fukuda. Human-machine cooperative telemanipulation with motion and force scaling using task-oriented virtual tool dynamics. *IEEE Trans. Robotics and Automation*, 16(5):505–516, 2000.
15. L. D. Joly and C. Andriot. Imposing motion constraints to a force reflecting telerobot through real-time simulation of a virtual mechanism. In *Proc. IEEE Int'l. Conf. on Robotics and Automation*, pages 357–362, 1995.
16. D. Kragic, P. Marayong, M. Li, A. M. Okamura, and G. D. Hager. Human-machine collaborative systems for microsurgical applications. *Int'l. J. Robotics Research*, 24(9):731–741, 2005.
17. A. B. Kuang, S. Payandeh, B. Zheng, F. Henigman, and C. L. MacKenzie. Assembling virtual fixtures for guidance in training environments. In *Proc. Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2004.
18. M. Li, A. Kapoor, and R. H. Taylor. A constrained optimization approach to virtual fixtures. In *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pages 2924–2929, 2005.
19. M. Li and A. M. Okamura. Recognition of operator motions for real-time assistance using virtual fixtures. In *Proc. Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, pages 125–131, 2003.
20. M. Li and R. H. Taylor. Spatial motion constraints in medical robot using virtual fixtures generated by anatomy. In *Proc. IEEE Int'l. Conf. on Robotics and Automation*, pages 1270–1275, 2004.
21. P. Y. Li and R. Horowitz. Passive velocity field control of mechanical manipulators. *IEEE Trans. Robotics and Automation*, 15(4):751–763, 1999.
22. P. Marayong, G. D. Hager, and A. M. Okamura. Effect of hand dynamics on virtual fixtures for compliant human-machine interfaces. In *Proc. Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2006.

23. P. Marayong and A. M. Okamura. Speed-accuracy characteristics of human-machine cooperative manipulation using virtual fixtures with variable admittance. *Human Factors*, 46(3):518–532, 2004.
24. A. Micaelli, C. Bidard, and C. Andriot. Decoupling control based on virtual mechanisms for telemanipulation. In *Proc. IEEE Int'l. Conf. on Robotics and Automation*, pages 1924–1931, 1998.
25. C. A. Moore, M. A. Peshkin, and J. E. Colgate. Cobot implementation of virtual paths and 3-D virtual surfaces. *IEEE Trans. Robotics and Automation*, 19(2):347–351, 2003.
26. S. Park, R. D. Howe, and D. F. Torchiana. Virtual fixtures for robotic cardiac surgery. In *Proc. Int'l. Conf. on Medical Image Computing and Computer-Assisted Intervention*, pages 1419–1420, 2001.
27. Y. S. Park, H. Kang, T. F. Ewing, E. L. Faulring, J. E. Colgate, and M. A. Peshkin. Enhanced teleoperation for D & D. In *Proc. IEEE Int'l. Conf. on Robotics and Automation*, pages 3702–3707, 2004.
28. S. Payandeh and Z. Stanisic. On application of virtual fixtures as an aid for telemanipulation and training. In *Proc. Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, pages 18–23, 2002.
29. L. Rosenberg. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Proc. IEEE Virtual Reality Int'l. Symposium*, pages 76–82, 1993.
30. J. Roy, D. L. Rothbaum, and L. L. Whitcomb. Haptic feedback augmentation through position based adaptive force scaling: Theory and experiment. In *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pages 2911–2919, 2002.
31. C. Sayers. *Remote Control Robotics*. Springer-Verlag, New York, 1999.
32. SensAble Technologies. <http://www.sensable.com>.
33. R. Taylor, P. Jensen, L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, P. Gupta, Z. Wang, E. deJuan, and L. Kavoussi. Steady-hand robotic system for microsurgical augmentation. *Int'l. J. Robotics Research*, 18(12):1201–1210, 1999.
34. N. Turro, O. Khatib, and E. Coste-Maniere. Haptically augmented teleoperation. In *Proc. IEEE Int'l. Conf. on Robotics and Automation*, pages 386–392, 2001.
35. M. Wu, J. J. Abbott, and A. M. Okamura. Effect of velocity on human force control. In *Proc. Joint EuroHaptics Conf. and Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems (World Haptics)*, pages 73–79, 2005.
36. C. B. Zilles and J. K. Salisbury. A constraint-based god-object method for haptic display. In *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pages 146–151, 1995.

Session Overview

Planning

Nicholas Roy¹ and Roland Siegwart²

¹ MIT

² ETH Zürich

When we discuss autonomous robots, we think of robots that move around, interacting with people and making changes in the world. The problem of actually choosing motor commands to achieve high level goals — such as moving to a desired destination or answering a query from a human — typically involves planning. Planning is of course one of the central questions of artificial intelligence, and the planning field has moved a long way from the early days when planning meant searching for a sequence of abstract actions that satisfied some symbolic predicate. Robots can now learn their own representations through statistical inference procedures, they can now reason using different representations and they can reason in worlds where action can have stochastic outcomes.

However, despite the successes of robots that use machine learning and statistical inference in such different areas as mapping, speech recognition, computer vision, etc., there remain open questions to be addressed before we will see ubiquitous, useful, mobile robots, and some of the most interesting problems are in the planning domain. Consider a mobile robot deployed in some populated environment such as the home. A human operator typically drives the robot around in order to collect sensor data. This data is then used to build a “good” map that is largely static. The robot planner then computes good paths through this map, assuming that the map is correct and complete. The planning system rarely has the ability to reason about the robot’s position within the map and how different plans may lead to better or worse localization. The planner almost never has ability to reason about the quality of the map itself and plan to gather more data in order to get a better map. In contrast, a planner that can reason about how much it knows about the world, and can plan to learn more when necessary, is likely to be a much more robust and general system.

What is becoming clear as robots become increasingly sophisticated is that there are three key issues in planning for mobile robots. Firstly, robots must be able to reason about uncertainty at all levels, both in the current state but also in the current representation. Secondly, robots must be able to plan in

extremely high dimensional spaces. Thirdly, robots must be able to plan in populated and dynamic spaces. Each of these three issues is addressed by the papers in the planning section of ISRR; these papers shed new light on these problems and provides new tools for autonomous robots.

Planning with uncertainty Pineau & Gordon's paper describes the PEMA algorithm for solving large Partially Observable Markov Decision Processes (POMDPs), in which planning decisions are made with respect to the full probability distribution over the state space. POMDPs in particular have been considered computationally intractable for any real world problems, but this paper demonstrates that good approximation techniques can be used to generate plans that lead to overall more robust performance for robots in uncertain worlds. Additionally, the PEMA algorithm addresses a fairly important problem of how a planning algorithm should reason about its model. PEMA uses sampled beliefs, or probability distributions, in the planning process; PEMA demonstrates an approach to choosing these samples intelligently, improving the overall plan.

Planning in high-dimensional spaces In order to find plans in high-dimensional problems, conventional discretization techniques have been superseded by techniques that sample configurations from the world and then retain only those samples that are useful configurations. Hsu, Latombe and Kurniawati address some important questions at the heart of stochastic sampling planners, in particular why these techniques work well, and they describe theoretically why some variants of the sampling techniques have not represented improvements. The critical issue is to recognize that the closer the sampling measure is to the desired plan, the better the performance. Most sampling techniques are a long way from achieving this goal, but this paper points the way to developing even more efficient planners.

Planning in populated worlds Finally, Alami et al.'s presentation on planning in human environments highlighted the need to start building human models into autonomous systems. For example, being able to deal with unpredictable people safely is a critical issue, and one of the results in this paper describes a motion planning algorithm with the objective of safety around people. Additionally, knowing how to behave reasonably around people in highly ambiguous situations is also essential.

It is worth pointing out that all three of these topics are highly related. Planning under uncertainty inevitably leads to planning in high-dimensional information spaces. Planning around people inevitably requires planning under uncertainty. These ideas will be essential for furthering the field of autonomous robots.

POMDP Planning for Robust Robot Control

Joelle Pineau¹ and Geoffrey J. Gordon²

¹ School of Computer Science, McGill University, Montreal CANADA
jpineau@cs.mcgill.ca

² Center for Automated Learning and Discovery, Carnegie Mellon University,
Pittsburgh PA ggordon@cs.cmu.edu

POMDPs provide a rich framework for planning and control in partially observable domains. Recent new algorithms have greatly improved the scalability of POMDPs, to the point where they can be used in robot applications. In this paper, we describe how approximate POMDP solving can be further improved by the use of a new theoretically-motivated algorithm for selecting salient information states. We present the algorithm, called PEMA, demonstrate competitive performance on a range of navigation tasks, and show how this approach is robust to mismatches between the robot's physical environment and the model used for planning.

1 Introduction

The Partially Observable Markov Decision Process (POMDP) has long been recognized as a rich framework for real-world planning and control problems, especially in robotics. However exact solutions are typically intractable for all but the smallest problems. The main obstacle is that POMDPs assume that world states are not directly observable, therefore plans are expressed over *information states*. The space of information states is the space of all beliefs a system might have about the world state. Information states are easy to calculate from sensor measurements, but planning over them is generally considered intractable, since the number of information states grows exponentially with planning horizon.

Recent point-based techniques for approximating POMDP solutions have proven effective for scaling-up planning in partially observable domains [5, 10, 11]. These reduce computation by optimizing a value function over a small subset of information states (or *beliefs*). Often, the quality of the solution depends on which beliefs were selected, but most techniques use ad-hoc methods for selecting beliefs.

In this paper, we describe a new version of the point-based value approximation which features a theoretically-motivated approach to belief point

selection. The main insight is to select points which minimize a bound on the error of the value approximation. This allows us to solve large problems with fewer points than previous algorithms, which leads to faster planning times. Furthermore because a reachability analysis is used to select candidate points, we restrict the search to relevant dimensions of the belief, thereby alleviating the curse of dimensionality.

The new algorithm is key to the successful control of an indoor mobile service robot, designed to seek and assist the elderly in residential environments. The experiments we present show the robustness of the approach to a variety of challenging factors, including limited sensing, sensor noise, and inaccurate models.

2 Background

The Partially Observable Markov Decision Process (POMDP) provides a general framework for acting optimally in partially observable domains. It is well-suited to a great number of robotics problems where decision-making must be robust to sensor noise, stochastic controls, and poor models. This section first establishes the basic terminology and essential concepts pertaining to POMDPs.

2.1 Basic POMDP Terminology

We assume the standard formulation, whereby a POMDP is defined by the n-tuple: $\{S, A, Z, b_0, T, O, R\}$. The first three components, S , A and Z denote finite, discrete sets, where S is the set of states, A is the set of actions, and Z is the set of observations. In general, it is assumed that the state at a given time t , s_t , is not observable, but can be partially disambiguated through the observation z_t . The next three quantities, b_0 , T , and O define the probabilistic world model that underlies the POMDP: b_0 describes the probability that the domain is in each state at time $t = 0$; $T(s, a, s')$ describes the state-to-state transition probabilities (e.g. robot motion model); $O(s, a, z)$ describes the observation probability distribution (e.g. sensor model). And $R(s, a) : S \times A \rightarrow \mathbb{R}$ is a (bounded) reward function quantifying the utility of each action for each state.

2.2 Belief Computation

POMDPs assume that the state s_t is not directly observable, but instead the agent perceives observations $\{z_1, \dots, z_t\}$ which convey information about the state. From these, the agent can compute a *belief*, or probability distribution over possible world states: $b_t(s) = Pr(s_t = s | z_t, a_{t-1}, z_{t-1}, \dots, a_0)$. Because POMDPs are instances of Markov processes, the belief b_t at time t can be

calculated recursively, using only the belief one time step earlier, b_{t-1} , along with the most recent action a_{t-1} and observation z_t :

$$b_t(s) = \tau(b_{t-1}, a_{t-1}, z_t) := \frac{\sum_{s'} O(s', a_{t-1}, z_t) T(s, a_{t-1}, s') b_{t-1}(s')}{Pr(z_t | b_{t-1}, a_{t-1})}. \quad (1)$$

This is equivalent to the Bayes filter, and in robotics, its continuous generalization forms the basis of the well-known Kalman filter. In many large robotics applications, tracking the belief can be computationally challenging. However in POMDPs, the bigger challenge is the generation of an action-selection policy. We assume throughout this paper that the belief can be computed accurately, and focus on the problem of finding good policies.

2.3 Policy Computation

The POMDP framework's primary purpose is to optimize an action-selection *policy*, of the form: $\pi(b) \rightarrow a$, where b is a belief distribution and a is the action chosen by the policy π . We say that a policy $\pi^*(b_t)$ is optimal when the expected future discounted reward is maximized:

$$\pi^*(b_t) = \operatorname{argmax}_{\pi} E_{\pi} \left[\sum_{t=t_0}^T \gamma^{t-t_0} r_t \middle| b_t \right]. \quad (2)$$

Computing an optimal policy over all possible beliefs can be challenging [2], and so many recent POMDP approximations have been proposed which gain computational advantage by applying value updates at a few specific belief points [7, 5, 10, 11]. These techniques differ in how they select the belief points, but all use the same procedure for updating the value over a fixed set of points. The key to updating a value function over a fixed set of beliefs, $B = \{b_0, b_1, \dots, b_q\}$, is in realizing that the value function contains at most one α -vector for each belief point, thus yielding a fixed-size solution set: $\Gamma_t = \{\alpha_0, \alpha_1, \dots, \alpha_q\}$.

The standard procedure for point-based value update is the following. First we generate intermediate sets $\Gamma_t^{a,*}$ and $\Gamma_t^{a,z}$, $\forall a \in A, \forall z \in Z$:

$$\Gamma_t^{a,*} \leftarrow \{\alpha^{a,*}\}, \text{ where } \alpha^{a,*}(s) = R(s, a) \quad (3)$$

$$\Gamma_t^{a,z} \leftarrow \{\alpha_i^{a,z} \mid \alpha_i \in \Gamma_{t-1}\}, \text{ where } \alpha_i^{a,z}(s) = \gamma \sum_{s' \in S} T(s, a, s') O(s', a, z) \alpha_i(s').$$

Next, we take the expectation over observations and construct Γ_t^b , $\forall b \in B$:

$$\Gamma_t^b \leftarrow \{\alpha^{a,b} \mid a \in A\}, \text{ where } \alpha^{a,b} = \Gamma_t^{a,*} + \sum_{z \in Z} \operatorname{argmax}_{\alpha \in \Gamma_t^{a,z}} \sum_{s \in S} \alpha(s) b(s). \quad (4)$$

Finally, we find the best action for each belief point:

$$\Gamma_t \leftarrow \{\alpha^b \mid b \in B\}, \text{ where } \alpha^b = \operatorname{argmax}_{\alpha \in \Gamma_t^b} \sum_s \alpha(s)b(s). \quad (5)$$

Because the size of the solution set Γ_t is constant, the point-based value update can be computed in polynomial time. And while these operations preserve only the best α -vector at each belief point $b \in B$, an estimate of the value function at any belief in the simplex (including $b \notin B$) can be extracted from the set Γ_t :

$$V_t(b) = \max_{\alpha \in \Gamma_t} \sum_{s \in S} \alpha(s)b(s). \quad (6)$$

2.4 Error Bound on Point-Based Value Updates

The point-based value update operation is an integral part of many approximate POMDP solvers. As shown in [5], given a fixed belief set B and planning horizon t , the error over multiple value updates is bounded by³:

$$\|V_t^B - V_t^*\|_\infty \leq \frac{(R_{\max} - R_{\min}) \max_{b' \in \Delta} \min_{b \in B} \|b - b'\|_1}{(1 - \gamma)^2}$$

where $b' \in \Delta$ is the point where the point-based update makes its worst error in value update, and $b \in B$ is the closest (1-norm) sampled belief to b' . Now let α be the vector that is maximal at b , and α' be the vector that would be maximal at b' . Then, we can show equivalently that

$$\begin{aligned} \epsilon(b') &\leq \alpha' \cdot b' - \alpha \cdot b' \\ &\leq (\alpha' - \alpha) \cdot (b' - b) \\ &\leq \sum_i \begin{cases} \left(\frac{R_{\max}}{1-\gamma} - \alpha_i\right)(b'_i - b_i) & b'_i \geq b_i \\ \left(\frac{R_{\min}}{1-\gamma} - \alpha_i\right)(b'_i - b_i) & b'_i < b_i. \end{cases} \end{aligned}$$

3 Error-Minimization Point Selection

Many recent point-based value approximations, which show good empirical success, use poorly informed heuristics to select belief points. We now describe a new algorithm for selecting provably good belief points. The algorithm directly uses the error bound above to pick those reachable beliefs $b \in \bar{\Delta}$ which most reduce the error bound. Figure 1a shows the tree of reachable beliefs, starting with the initial belief (top node). Building the tree (to a finite depth) is easily done by recursively using Equation 1.

³ The error bound proven in [5] depends on the sampling density over the belief simplex Δ . But when the initial belief b_0 is known, it is not necessary to sample all of Δ densely. Instead, we can sample the set of reachable beliefs $\bar{\Delta}$ densely; the error bound holds on $\bar{\Delta}$.

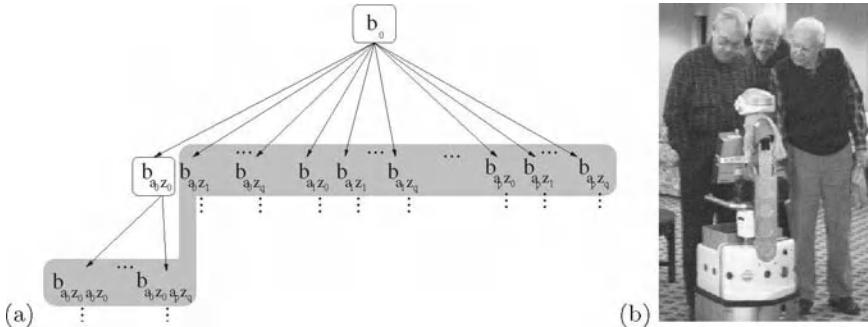


Fig. 1. (a) The set of reachable beliefs. Each node corresponds to a specific belief, and increasing depth corresponds to an increasing plan horizon.(b) Pearl the Nursebot interacting with patients in a nursing facility.

Applying point-based value updates to *all* reachable beliefs would guarantee optimal performance, but at the expense of computational tractability: a planning problem of horizon t has $O(|A||Z|^t)$ reachable beliefs. So we select from our reachable beliefs those most likely to minimize the error in our value function. Given the belief tree in Figure 1a, we consider three sets of nodes. Set 1 includes all points already in B (in this example b_0 and b_{a_0, z_0}). Set 2 contains the set of candidates from which we will select new points to be added to B . We call this set the *fringe* (denoted \bar{B}). Set 3 contains all other reachable beliefs.⁴

Now we need to decide which belief b should be removed from the fringe \bar{B} and added to the set of active points B . Every new point added to B should improve our estimate of the value function as much as possible. To find the point most likely to do this, we consider the theoretical analysis of Section 2.4. Consider $b' \in \bar{B}$, a belief point candidate, and $b \in B$, some belief which we have already selected. While one could simply pick the candidate $b' \in \bar{B}$ with the largest error bound, $\epsilon(b')$, this would go against the most useful insight from earlier work on point-based approaches: namely that *reachability* considerations are important. So we need to factor in the probability of each candidate belief point occurring. We first note that the error bound at any given belief point b in the tree can be evaluated from that of its immediate descendants:

$$\bar{\epsilon}(b) = \max_{a \in A} \sum_{z \in Z} O(b, a, z) \cdot \epsilon(\tau(b, a, z)) \quad (7)$$

⁴ In Figure 1a, the fringe (\bar{B}) is restricted to the immediate descendants of the points in B . The rest of the paper proceeds on this assumption, but we could assume a deeper fringe.

where $\tau(b, a, z)$ is the belief update equation (Eqn 1), and $\epsilon(\tau(b, a, z))$ is evaluated as in Section 2.4 (unless $\tau(b, a, z) \in B$, in which case $\epsilon(\tau(b, a, z)) = 0$). So we use Equation 7 to find the existing point $b \in B$ with the largest error bound, then pick as a new point its descendant $\tau(b, a, z)$ which has the largest impact on $\bar{\epsilon}(b)$. Points on the fringe are picked one at a time, allowing us to look deep in the tree; in the experiments presented below, beliefs at 40+ levels are in fact selected.

This concludes the presentation of our new error-minimization point selection technique. In practice, the addition of new points is always interleaved with the point-based value updates described in Section 2.3 to form a full POMDP solution. The complete approach, called PEMA (**P**oint-based **E**rror **M**inimization **A**lgorithm), is now evaluated empirically in a series of robot control experiments.

4 Empirical Evaluation

We begin our empirical evaluation with a few well-studied maze navigation domains. Most have been used strictly in simulation, but feature robot-like assumptions, such as non-deterministic motion and noisy sensors. The Tiger-grid, Hallway and Hallway2 problems are described in [3]. The Tag domain was introduced in [5]. The goal of these preliminary experiments is simply to compare the performance of PEMA with earlier POMDP approximations on standard problems. More extensive robot navigation domains are presented in the following section.

Error estimates. A first set of results on PEMA’s performance are shown in Figure 2. For each problem domain, we first plot PEMA’s reward performance as a function of the number of belief points (top graphs), and then plot the error estimate of each point selected according to the order in which points were picked (bottom graphs). As shown in these, PEMA is able to solve all four problems with relatively few beliefs (sometimes fewer than the number of states).

Considering the error bound graphs, we see that overall there seems to be reasonably good correspondence between an improvement in performance, and a decrease in the error estimates. We can conclude from these plots that the error bound used by PEMA is quite informative in guiding exploration of the belief simplex.⁵

⁵ While the decrease in error over a fixed point (e.g. b_0) is monotonic, the decrease in error over *each new points* (in the order it was added) is not necessarily monotonic, which explains the large jumps in the bottom graphs. These jumps suggest that PEMA could be improved by maintaining a deeper fringe of candidate belief points, in which case the time spent selecting points would have to be carefully balanced with the time spent planning. Currently, we spend less than 1% of computation time selecting belief points; the rest is spent estimating the value function.

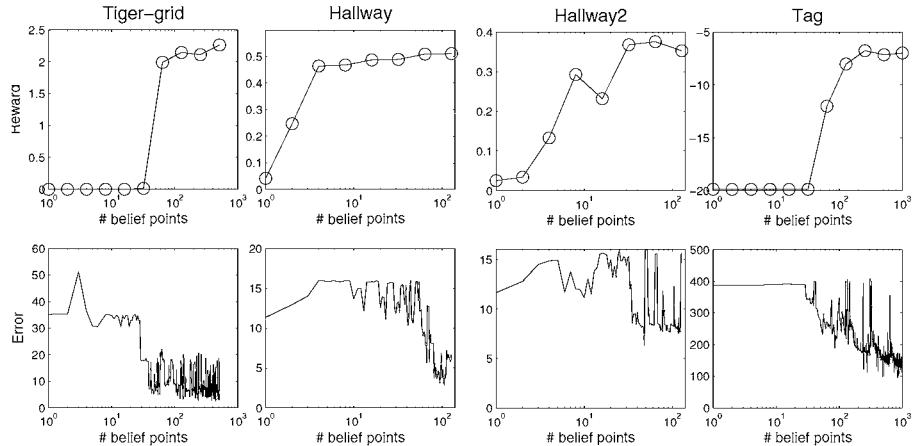


Fig. 2. Policy performance (top row) and estimate of the bound on the error (bottom row) for selected belief points

Comparative analysis. While the results outlined above show that PEMA is able to handle a wide spectrum of large-scale POMDP domains, it is also useful to compare its performance to that of alternative approaches, on the same set of problems. Figure 3 compares both reward performance and policy size⁶ (# of nodes in controller) for a few recent POMDP algorithms, on the three larger problems (Hallway, Hallway2, and Tag). The algorithms included in this comparison were selected simply based on the availability of published results for this set of problems.

As is often the case, these results show that there is not a single algorithm that is best for solving all problems, so it is difficult to draw broad generalizations. But we can point out a few salient effects. First, the baseline QMDP [3] approximation is clearly outclassed by other more sophisticated methods. We also observe that some of the algorithms achieve sub-par performance in terms of expected reward: BPI [9] (on Hallway2 and Tag)⁷, PBVI [5] (on Tag) and BBSLS [1] (on Tag). While each of these is theoretically able to reach optimal performance, they would require larger controllers (and therefore longer computation time) to do so.

The remaining algorithms—HSVI [10], Perseus [11], and PEMA—offer comparable performance. HSVI offers good control performance on the full range of tasks, but requires bigger controllers. HSVI and PEMA share many

⁶ The results were computed on different platforms, so time comparisons are difficult. The size of the final policy is often a useful indicator of computation time, but should be considered with care.

⁷ Better results for BPI have since been published in [8].

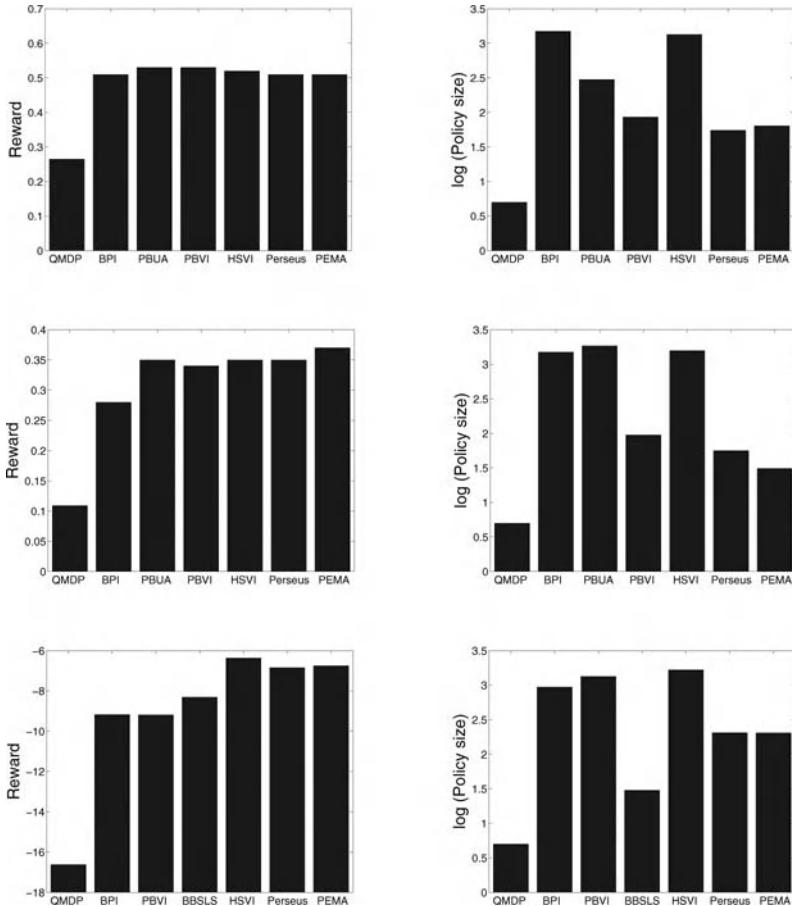


Fig. 3. Results for standard POMDP domains. Top row: Hallway problem. Middle row: Hallway2 problem. Bottom row: Tag problem.

similarities: both use an error bound to select belief points. HSVI's upper-bound is tighter than PEMMA's, but requires costly LP solutions. PEMMA solves problems with fewer belief points, we believe this is because it updates all belief points more frequently, thus generalizing better in poorly explored areas of the belief simplex.

Between Perseus and PEMMA, the trade-offs are less clear: the planning time, controller size and performance quality are quite comparable. These two approaches in fact share many similarities. Perseus uses the same point-based backups as in PEMMA (see Section 2.3), but it differs in both how the set of belief points is selected (Perseus uses random exploration traces), and the order in which it updates the value at those points (also randomized). The

effect of these differences is hard to narrow. We did experiment informally with Perseus-type random updates within PEMA, but this did not yield significant speed-up. It is likely that randomizing value updates is not as beneficial when carefully picking a small set of essential points. We speculate that PEMA will scale better to higher dimensions because of the selective nature of the belief sampling. This is the subject of ongoing work.

5 Robotic Applications

Much of the algorithmic development described in this paper is motivated by our need for high-quality robust planning for interactive mobile robots. In particular, we are concerned with the problem of controlling a nursing assistant robot. This is an important technical challenge arising from the Nursebot project [6]. This project aims to develop personalized robotic technology that can improve the level of personal care and services for elderly individuals. The robot Pearl (Fig. 1b) is the main experimental platform used in this project. It is equipped with standard indoor navigation abilities and is programmed with the CARMEN toolkit [4]. An important task for this robot is to provide timely cognitive reminders (e. g. medications to take, appointments to attend, etc.) to its target population. It is therefore crucial that the robot be able to find the person whenever it is time to issue a reminder. We model this task as a POMDP, and use PEMA to optimize a strategy with which the robot can robustly find the person, even under very weak assumptions over the person's initial location and ease of mobility.

We begin by considering the environment in which the robot operates. Figure 5 shows a 2D robot-generated map of its physical environment. The goal is for the robot to navigate in this environment until it finds the patient and then deliver the appropriate reminder. To successfully find the patient, the robot needs to systematically explore the environment, while reasoning about both its spatial coverage and the likely motion pattern of the person.

5.1 POMDP Modeling

To model this task as a POMDP, we assume a state space consisting of two features: *RobotPosition*, and *PersonPosition*. Each feature is expressed through a fixed discretization of the environment (roughly 25 cells for each feature, or 625 total states.) We assume the person and robot move freely, constrained only by walls and obstacles. The robot's motion is deterministic (as a function of the action=*{North, South, East, West}*). A fifth action (*DeliverMessage*) concludes the scenario if applied when the robot and person are in the same location. We assume the person's motion is stochastic, and in one of two modes: (1) whenever the person is far from the robot, s/he moves according to Brownian motion (i. e. in each cardinal direction with $Pr = 0.1$ or stays in place), this corresponds to a random walk and is a conservative assumption

regarding people’s motion; or (2) whenever the robot is within sight ($< 4\text{m}$), the person tries to avoid the robot and moves away from it (with noise), which makes the task more challenging.

The observation function has two parts: what the robot senses about its own position, and what it senses about the person’s position. First we assume that the robot’s position is fully known; this is reasonable since planning is done at a much coarser resolution (2m), than the typical localization precision (10cm). When testing policies however, probabilistic localization is performed by the CARMEN toolkit, and the robot’s belief incorporates any positional uncertainty. For the person’s position, we assume that the robot perceives nothing unless the person is within 2 meters. This is plausible given the robot’s sensors. Even at short-range, there is a small probability ($Pr = 0.01$) that the robot will miss the person.

The reward function is straightforward: $R = -1$ for any motion, $R = 10$ when the robot decides to *DeliverMessage* and is within range ($< 2\text{m}$) of the person, and $R = -100$ when the robot decides to *DeliverMessage* in the person’s absence. The task terminates when the robot successfully delivers the message. We assume a discount factor proportional to the map’s resolution ($\gamma = 0.98$).

With these POMDP parameters, we can run PEMA to optimize the robot’s control strategy. Given the complexity of POMDP planning we do assume that PEMA will be used as an off-line algorithm to optimize the robot’s performance prior to deployment. The results presented below describe the performance of an optimized control policy when tested onboard the CARMEN simulator.

5.2 Experimental Results

We first consider PEMA’s performance on this task, as a function of planning time. As shown in Figure 4, PEMA is in fact able to solve the problem within 1800 seconds, using only 128 belief points. In comparison, an MDP-type approximation (in this case the QMDP technique [3]) proves to be inadequate for a problem exhibiting such complex uncertainty over the person’s position. Using PEMA, the patient was found in 100% of trials, compared to 35% for QMDP.

Figure 5 shows PEMA’s policy through five snapshots from one run. The policy is optimized for any start positions (for both the person and the robot); the execution trace in Figure 5 is one of the longer ones since the robot searches the entire environment before finding the person. In this scenario, the person starts at the far end of the left corridor. The person’s location is not shown in the figure since it is not observable by the robot. The figure instead shows the *belief* over person positions, represented by a distribution of point samples (grey dots). We see the robot starting at the far right end of the corridor (Fig. 5a), moving towards the left until the room’s entrance (Fig. 5b), and searching the entire room (Fig. 5c). Once sufficiently certain that the person

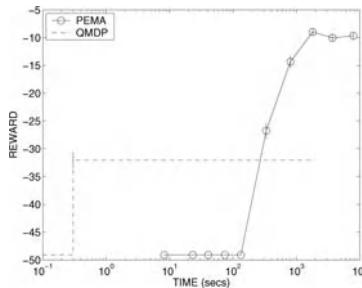


Fig. 4. Find-the patient domain - Performance results.

is not there, it exits the room (Fig. 5d), and moves towards the left until it finally finds the person at the end of the corridor (Fig. 5e).

It is interesting to compare snapshots (b) and (d). The robot position in both is practically identical. Yet in (b) the robot chooses to go up into the room, whereas in (d) the robot chooses to move toward the left. This is a direct result of planning over *beliefs*, rather than over *states*.

These results show that PEMA is able to handle realistic domains. In particular, throughout these experiments, the robot simulator was in no way constrained to behave as described in our POMDP model. For example the robot's actions often had stochastic effects, the robot's position was not always fully observable, and belief tracking had to be performed asynchronously (i. e. not a straight alternation of actions and observations). Despite this mismatch between the model assumed for planning and the execution environment, the control policy optimized by PEMA successfully completed the task.

5.3 Robustness to Modeling Errors

Like most POMDP solvers, PEMA assumes exact knowledge of the POMDP model. In reality, this model is often hand-crafted and may bear substantial error. In our experience, such a mismatch between model and the real system does not necessarily render our solution useless. The robustness built in to POMDPs to overcome state uncertainty often goes a long way towards overcoming model uncertainty. Nonetheless, there are cases where a poor model can be catastrophic. In this section, we try to gain a better understanding of the impact of errors in the model we used for the Find-the-patient domain.

Our model assumes that the robot can see the patient with $Pr = 0.99$, whenever s/he is within 2m. We use this parameter both for solving and tracking. But it could be that in fact the person is only detected with $Pr = 0.8$.

What would be the loss in performance, compared to if we had planned and tracked with the correct parameter? Table 1 examines the effects of this type of modeling error. It shows the performance (avg. sum of rewards over 1000 trajectories) when applying PEMA and tracking the belief with the sensor accuracy in the left column, but testing with the accuracy in the top row.

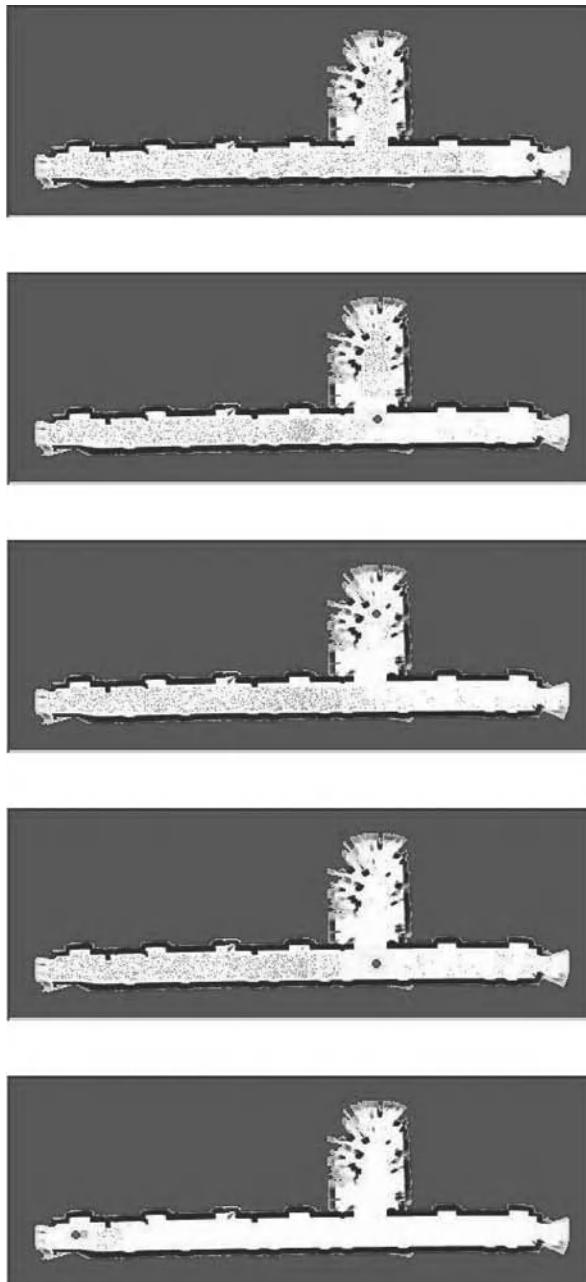


Fig. 5. Find-the patient domain - Sample trajectory.

Table 1. Sensitivity analysis over observation probabilities. (CI for all: [0.7,1.4])

$\Pr_{model}(z)$	$\Pr_{real}(z)$			
	0.99	0.90	0.80	0.70
0.99	-9.7	-11.3	-13.2	-15.5
0.90	-12.0	-13.1	-15.6	-19.0
0.80	-9.7	-11.5	-13.1	-14.5
0.70	-17.8	-19.4	-22.0	-22.6

The main diagonal contains cases where the model is correct. These results suggest two things. First, as expected, performance degrades as the real noise level increases (i.e. left-to right effect for any given row.) Second, and this was not anticipated, the dominating performance factor is in fact the noise in the assumed model: regardless of what conditions are used for testing, results are better for some values of \Pr_{model} (0.99 and 0.8) and worse for others (0.9 and 0.7). We hypothesize that this happens because in some models, PEMA did not have sufficient belief points to perform well (all policies were optimized with $|B|=512$). When we repeated experiments for $\Pr_{model}(z)=0.9$ with more beliefs points, the performance improved (for all $\Pr_{real}(z)$) to the level of the top row. This suggest that in some domains it may be best to optimize policies assuming false models (e. g. low sensor noise), because an equally good policy can be obtained with fewer belief points. We are currently investigating this, as well as the impact of modeling errors in the transition model.

6 Conclusion

This paper describes a new algorithm for planning in partially observable domains, which features a theoretically-motivated technique for selecting salient information states. This improves the scalability of the approach, to the point where it can be used to control a robot seeking a missing person. We also demonstrate that the algorithm is robust to noise in the assumed model. Future work focuses on improving performance under even weaker modeling assumptions.

References

1. D. Braziunas and C. Boutilier. Stochastic local search for POMDP controllers. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI)*, pages 690–696, 2004.
2. A. Cassandra, M. L. Littman, and N. L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 54–61, 1997.

3. M. L. Littman, A. R. Cassandra, and L. P. Kaelbling. Learning policies for partially observable environments: Scaling up. In *Proceedings of Twelfth International Conference on Machine Learning*, pages 362–370, 1995.
4. M. Montemerlo, N. Roy, and S. Thrun. Perspectives on standardization in mobile robot programming: The Carnegie Mellon navigation (CARMEN) toolkit. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages pp 2436–2441, 2003.
5. J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1025–1032, 2003.
6. J. Pineau, M. Montermerlo, M. Pollack, N. Roy, and S. Thrun. Towards robotic assistants in nursing homes: challenges and results. *Robotics and Autonomous Systems*, 42(3-4):271–281, 2003.
7. K.-M. Poon. A fast heuristic algorithm for decision-theoretic planning. Master’s thesis, The Hong-Kong University of Science and Technology, 2001.
8. P. Poupart. *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. PhD thesis, University of Toronto, 2005.
9. P. Poupart and C. Boutilier. Bounded finite state controllers. In *Advances in Neural Information Processing Systems (NIPS)*, volume 16, 2004.
10. T. Smith and R. Simmons. Heuristic search value iteration for POMDPs. In *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2004.
11. N. Vlassis and M. T. J. Spaan. A fast point-based algorithm for POMDPs. In *Proceedings of the Belgian-Dutch Conference on Machine Learning*, 2004.

Humanoids

On the Probabilistic Foundations of Probabilistic Roadmap Planning

David Hsu^{1,2}, Jean-Claude Latombe^{3,*}, and Hanna Kurniawati¹

¹Department of Computer Science, National University of Singapore, Singapore

²Singapore MIT Alliance, Singapore

³Department of Computer Science, Stanford University, Stanford, CA, USA

Summary. Why are probabilistic roadmap (PRM) planners “probabilistic”? This paper tries to establish the probabilistic foundations of PRM planning and re-examines previous work in this context. It shows that the success of PRM planning depends mainly and critically on the assumption that the configuration space C of a robot often verifies favorable “visibility” properties that are not directly dependent on the dimensionality of C . A promising way of speeding up PRM planners is to extract partial knowledge on such properties during roadmap construction and use this knowledge to adjust the sampling measure continuously. This paper also shows that the choice of the sampling source—pseudo-random or deterministic—has small impact on a PRM planner’s performance, compared to that of the sampling measure. These conclusions are supported by both theoretical arguments and empirical results.

1 Introduction

Probabilistic roadmap (PRM) planners [3, Chapter 7] solve seemingly difficult motion planning problems such as the one in Fig. 1, where the robot’s configuration space C is 6-D and the environment consists of tens of thousands of triangles. While an algebraic planner would be overwhelmed by the high cost of computing an exact representation of the free space F , defined as the collision-free subset of C , a PRM planner builds only an extremely simplified representation of F , called a probabilistic roadmap. The nodes of a roadmap R are configurations sampled from F with a suitable probability measure. The edges of R are simple collision-free paths, *e.g.*, straight-line segments, between the sampled configurations. PRM planners work surprisingly well in practice. Why?

Previous work has partially addressed this question by identifying and formalizing free space properties that provide sufficient conditions to guarantee that a PRM planner using a uniform sampling measure works well. However, the underlying question “Why are PRM planners probabilistic?” has received little attention so far, and consequently the role of non-uniform sampling measures in

* Part of this work was completed while the author was at the National University of Singapore, supported by the Kwan Im Thong Hood Cho Temple Professorship.

PRM planning remains poorly understood. Since no inherent randomness or uncertainty exists in the classic formulation of motion planning problems like the one depicted in Fig. 1, one may wonder why probabilistic sampling helps to solve them.

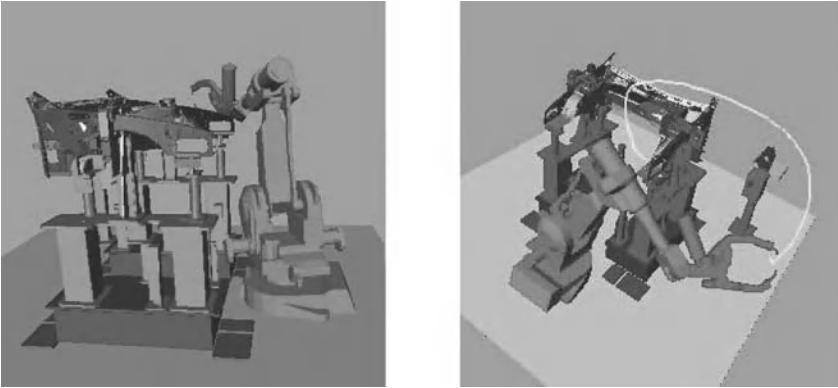


Fig. 1. A practical motion planning problem.

In this paper, we attempt to fill this gap, with the intent of identifying promising directions to improve future PRM planners. We introduce the probabilistic foundations of PRM planning (Section 2). We then examine previous work in this context and argue that the empirical success of PRM planning tells us as much about the nature of motion planning problems encountered in practice as about PRM planning itself (Section 3). We emphasize the important distinction between the sampling *measure*, a notion firmly rooted in probability theory, and the sampling *source*, and show that the source has small impact on a planner’s performance compared to the measure (Sections 4 and 5).

The main questions addressed in this paper are summarized below:

- **Why is PRM planning “probabilistic”?** A foundational choice in PRM planning is to avoid computing an exact representation of F . So the planner never knows the exact shape of F , in particular, its connectivity. It works very much like a robot exploring an *unknown* environment to build a map. At any moment during planning, many hypotheses on F are consistent with the configurations sampled so far. The probability measure for sampling F reflects this uncertainty. Hence, PRM planning trades the cost of computing F exactly against the cost of dealing with uncertainty. This choice is beneficial only if a small roadmap can represent the shape of F well enough to answer motion-planning queries correctly.
- **Why does PRM planning work well?** One can think of the nodes of a roadmap as a network of guards *watching* over F . To guarantee that a PRM planner finds a solution quickly whenever one exists, F should satisfy favorable “visibility” properties. A key contribution of PRM planning is to reveal that in practice, many free spaces satisfy such properties, despite their high algebraic complexity. Since visibility properties can be defined in terms of volume ratios

over certain subsets of F , they do not directly depend on $\dim(C)$, the dimensionality of C . This explains why PRM planning scales up reasonably well when $\dim(C)$ increases.

- **How important is the sampling measure?** In every PRM planner, a probability measure prescribes how sampled configurations are distributed over F . Since visibility properties are in general not uniformly favorable over F , this measure plays a critical role in the efficiency of PRM planning by allocating a higher density of samples to regions with poor visibility properties. Existing PRM planners use mostly simple, heuristic estimates of visibility properties, but experiments show that they dramatically improve the performance of PRM planning.
- **How important is the sampling source?** A PRM planner needs a source S of *uniformly* distributed pseudo-random or deterministic numbers for sampling C . Usually, it calls S to pick a point uniformly from $[0,1]^{\dim(C)}$ and then maps the point into C according to a given probability measure. The source S has only a limited effect on the efficiency of PRM planning. When $\dim(C)$ is small, low-discrepancy or low-dispersion deterministic sources achieve some speedup over pseudo-random sources [13]; however, the speedup is very modest compared to that achieved by good sampling measures and fades away quickly, as $\dim(C)$ increases.

This paper does not introduce any new PRM planner or sampling strategy. Instead, its contribution is to articulate a coherent framework centered on the probabilistic foundations of PRM planning and evaluate several ideas, considered separately before, in this framework. It brings new understanding of what makes PRM planning effective, which in turn may help us to design better planners in the future.

2 Why Is PRM Planning “Probabilistic”?

For many robots, computing an exact representation of the free space F takes prohibitive time, but fast, exact algorithms exist to test whether a given configuration or path is collision-free [16]. PRM planners use two *probes* based on such algorithms to access geometric information from the configuration space C :

- For any $q \in C$, $\text{FreeConf}(q)$ is true if and only if $q \in F$.
- For any pair $q, q' \in C$, $\text{FreePath}(q, q')$ is true if and only if q and q' can be connected with a straight-line path lying entirely in F .

The choice of using only these two probes is foundational for PRM planning. Since a PRM planner does not compute the exact shape of F , it never gains this information. At any moment, many hypotheses on F are consistent with the information gathered so far by the probes, and each hypothesis has some probability of being correct. The probabilistic nature of PRM planners comes from the fact that this uncertainty is modeled implicitly by a probability measure over the set of hypotheses.

In this paper, we use the following scheme, which we call **Basic-PRM**, as a reference planner. Like the original PRM planner [12], it operates in two stages, roadmap construction and roadmap query.

- **Roadmap construction.** The procedure below takes a single input argument N , the number of nodes for the roadmap R to be constructed. The nodes of R are collision-free configurations sampled from F . The edges represent collision-free straight-line paths between the nodes.

Procedure Roadmap-Construction(N)

1. **repeat** until N nodes have been generated
 2. Sample a configuration \mathbf{q} from C uniformly at random.
 3. **if** FreeConf(\mathbf{q}) is true **then** add \mathbf{q} as a new node of R .
 4. **for** every node \mathbf{q}' of R such that $\mathbf{q}' \neq \mathbf{q}$ **do**
 5. **if** FreePath(\mathbf{q}, \mathbf{q}') is true **then** add $(\mathbf{q}, \mathbf{q}')$ as a new edge of R .
 6. **return** R .
-

Most PRM planners use better sampling strategies than the uniform random one in Line 2, as well as better connection strategies in Lines 4–5.

A sampling strategy (π, S) is characterized by a probability *measure* π that prescribes how sampled configurations are distributed over C and a *source* S of uniformly distributed pseudo-random or deterministic numbers. We will show in Sections 4–5 that designing good sampling measures is one of the most promising ways to speed up PRM planning.

- **Roadmap query.** A query is defined by two configurations \mathbf{q}_1 and \mathbf{q}_2 in F . Given a roadmap R , the procedure Roadmap-Query tries to connect each \mathbf{q}_i , $i=1,2$, to a node of R . For each \mathbf{q}_i , it samples uniformly at random K configurations so that for each such configuration \mathbf{q} , FreePath(\mathbf{q}_i, \mathbf{q}) is true. It then checks whether there is a node v_i of R such that FreePath(\mathbf{q}, v_i) is true. If so, \mathbf{q}_i and v_i can be connected via \mathbf{q} . If either \mathbf{q}_1 or \mathbf{q}_2 cannot be connected to a node of R , Roadmap-Query returns FAILURE. Otherwise, it searches for a path in R between v_1 and v_2 . If one is found, it returns a path between \mathbf{q}_1 and \mathbf{q}_2 . Otherwise, it returns NO PATH.

If Roadmap-Query returns a path, the answer is always correct, but the NO PATH answer may not be correct, as disconnected components of R may lie in the same connected component of F . The answer FAILURE means that R is insufficient to answer the query.

Let us now return to the question “Why is PRM planning probabilistic?”. Suppose that while constructing a roadmap, Roadmap-Construction could maintain a representation (H, η) , where H is the set of all hypotheses over the shape of F and η is a probability measure that assigns to each hypothesis in H the probability of it being correct. Suppose further that we can define what a good roadmap is (see Section 3). Then, in each iteration of Roadmap-

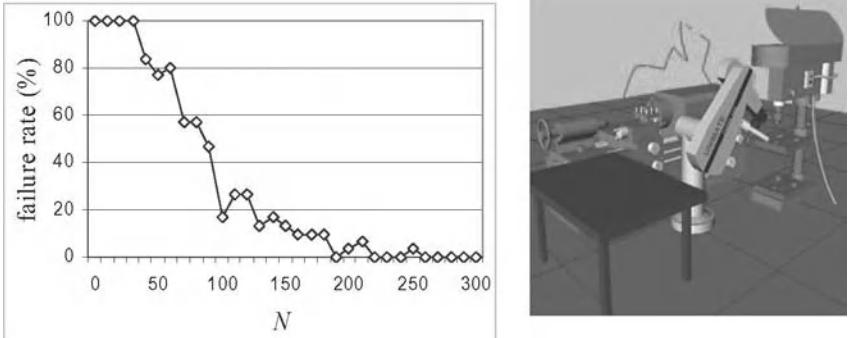


Fig. 2. The experimental convergence rate of Basic-PRM. The plot shows the percentage of unsuccessful outcomes out of 100 independent runs for the same query in the environment shown on the right, as the number of roadmap nodes increases.

Construction, the optimal sampling measure π^* is the one that minimizes the expected number of remaining iterations until a good roadmap is reached, and π^* can be inferred from (H, η) . However, maintaining (H, η) explicitly is expensive. So existing PRM planners use heuristics to select the sampling measure π (see Section 4).

3 Why Does PRM Planning Work Well?

In general, Basic-PRM may return an incorrect NO PATH or FAILURE answer with some probability γ , but the efficiency of PRM planners in practice indicates that γ is usually small. Experiments show that even in complex geometric environments, γ often converges to 0 quickly, as N , the number of roadmap nodes, increases (Fig. 2). Yet one can also easily construct apparently simple environments where PRM planners perform terribly (Fig. 3). Together, these two examples suggest that many environments encountered in practice satisfy favorable properties that PRM planners exploit well. What are these properties?

We now review results from [9, 11], showing that if F satisfies a rather general visibility property, called *expansiveness*, then Basic-PRM answers planning queries correctly with high probability. In the following, the phrase “with high (low) probability in n ” means that the probability converges to 1 (0) at an exponential rate, as n increases.

3.1 Visibility in the Free Space

We say that two points q and q' in F see each other if $\text{FreePath}(q, q')$ is true. The *visibility set* of $q \in F$ is the set $V(q) = \{ q' \in F \mid \text{FreePath}(q, q') \text{ is true} \}$. The

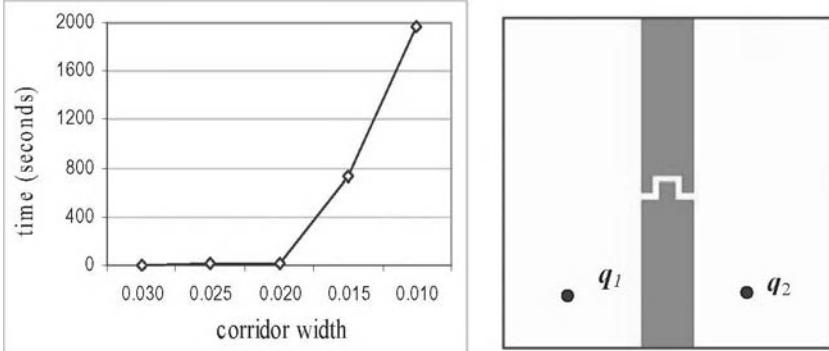


Fig. 3. A difficult example for PRM planning. F consists of two rectangular chambers connected by a narrow corridor. The plot shows the average running time for Basic-PRM to connect the two query configurations, as the corridor width decreases.

definition of a visibility set is extended to any subset M of points in F by defining $V(M) = \cup_{q \in M} V(q)$.

The first two theorems below say that if F satisfies a property called ε -goodness, then Basic-PRM generates a roadmap that provides good *coverage* of F so that FAILURE rarely occurs.

Definition 1. Given a constant $\varepsilon \in (0,1]$, a point $q \in F$ is ε -good if it sees at least an ε -fraction of F , i.e., if $\mu(V(q)) \geq \varepsilon \times \mu(F)$, where $\mu(S)$ denotes the volume of S for any $S \subseteq C$. F is ε -good if every point $q \in F$ is ε -good.

Definition 2. A roadmap R provides *adequate coverage* of an ε -good free space F if the subset of F not seen by any node of R has volume at most $(\varepsilon/2)\mu(F)$.

Theorem 1. If F is ε -good, then with high probability in N , Roadmap-Construction(N) generates a roadmap that provides adequate coverage of F [11].

Theorem 2. If a roadmap provides adequate coverage of F , then Roadmap-Query returns FAILURE with low probability in K [11].

(Recall that K is the number of configurations sampled randomly in the neighborhood of each of the query configurations. See Section 2.)

Adequate coverage only protects us from FAILURE, but does not prevent an incorrect NO PATH answer, because ε -goodness is too weak to imply anything on roadmap connectivity. A stronger property is needed to “link” a visibility set to its complement in F .

Definition 3. Let F' be a connected component of F , G be any subset of F' , and β be a number in $(0,1]$. The β -LOOKOUT of G is the set of all points in G that see at least a β -fraction of the complement of G in F' :

$$\beta\text{-LOOKOUT}(G) = \{q \in G \mid \mu(V(q) \setminus G) \geq \beta \times \mu(F' \setminus G)\}.$$

Suppose that the volume of $\beta\text{-LOOKOUT}(G)$ is at least $\alpha \times \mu(G)$. If either α or β is small, then it would be difficult to sample a point in G and another in $F \setminus G$ so that the two points see each other, hence to build a roadmap that represents the connectivity of F' well. This happens in the free space of Fig. 3 when the corridor is very narrow. These considerations lead to the concept of expansiveness.

Definition 4. Let ε , α , and β be constants in $(0,1]$. A connected component F' of F is $(\varepsilon, \alpha, \beta)$ -expansive if (i) every point $q \in F'$ is ε -good and (ii) for any set M of points in F' , $\mu(\beta\text{-LOOKOUT}(V(M))) \geq \alpha \times \mu(V(M))$. F is $(\varepsilon, \alpha, \beta)$ -expansive, if its connected components are all $(\varepsilon, \alpha, \beta)$ -expansive.

Theorem 3. If F is $(\varepsilon, \alpha, \beta)$ -expansive, then with high probability in N , Roadmap-Construction generates a roadmap whose connected components have one-to-one correspondence with those of F [9].

Expansiveness guarantees that the visibility set $V(M)$ of any set M of points in a connected component F' of F has a large lookout. So it is easy to sample at random a set of configurations and construct a roadmap that both provides good coverage of F and represents the connectivity of F well. The values of ε , α , and β measure the extent to which F is expansive. For example, if F is convex, then $\varepsilon = \alpha = \beta = 1$. The larger these values are, the smaller N needs to be for Basic-PRM to answer queries correctly. Although for a given motion planning problem, we often cannot compute these values in advance, they characterize the nature of free spaces in which PRM planning works well.

3.2 What Does the Empirical Success of PRM Planners Imply?

In practice, a small number of roadmap nodes are often sufficient to answer queries correctly. This frequent success suggests that the main reason for the empirical success of PRM planners is that free spaces encountered in practice often satisfy favorable visibility properties, such as expansiveness. If a connected component F' of F had very small values of ε , α , and β , then a planner would likely encounter a set M of sampled nodes such that $V(M)$ has a small lookout. It would then be difficult to sample a node in this lookout and eventually create a connected roadmap in F' . PRM planners scale up well when $\dim(C)$ increases, because visibility properties can be defined in terms of volume ratios over subsets of F and do not directly depend on $\dim(C)$. So, the empirical success of PRM planning says as much about the nature of motion-planning problems encountered in practice as about the algorithmic efficiency of PRM planning. The fact that many free spaces, despite their high algebraic complexity, verify favorable visibility properties is not obvious a priori. An important contribution of PRM planning has been to reveal this fact.

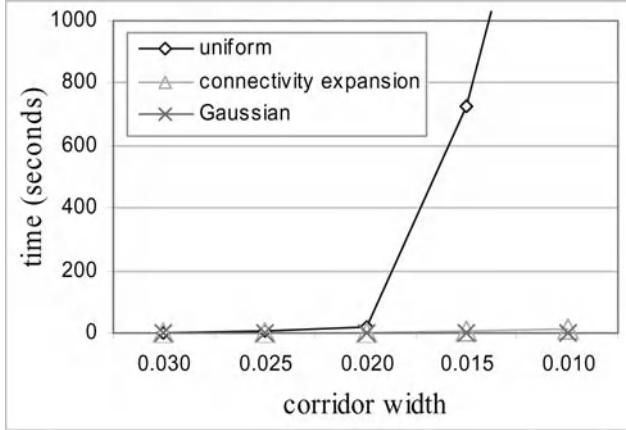


Fig. 4. Comparison of three strategies with different sampling measures. The plot shows the average running times over 30 runs on the problem in Fig. 3 as the corridor width decreases.

According to Theorems 1–3, expansiveness provides a sufficient condition for Basic-PRM work well. One can also prove that expansiveness is necessary in the following sense: if F is not expansive for large enough values of ε , α , and β , then it is always possible to choose a query in F so that Basic-PRM fails to succeed with high probability. This indicates that expansiveness is a good characterization of the complexity of the free space for PRM planning. We do not have a proof that expansiveness is the *minimal* property that F must satisfy for PRM planners to work well, but few alternatives exist (*e.g.*, path clearance and ε -complexity) and they are more specific. However, since the values of ε , α , and β are determined by the worst configurations and lookouts in F , they do not reflect the variation of visibility properties over F . This is precisely what non-uniform sampling measures described below try to exploit.

4 How Important Is the Sampling Measure?

In the previous section, we have analyzed the performance of Basic-PRM when the uniform sampling measure is used. However, most PRM planners employ non-uniform sampling measures that dramatically improve performance. To illustrate, Fig. 4 compares the average running times of three versions of Basic-PRM using sampling strategies with different measures: the uniform strategy, the two-phase connectivity expansion strategy [12], and the Gaussian strategy [2]. The last two strategies perform much better than the uniform one. How can such improvement be explained? What information can a PRM planner use to bias the sampling measure to its advantage?

If nothing is assumed on F , all hypotheses on the shape of F are equally likely. There is no reason to sample one region of C more densely than another, and the uniform sampling measure is the best that a PRM planner can use. More generally, with no prior assumptions, there is little that we can say about the expected performance of PRM planners. If we persist in using PRM planners, the reason must be that F is expected to satisfy certain favorable properties. Note here the analogy with the theory of PAC learning, where one can expect to learn a concept from examples only if the concept is assumed to have a simple representation. Similarly, *we can expect a PRM planner to work well – i.e., to “learn” the shape of F from sampled configurations – only if we assume that F satisfies favorable visibility properties, which allow it to be adequately represented by a small roadmap.*

Now, if F is expansive, can non-uniform sampling measures work better than the uniform one? Since visibility properties are not uniformly favorable over F , a PRM planner should exploit the partial knowledge acquired during roadmap construction to identify regions with poor visibility properties and adjust the probability measure to sample these regions more densely. Now not only is the sampling measure non-uniform over F , but also it changes over time. In each sampling operation, the optimal measure is the one that minimizes the expected number of remaining sampling operations needed to reach a good roadmap.

The problem of constructing good sampling measures is still poorly understood. Existing strategies mostly rely on simple, heuristic estimates of visibility properties, for instance:

- The *two-phase connectivity expansion strategy* [12] builds an initial roadmap by sampling C uniformly at random. While doing so, it identifies the nodes that frequently fail to connect to other nodes nearby. Then the strategy samples more configurations around these identified nodes. The final distribution of sampled configurations is denser in regions having poor visibility. See the circled region in Fig. 5a around the corridor.
- In each sampling operation, the *Gaussian strategy* [2] samples a pair of configurations, whose distance between them is chosen according to the Gaussian measure. If exactly one configuration lies in F , this configuration is retained as a roadmap node. Otherwise, both configurations are discarded. This strategy yields a distribution of sampled configurations that is denser near the boundary of F (Fig. 5b). The rationale is that points inside narrow passages, which have poor visibility, often lie near the boundary. Focusing on the boundary may increase the sampling density inside narrow passages.

Fig. 4 shows that these two strategies are effective in exploiting the non-uniformity of visibility properties in F . When the corridor width is small, regions near the corridor have poor visibility, and the non-uniform strategies achieve huge speedup over the uniform one. As the corridor width increases, visibility properties become more uniformly favorable. The benefit of non-uniform sampling then decreases.

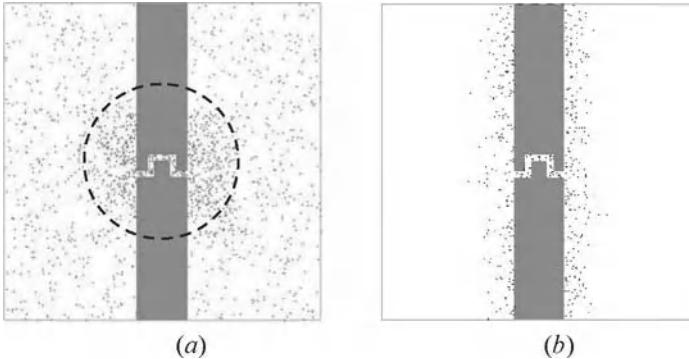


Fig. 5. Sampled configurations generated by (a) the two-phase connectivity expansion strategy and (b) the Gaussian strategy.

The above two non-uniform strategies are chosen here only for illustration. Other strategies have been proposed, and some of them achieve even greater speedup. They use various techniques to increase sampling density in subsets of F expected to have poor visibility. For instance, the bridge test extends the Gaussian strategy and samples three configurations, instead of two, to better identify narrow passages [7]. Other techniques identify narrow passages in a robot's workspace (e.g., by computing the medial axis) and use this information to sample more densely in regions of F likely to contain narrow passages [4, 5, 20]. For a robot manipulator arm, it has been shown that over-sampling near singular configurations improves performance [14]. Indeed, at a singular configuration q_s , the arm's end-effector loses some degrees of freedom. Thus the region of F near q_s has a flattened shape, resulting in poor visibility. Instead of using heuristics to locate regions with poor visibility, an alternative is to check directly the definition of visibility to prune a roadmap and avoid wasting effort in regions with good visibility [19], but this may involve high computational cost. A quite different approach is to slightly dilate F [8, 18]. As visibility in dilated F is more favorable, planning becomes easier. A path found in the dilated space is then deformed into one in F .

5 How Important Is the Sampling Source?

We have mentioned in Section 2 that a sampling strategy (π, S) is characterized by a probability measure π and a source S . The most commonly used source in PRM planning is the pseudo-random source S_{ran} . Given a fixed seed, S_{ran} generates a sequence of numbers that closely approximate the statistical properties of true random numbers. In particular, a pseudo-random sequence is slightly irregular to

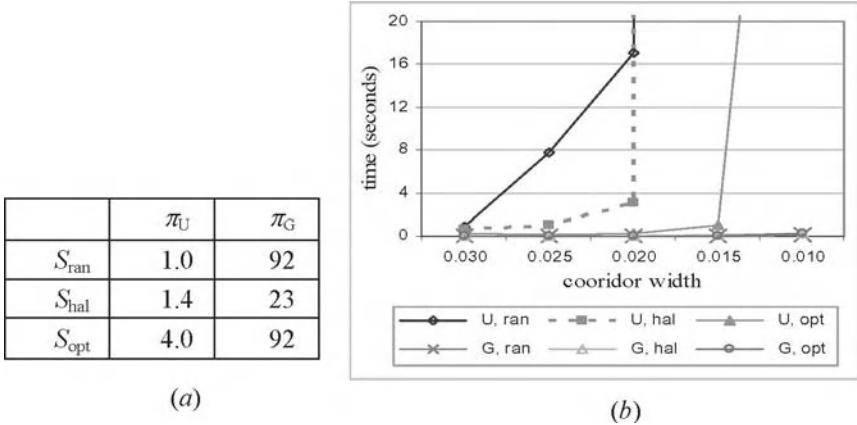


Fig. 6. Comparison of six sampling strategies on the problem of Fig. 3 when (a) the corridor width is set to 0.03 and (b) the width decreases.

simulate the effect that each number is chosen independently. Note that if we fix the seed of a pseudo-random source, the numbers generated are in fact deterministic. To get multiple independent runs of a PRM planner, we must use a different seed for each run. In the proofs of Theorems 1–3, this independence guarantees that samples spread evenly over F according to the uniform measure. However, deterministic sources can achieve the same goal, sometimes even better [13]. A familiar deterministic source is a grid. In this section, we compare pseudo-random and deterministic sources. We also compare the impact of sampling sources with that of sampling measures on the overall efficiency of PRM planning.

In our experiments, we use a pseudo-random source S_{ran} as well as two deterministic sources, the Halton sequence S_{hal} [17] and the incremental discrepancy-optimal sequence S_{opt} [15], both of which have been reported to often outperform S_{ran} [6, 13, 15]. We then pair each source with two probability measures, the uniform measure π_U and the measure π_G used in the Gaussian strategy. This leads to six sampling strategies $\{\pi_U, \pi_G\} \times \{S_{\text{ran}}, S_{\text{hal}}, S_{\text{opt}}\}$, each embedded in a distinct version of `Basic-PRM` to be tested experimentally.

- **The sampling measure versus the sampling source.** Fig. 6a compares the six strategies on the example in Fig. 3, when the corridor width is set to 0.03. Each table entry gives the ratio of the running time of the uniform random strategy (π_U, S_{ran}) versus that of the strategy of the entry. So, the table reports the *speedup* over (π_U, S_{ran}) . The running times for (π_U, S_{ran}) and (π_G, S_{ran}) are averaged over 30 independent runs. The second column (π_U) shows that S_{hal} and S_{opt} indeed achieve some speedup over S_{ran} , but far greater speedup is achieved by switching to π_G . Furthermore, the advantage of S_{hal} and S_{opt} over S_{ran} observed with π_U vanishes when we switch to π_G . These results are reinforced in Fig. 6b, which plots the running times of the six strategies, as the corridor width decreases. The three

	π_U	π_G
S_{ran}	1.0	40.3
S_{hal}	3.9	33.2
S_{opt}	0.9	42.2

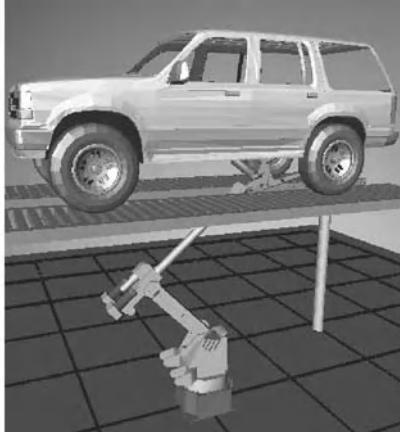


Fig. 7. Comparison of six sampling strategies on a more realistic problem.

curves bundled together at the bottom of the plot all correspond to strategies using π_G , demonstrating the importance of the sampling measure on the overall efficiency of the planner. Similar results have been obtained on more realistic problems, *e.g.*, the one in Fig. 7, in which a 6-degrees-of-freedom robot manipulator needs to access the bottom of a car through the narrow space between the lift supports.

- **Dependence on dimensionality.** The main basis for deterministic sources is that they minimize criteria such as discrepancy or dispersion. However, the computational cost of maintaining a fixed discrepancy or dispersion increases *exponentially* with $\dim(C)$ [17]. The samples generated by a deterministic source distribute evenly and regularly over $[0,1]^{\dim(C)}$, and so they roughly correspond to a grid with $N^{1/\dim(C)}$ discretized intervals per axis, where N is the number of samples. In typical PRM planning problems, N is relatively small, while $\dim(C)$ could be large (greater than 6). This leads to large discrepancy and dispersion, even when a deterministic source is used. Hence, the advantage that deterministic sources can possibly achieve over pseudo-random sources fades away as $\dim(C)$ increases. Fig. 8 gives an example, showing the running times of the six strategies as $\dim(C)$ increases from 3 to 8. The robot is a planar linkage with a mobile base. We increase $\dim(C)$ by adding more links. Fig. 8 shows that the running time of (π_U, S_{opt}) increases quickly with $\dim(C)$. The increase is slower with (π_U, S_{hal}) and even slower with (π_U, S_{ran}) . It is interesting to observe that (π_U, S_{hal}) performs slightly better than (π_U, S_{ran}) when $\dim(C) \leq 6$, but worsens afterwards (see the inset in the plot). The three strategies using π_G all have only moderate increases in running times. As $\dim(C)$ increases, visibility properties become less uniformly favorable over F , and the advantage of π_G over π_U grows.

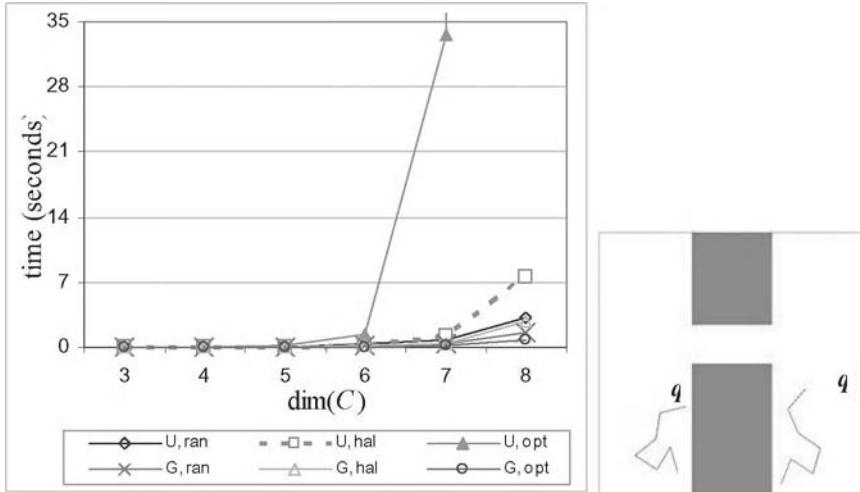


Fig. 8. The dependence of six sampling strategies on $\text{dim}(C)$. The inset in the left plot is a zoom of the lower portions of the curves.

6 Conclusion

The success of PRM planning depends mainly and critically on the assumption that, in practice, free spaces often verify favorable visibility properties. Non-uniform sampling measures dramatically improve the efficiency of PRM planning by exploiting these properties. In contrast, the choice of sampling sources has only small impact.

To speed up PRM planning, one promising research direction is to design better sampling strategies (and perhaps connection strategies as well) by exploiting the partial knowledge acquired during roadmap construction to adjust the sampling measure π continuously. Initial work along this line has appeared recently [1, 10]. In [1], an approximate model of the configuration space is built and used to sample configurations so that the expected value of a utility function is maximized. A crucial issue here is to define a utility function that closely approximates the expected cost of reaching a good roadmap. In [10], the sampling measure π is constructed as a linearly weighted combination of component measures with complementary strengths. To adjust π , the weights are updated after each sampling operation during roadmap construction to favor the component measures that give the most promising results. An important issue here is then to develop good criteria to assess the performance of component measures.

Acknowledgements. D. Hsu is supported by NUS grant R252-000-145-112. J.C. Latombe is supported by NSF grants ACI-0205671, IIS-0412884, and DMS-0443939, and NIH grant 5R33 LM007295. H. Kurniawati is supported an NUS Postgraduate Scholarship. We thank T. Bretl, L. Kavraki, M. Saha, and G. Sánchez-Ante for their comments on early drafts of this paper. We also thank S. Lindemann and S. LaValle for making available the code for generating deterministic low-discrepancy sequences.

References

1. B. Burns and O. Brock. Sampling-based motion planning using predictive models. In *Proc. IEEE Int. Conf. on Robotics & Automation*, 2005.
2. V. Boor, M.H. Overmars, and A.F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. *Proc. IEEE Int. Conf. on Robotics & Automation*, pp. 1018-1023, 1999.
3. H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*, Chapter 7, MIT Press, 2005.
4. M. Foskey, M. Garber, M.C. Lin, and D. Manocha. A Voronoi-based hybrid motion planner. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 55–60, 2001.
5. L. Guibas, C. Holleman, and L.E. Kavraki. A probabilistic roadmap planner for flexible objects with a workspace medial-axis based sampling approach. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 254–260, 1999.
6. R. Geraerts and M.H. Overmars. A comparative study of probabilistic roadmap planners. In J.D. Boissonnat et al. (eds.), *Algorithmic Foundations of Robotics V*, pp. 43-57, Springer, 2004.
7. D. Hsu, T. Jiang, J. Reif, and Z. Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 4420–4426, 2003.
8. D. Hsu, L.E. Kavraki, J.C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In P.K. Agarwal et al., editors, *Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics*, pages 141–154. A. K. Peters, Wellesley, MA, 1998.
9. D. Hsu, J.C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pp. 2219–2226, 1997.
10. D. Hsu, G. Sánchez-Ante, and, Z. Sun. Hybrid PRM sampling with a cost-sensitive adaptive strategy. In *Proc. IEEE Int. Conf. on Robotics & Automation*, 2005.
11. L.E. Kavraki, J.C. Latombe, R. Motwani, and P. Raghavan. Randomized query processing in robot motion planning. In *Proc. ACM Symp. on Theory of Computing (STOC)*, pp. 353-362, 1995.

12. L.E. Kavraki, P. Švestka, J.C. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. on Robotics & Automation*, 12:566–580, 1996.
13. S.M. LaValle, M.S. Branicky, and S.R. Lindemann. On the relationship between classical grid search and probabilistic roadmaps. *Int. J. of Robotics Research*, 23(7-8):673-692, 2004.
14. P. Leven and S. Hutchinson. Using manipulability to bias sampling during the construction of probabilistic roadmaps. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 2134–2140, 2002.
15. S.R. Lindemann and S.M. LaValle. Incremental low-discrepancy lattice methods for motion planning. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 2920–2927, 2003.
16. M. Lin and D. Manocha. Collision and proximity queries. In J.E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 35. CRC Press, 2004.
17. J. Matousek. *Geometric Discrepancy*. Springer-Verlag, Berlin, 1999.
18. M. Saha and J.C. Latombe. Finding narrow passages with probabilistic roadmaps: The small step retraction method. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2005.
19. T. Siméon, J.P. Laumond, and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. *J. of Advanced Robotics*, 14(6):477-494, 2000.
20. Y. Yang and O. Brock. Adapting the Sampling Distribution in PRM Planners Based on an Approximated Medial Axis. In *Proc. IEEE Int. Conf. on Robotics & Automation*, 2004.

Session Overview

Humanoids

Hirohisa Hirukawa

National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1 Umezno, Tsukuba 305-8568 Japan

An epoch of humanoid robotics started from the astonishing reveal of Honda P2 in 1996, and the focus of interest in the field has been the motion control of humanoid robots as well as the development of the hardware in the beginning of the decade. A reliable hardware with the minimum level of the mobility can be a research platform of humanoid robotics as well as mobile robot platforms like Nomad. Several research platforms are available currently including HRP-2 with software platform OpenHRP and HOAP series, and the interests in humanoid robotics can spread over various topics; that is, intelligence, interactions with humans and a tool of cognitive science. The state of the art of humanoid robotics has arrived at the level of the beginning of mobile robot technologies in 1980s, and every aspect of robotics is now expected to be integrated on humanoid robots.

The paper by Satoshi Kagami et al. extends the autonomy of humanoid robots. They developed an enhanced version of humanoid robot HRP-2, called HRP2-DHRC, equipped with three d.o.f. hands, three d.o.f. wrists, one d.o.f. toes, higher resolution stereo cameras and laser range fingers. The autonomy embedded on the robot includes a footstep planning with mixed reality with an online motion capture system, a vision guided footstep planning, an object localization from a depth matching, a navigation from 3D localization, and that among movable obstacles. HRP2-DHRC should be one of most advanced humanoid robots from the viewpoint of autonomy.

The paper by Hiroshi Ishiguro proposes android science as a new cross-interdisciplinary framework. He found that the appearance of the robot should have a significant influence on the impression of a humanoid robot as well as its behaviors. He developed humanoid robots that look like humans and executed a cognitive experiment in which subjects are asked to judge if a figure should be an android or a real human in two seconds. The result of the experiment told that the subjects should judge the figure is a real human in more chance when the figure has a real appearance with some human-like behavior. We had intensive discussions on his talk, especially on the significance of his work. It

was claimed out that the objective of the research may not be clarified, but Ishiguro tried to defend his approach.

The paper by Yoshihiro Nakamura et al. investigated the interaction between a human and a humanoid. They introduced a meta proto-symbol which is an abstract analogy of the proto-symbol. The meta proto-symbol is applied to recognize and generate the relationship of a human and a humanoid. They applied the proposed concept to a fight between a humanoid robot and a subject in a virtual world. The robot was able to recognize the human behaviors and generate the responses through mimetic communications with the human.

The papers contributed to the session showed the three directions in which humanoid robotics should be enriched in the coming decade, that is, autonomy, cognitive science and interaction with humans.

Humanoid HRP2-DHRC for Autonomous and Interactive Behavior

S. Kagami^{1,2}, K. Nishiwaki^{1,2}, J. Kuffner^{3,1}, S. Thompson¹, J. Chestnutt³, M. Stilman³, and P. Michel³

¹ Digital Human Research Center, AIST. s.kagami@aist.go.jp

² CREST Program, JST.

³ Robotics Institute, Carnegie-Mellon University.

1 Introduction

Recently, research on humanoid-type robots has become increasingly active, and a broad array of fundamental issues are under investigation. However, in order to achieve a humanoid robot which can operate in human environments, not only the fundamental components themselves, but also the successful integration of these components will be required. At present, almost all humanoid robots that have been developed have been designed for bipedal locomotion experiments. In order to satisfy the functional demands of locomotion as well as high-level behaviors, humanoid robots require good mechanical design, hardware, and software which can support the integration of tactile sensing, visual perception, and motor control. Autonomous behaviors are currently still very primitive for humanoid-type robots. It is difficult to conduct research on high-level autonomy and intelligence in humanoids due to the development and maintenance costs of the hardware. We believe low-level autonomous functions will be required in order to conduct research on higher-level autonomous behaviors for humanoids.

This paper describes our research efforts aimed at developing low-level autonomous capabilities required for moving & manipulation tasks involving humanoid-type robots. In that purpose, Humanoid HRP2-DHRC(Fig.1) is designed by improved from original HRP2[1] to have extra joints and sensors, and it is manufactured by Kawada Industries Inc. On this platform, sense-plan-act loop is implemented for autonomous moving & manipulation. Augmented reality based humanoid robot experiment system is also developed to help developing each functions.

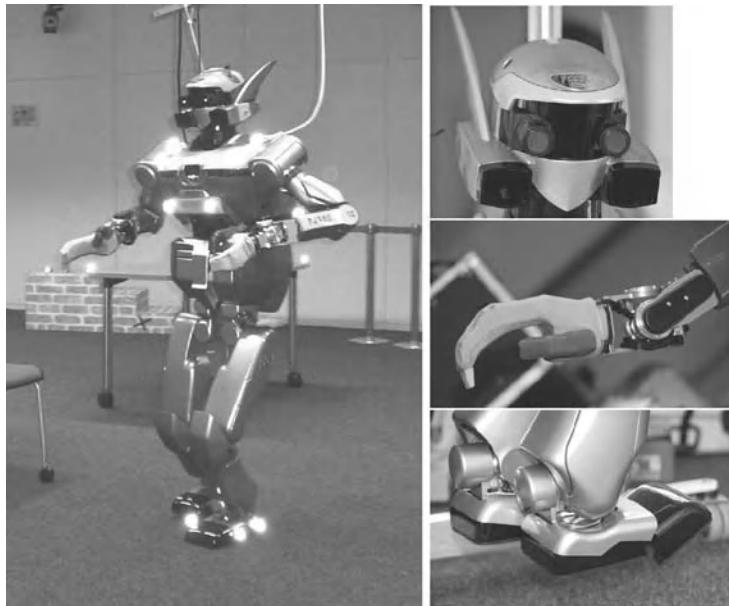


Fig. 1. Humanoid HRP2-DHRC

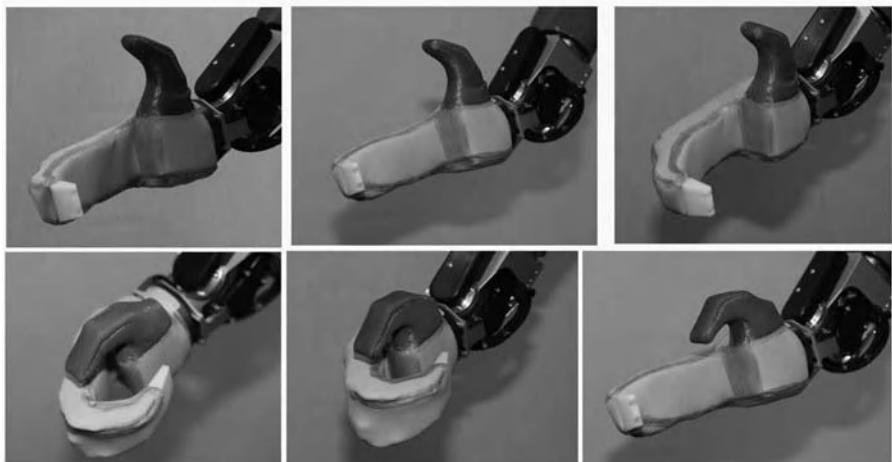


Fig. 2. Hand postures of HRP2-DHRC

2 Humanoid HRP2-DHRC Hardware Improvements

2.1 Additional Joints

Original HRP2 has 30 DOF in total (6 DOF for arm & leg, 2 DOF for neck & waist, 1 DOF for gripper). There are three part that joints added for HRP2-

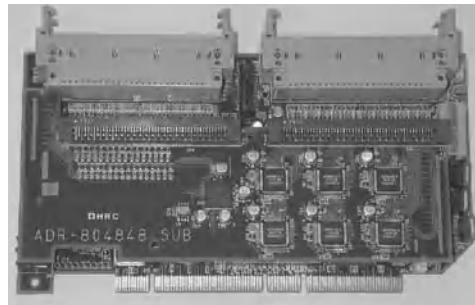


Fig. 3. DHRC-ADR804848: PCI half size IO board

DHRC: 1) wrist, 2) hand and 3) toe. HRP2-DHRC has 38 DOF in total and it is 158cm in height & 58kg in weight(Fig.1).

As for 1) arm, in order to increase high manipulability region, 1 DOF is added at wrist joint. As for 2) hand, 3 DOF hand that can grasp objects in several different ways is designed by Prof. Inoue and attached to Humanoid JSK-H7[2]. We adopted the same mechanisms to HRP2-DHRC(Fig.2). As for 3) foot, 1 DOF is added at toe in order to improve walking motion as like H7.

2.2 Control Board

An PCI I/O board is newly developed to achieve current sensor based torque control(Fig.3). The board has 80ch AD(14bit), 48ch DA(12bit) & encoder counter, 32ch DIO. It can achieve up to 10khz sampling for all input/output usage situation (up to 48 joints) by using DMA data transmission through 32bit/64bit PCI bus. Current sensor signal of motor driver is connected to AD input, so that torque control is possible. Board has almost 1 slot PCI half size.

2.3 Head Sensor

Original HRP2 has three synchronized mono cameras at head. Firewire (IEEE1394) stereo camera Videre design STH-DCGS is adopted together with time of flight type laser range sensor Hokuyo URG-04LX at head. Videre design STH-DCGS has global shuttered VGA stereo camera and has about 90 degrees view angle in horizontal.

URG-04LX is a small (160g) range sensor that measures up to 4m and covers 270 degrees in 0.36 degrees resolution.

2.4 Experimental Sensor

Foot force distribution sensor and HD resolution stereo camera are under developing functions for humanoid robots.

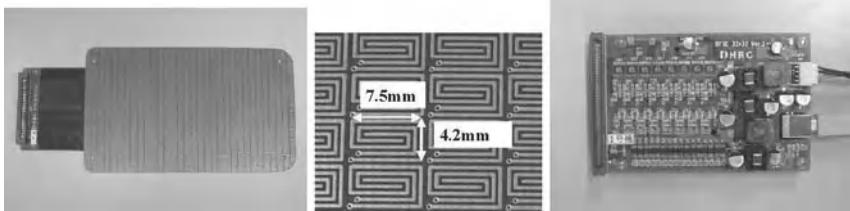


Fig. 4. Electrode part of the sensor, its sensing area close up and IO board

Foot Force Distribution Sensor

Scanning Circuit

A 32×32 matrix scan type high-speed pressure sensor for the feet of humanoid robots that has 1[kHz] sampling rate is developed(Fig.4). This sensor has matrix scan circuit. The matrix scan method has a problem of interference by bypass current. In order to resolve this problem, a novel method is proposed.

We adopted very thin(0.6[mm]) force sensing conductive rubber sheet for high speed sensing. Each sensing area is 4.2×7.0 [mm] and can measure vertical force of approximately 0.25–20[N]. Walking cycle of humanoid robot as well as human being is about 0.4–0.8[s] and dual leg phase is about 0.1–0.15[s]. The target of the sensor is biped walk stabilization so that high-speed input is important. Matrix scan type circuit is connected to sensor, and the system runs 1[kHz] with 14[bit] resolution at 4.2×7.0 [mm] grid for 32×32 points, and the sensor size is the same as humanoid robot foot 135×228 [mm]. The system is running high-speed because of very thin conductive rubber and simultaneous measurement.

Electrodes which are shown in Fig.4 are arranged in the shape of a grid. There is a flexible cable part which has connector at the left side of electrode part to avoid collision to the ground and robot itself is important. The control circuit board is attached to the shank link of our humanoid robot H7, and only USB2 cable goes through the joints to the controlling PC mounted on the torso.

Thin force sensing rubber

Thickness of developed force sensing rubber is 0.6[mm] (Inaba Industries Inc.). Conductive carbon composite grain are mixed in the rubber. Thinness is better to achieve small time constant and sensitivity, so that system can realize higher scan rate. Table.1 shows a specification of our rubber sheet.

With no load, resistance on a surface and on a volume are both about $10^7[\Omega]$. As the pressure is exerted, rubber deforms and conductive path augments, so that relationship in between pressure and resistance changes monotonic and smooth.

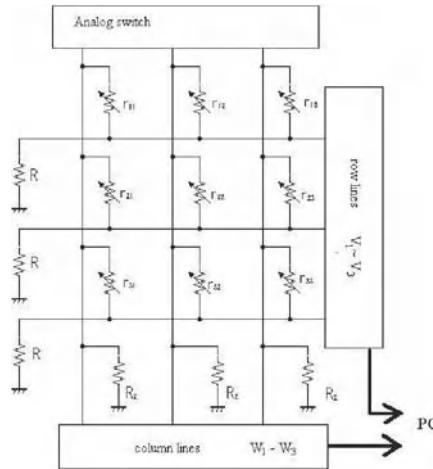


Fig. 5. Pressure sensor circuit diagram. (Example 3x3 matrix)

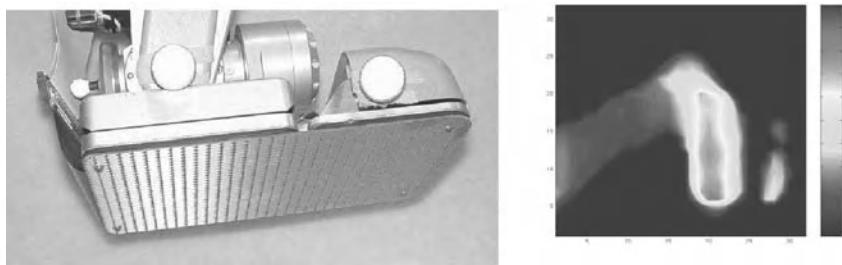


Fig. 6. Humanoid robot foot with pressure sensor grid and human foot pressure result

The usual matrix scan method is equipped with AD converters on the column lines. The system in this paper is equipped AD converters both on the column lines and row lines of the sensor matrix(Fig.5). When the column line 1 is applied the voltage, the following formula is led from Kirchhoff's current rule at the top row:

$$\frac{^1V_1}{R} = \sum_j \frac{^1W_j - ^1V_1}{r_{1j}}.$$

Similarly, the following formula is led also about the i -th row and k -th column line:

Table 1. Force Sensing Rubber Resister

Max. Voltage	30V
Recomm. Voltage	6V
Max. Current	20mA
Recomm. Current	5mA
No-load Resistance	20MΩ
Maximum Load	100N
Recomm. Load	40N

$$\frac{{}^k V_i}{R} = \sum_j \frac{{}^k W_j - {}^k V_i}{r_{ij}}.$$

Let x_{ij} be $\frac{1}{r_{ij}}$:

$$\frac{{}^k V_i}{R} = \sum_j ({}^k W_j - {}^k V_i) \cdot x_{ij}.$$

This formula means simultaneous equation. When $i = 1$, the formula is expressed with the following matrix:

$$\mathbf{V}_1 = M_1 \mathbf{X}_1.$$

Here, $\mathbf{V}_1, M_1, \mathbf{X}_1$ are as follows:

$$\begin{aligned} \mathbf{V}_1 &= \begin{pmatrix} {}^1 V_1 / R \\ \vdots \\ {}^n V_1 / R \end{pmatrix}, \\ M_1 &= \begin{pmatrix} {}^1 W_1 - {}^1 V_1 & \cdots & {}^1 W_n - {}^1 V_1 \\ \vdots & \ddots & \vdots \\ {}^n W_1 - {}^n V_1 & \cdots & {}^n W_n - {}^n V_1 \end{pmatrix}, \\ \mathbf{X}_1 &= \begin{pmatrix} x_{11} \\ \vdots \\ x_{1n} \end{pmatrix}. \end{aligned}$$

Therefore \mathbf{X}_1 is led by using inverse matrix of M :

$$\mathbf{X}_1 = M_1^{-1} \mathbf{V}_1.$$

$$\mathbf{X}_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1n} \end{pmatrix} = \begin{pmatrix} 1/r_{11} \\ \vdots \\ 1/r_{1n} \end{pmatrix}$$

Similarly, i -th \mathbf{X}_i is calculated.

$$\mathbf{X}_i = M_i^{-1} \mathbf{V}_i = \begin{pmatrix} 1/r_{i1} \\ \vdots \\ 1/r_{in} \end{pmatrix}$$

Value of resistance r_{ij} is found by taking the reciprocal of each element of vector \mathbf{X}_i .

Experimental results

Dynamic pressure was applied to the sensor(Fig.6(right)). As the dynamic pressure, the subject(male, weight:65kg, foot size:27cm) run on the sensor. Scanning rate was 300[Hz].

The pressure sensor is attached to our humanoid robot H7 foot(Fig.6). Distributed pressure is measured and COP trajectory is calculated.

Developed thin force sensing conductive resistance rubber has about 1500-1[Ω] at 0.25-20[N]. Matrix scan is achieved with a novel method. Resistance at each sensing point is calculated by solving the simultaneous equations from column and row lines voltage. Interference by bypass current is suppressed by this method. The result of load and output voltage is monotonic, and doesn't have large hysteresis. The high-speed(1[kHz]) sensor was realized by measuring voltage simultaneously and thin(0.6[mm]) force sensing conductive rubber.

HD Stereo Camera

Humanoid vision sometimes requires to have multi-resolution or zooming function. For example, avoiding obstacles, looking for a path to given goal, detecting human posture, such tasks requires to have wide view angle. However, finding face, detecting grasping position of target object, measuring distance to next step, such tasks requires to have narrow view angle to measure precise accuracy.

There exists several humanoid systems that have two (or more) stereo camera sets which has different lenses. However, alignment of those multiple stereo camera causes difficult problem. Also data bus speed is another limitation.

In order to overcome these problem, we developed stereo camera that has HD resolution CMOS, and simultaneously captures use whole image (but subsampled) and dense image at desired position (such as center) (Fig.7).

Imager is Altasens ProCamHD3560 (2/3" CMOS) that has 1920×1080 in 60P global shutter. We also developed C-mount HD resolution lens of $f=4.8\text{mm}$ (about 90deg). This lens has HD resolution at fringe. This camera is connected to PC by using USB2 bus. Bandwidth of USB2 bus is not sufficient to handle HD 60P raw color image. Therefore, we prepare dual CIF/VGA



Fig. 7. HD stereo camera, whole image, and 320x240 left upper corner image

resolution stereo mode that captures two stereo pair of 1) full screen (sub-sampled) and 2) dense partial image (about 15 degrees). In CIF mode, the camera achieves 60Hz capturing. The camera size is 195x85x65[mm], weight is 225[g] and consumes about 10[W].

3 Augmented Reality Based Development System

In order to develop more sophisticated autonomous humanoid behaviors, thorough testing of various interconnected hardware and software components for sense, plan and control becomes increasingly difficult. Many software tools are available for dynamic simulation and visualization in simulation stage. However, when robots are put to the test in real environments these tools are only used offline for processing the data of an experiment. We encountered difficulty to achieve real-world autonomy even after developing each sense-plan-act functions. There are problems such as follows: a) perception error (accuracy, repeatability) is hard to examine because of the lack of global information such as relationship between robot and environment, b) planning and control software error caused by particular perception are hard to find because of lack of repeatability, c) planning and control software tuning are also difficult.

We propose an alternate paradigm for real-world experimentation that utilizes a real-time optical tracking system to form a complete hybrid real/virtual testing environment.

Our proposed system has two objectives: to present the researcher with a ground truth model of the world and to introduce virtual objects into actual real world experiments. Conceptually it is real bi-directional augmented reality.

To see the relevance of these tools, consider an example of how the proposed system is used in our laboratory. A humanoid robot with algorithms for vision, path planning and ZMP stabilization is given the task of navigation in a field of obstacles. During an online experiment, the robot unexpectedly contacts one of the obstacles. Did our vision system properly construct a model of the environment? Did the navigation planner find an erroneous path? Was our controller properly following the desired trajectory? A ground truth

model helps resolve ambiguities regarding the source of experimental failures by precisely identifying the locations of the obstacles and the robot. Just as in simulation, we can immediately determine whether the vision algorithm identified the model, or whether the controller followed the trajectory designed by the planner. In some cases, we can avoid the undesired interaction entirely. Having established a correspondence between virtual components such as environment models, plans, intended robot actions and the real world, we can then visualize and identify system errors prior to their occurrence.

In this section, we describe the implementation of the hybrid experimental environment. We develop tools for constructing a correspondence between real and virtual worlds. Using these tools we find substantial opportunities for experimentation by introducing virtual obstacles, virtual sensors and virtual robots into a real world environment. We describe how adding such objects to an experimental setting aids in the development and thorough testing of vision, planning and control[3].

3.1 System Configuration

To construct a hybrid real/virtual environment, we instrumented our lab space with the Eagle-4 Motion Analysis motion capture system. The environment also contains cameras and furniture objects. Our experiments focused on high level autonomous tasks for the humanoid robot HRP-2. For instance, the robot navigated the environment while choosing foot locations to avoid obstacles and manipulated obstacles to free its path. We partitioned these experiments according to the subsystems of vision, planning and control to provide a general groundwork for how a hybrid real/virtual testing environment can be used in a larger context of research objectives.

The Eagle-4 system consists of 12 cameras, covering a space of $5 \times 5 \times 1.8$ meters. Distances between markers that appear in this space can be calculated to 0.3% accuracy. In our experiments, the motion capture estimate of the distance between two markers at an actual distance of 300mm has less than 1mm error.

In terms of processing speed, we employ a dual Xeon 3.6GHz processor computer to collect the motion capture information. The EVa Real-Time Software (EVaRT) registers and locates 3D markers at maximum rate of 480Hz with an image resolution of 1280×1024 . When larger numbers of markers are present, the maximum update speed decreases. Still, when tracking approximately 60 markers the lowest acquisition rate we used was 60Hz. Marker localization was always performed in real-time.

EVaRT groups the markers attached to an object. We refer to this set of points as the object template. Under the assumption that a group of markers is attached to a rigid object, any displacement of the object corresponds to a rigid transformation T of the markers.

During online execution, EVaRT uses distance comparisons to identify groupings of markers, as well as the identities of markers in these groupings.



Fig. 8. (a) Real chair with retroreflective markers illuminated. (b) 3D model of chair as recovered by a laser scanner. (c) Virtual chair is overlaid in real-time. Both the chair and the camera are in motion.

We are then interested in the inverse problem of finding a transform T that aligns the template marker locations with those found in the scene by motion capture.

3.2 Geometry Reconstruction

The transformation of a rigid body's coordinate frame tells us the displacement of all points associated with the body. To reconstruct the geometry of a scene, we need to establish the geometry of each object in its local reference frame.

In our work, we have chosen to use 3D triangular surface meshes to represent environment objects. We constructed preliminary meshes using a Minolta VIVID 910 non-contact 3D laser digitizer. The meshes were manually edited for holes and automatically simplified to reduce the number of vertices.

Fig.8 demonstrates the correspondence between a chair in the lab environment and its 3D mesh in our visualization. We are able to continuously re-compute the transformation of a lab object at a rate of 30Hz. The virtual environment can then be updated in real-time to provide a visualization of the actual object's motion in the lab.

3.3 Real and Virtual Cameras

In this section we consider the latter case of placing a camera in the viewable range of motion capture. We show that tracking a camera lets us to establish a correspondence between objects in the ground truth model and objects in the camera frustum.

As with other rigid bodies, the camera is outfitted with retro-reflective markers that are grouped in EVaRT and then tracked using our algorithm. The position and orientation of the camera computed from motion capture form the extrinsic camera parameters. The translation vector t corresponds

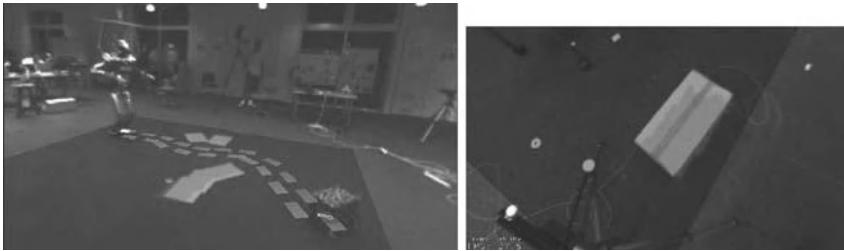


Fig. 9. Environment reconstructions overlaid onto the world. (a) Occupancy grid generated from image-based reconstruction using the robot's camera. (b) planar projection of an obstacle recovered from range data.

to the world coordinates of the camera's optical center and the 3×3 rotation matrix R represents the direction of the optical axis. Offline camera calibration using Tsai's camera model is performed once to recover the the 3×3 upper triangular intrinsic parameter matrix K . Incoming camera images can then be rectified on the fly. The extrinsic and intrinsic parameters allow us to recover the full camera projection matrix M . M uniquely maps a scene point $P = (x, y, z, 1)^T$ to a point on the image plane $p = (u, v, 1)^T$ via the standard homogeneous projection equation.

Therefore, we can recover not only the locations of motion capture markers but also any points that compose the surface mesh of a tracked object.

We can use existing 3D display technology such as OpenGL to efficiently compute surface models as they would appear in the camera projection. Overlaying the virtual display on the camera display creates the a correspondence between the camera view and the ground-truth motion capture view.

3.4 Examination of Humanoid Sensing

Given a representation of the robot environment reconstructed by image warping or from range data, we can visually evaluate the accuracy of our perception algorithms and make parameter adjustments on the fly by overlaying the environment maps generated back onto a camera view of the scene. This enables us to verify that obstacles and free space in our environment reconstructions line up with their real-world counterparts, as illustrated in Fig.9.

3.5 Examination of Humanoid Planning

Fig.9(left) and Fig.10 (left) show examples of control system visualization during online robot experiments. The system has planned out the sequence of footsteps it wishes to take to reach some goal configuration. For each step, it has computed the 3D position and orientation of the foot. Through the use of augmented reality, the planned footsteps can be overlaid in real-time onto the environment.



Fig. 10. Augmenting reality for visualization of planning and execution. (a) Foot-step plan displayed onto the world. (b) Augmented reality with a simulated robot amongst real obstacles.

The red and blue rectangles represent the steps for the right and left feet that the robot intends to take. This path is constantly updated as the robot replans while walking. This display helps expose the planning process to identify errors and gain insight into the performance of the algorithm.

Temporal Projection: Virtual Robot

One of the components of our overall system that we would like to replace for testing purposes is the robot itself. One solution to is to build a simulated environment for experimentation. However, we would like to continue to use the real world as much as possible, rather than using a completely fabricated environment. Within our framework, we can continue to use real-world obstacles and sensors, and merely replace the robot with a simulated avatar. Fig.10 (right) shows the augmented reality of our simulated robot traversing a real environment. Note that for this navigation task, the robot is not manipulating the environment. The obstacles themselves can be moved during the experiments, but we do not need to close the loop on robotic manipulation.

Objects and the Robot’s Perception

In addition to complete replacement of all sensing with perfect ground truth data, we can simulate varying degrees of realistic sensors. We can slowly increase the realism of the data which the system must handle. This approach can isolate specific sources of error, and determine to which the control system is most sensitive. For example, by knowing the locations and positions of all objects as well as the robot’s sensors, we can determine which objects are detectable by the robot at any given point in time. Hence, simulated sensors can be implemented with realistic limits and coverage.

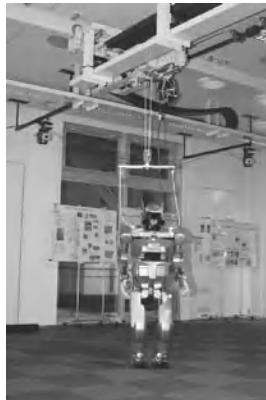


Fig. 11. Automatic following gantry & HRP2-DHRC with markers

3.6 Gantry

During any task of locomotion or manipulation, a humanoid robot is at risk of falling. Typically, a small gantry is used to closely follow and secure the robot. However, the physical presence of the gantry and its operator prevent us from testing fine manipulation or navigation that requires the close proximity of objects.

To overcome this problem, our laboratory implements a ceiling suspended gantry (10×7.5 [m]) that can follow the robot throughout the experimental space. It is controlled by standard PC with Timesys Linux realtime operating system (as like HRP2-DHRC). Having acquired the absolute positioning of the robot from motion capture, this gantry is PD controlled to follow the robot as it autonomously explores the space. This final component not only lets us to test the robot in arbitrary cluttered environments, but also enables experiments that typically require four or five operators to be safely performed by a single researcher.

4 Concluding Remarks

HRP2-DHRC humanoid robot is developed as a research platform for humanoid autonomy research (as like previous our H7 humanoid robot). Using HRP2-DHRC, we conducted research on sense-plan-act based humanoid autonomy as shown in Fig.12. As for “Sense” part, plane segmentation & 3D labeling[4], 6D visual odometry & world reconstruction[5], particle filter based localization[6], foot distributed force sensor[7] are studied. As for “Plan” part, footstep planning[8], arm motion planning by RRT[9], NAMO[10], manipulatibily maximization arm trajectory planning[11] are studied. As for

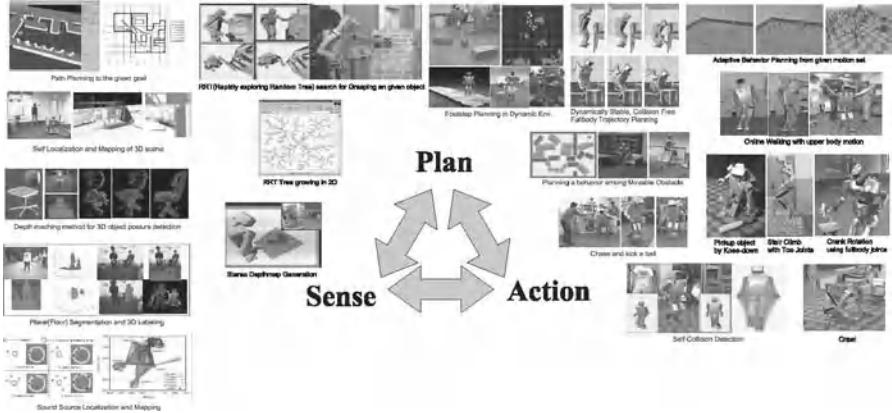


Fig. 12. Sense-Plan-Act Functions for HRP2-DHRC Low-level Autonomy

“Act” part, whole body cooperated reaching motion generation[12], whole body coordinated hand manipulation[13] are studied.

One fundamental achievement in this paper is augmented reality based development system. It is a novel experimental paradigm that leverages the recent advances in optical motion capture speed and accuracy to enable simultaneous online testing of complex robotic system components in a hybrid real-virtual world. We believe that this new approach enabled us to achieve rapid development and validation testing on each of the perception, planning, and control subsystems of our autonomous humanoid robot platform. We hope that this powerful combination of vision technology and robotics development will lead to faster realization of complex autonomous systems with a high degree of reliability.

References

1. H.Hirukawa, F.Kanehiro, K.Kaneko, S.Kajita, K.Fujiwara, Y.Kawai, F.Tomita, S.Hirai, K.Tanie, T.Isozumi, K.Akachi, T.Kawasaki, S.Ota, K.Yokoyama, H.Handa, Y.Fukase, J.Maeda, Y.Nakamura, and H.Inoue, “Humanoid Robotics Platforms Developed in HRP,” *Robotics and Autonomous Systems*, vol. 48, no. 4, pp. 165–175, October 2004.
2. S. Kagami, K. Nishiwaki, J. Kuffner, K. Okada, Y. Kuniyoshi, M. Inaba, and H. Inoue, “Low-level Autonomy of Remote Operated Humanoid Robot H6 & H7,” *Robotics Research*, 2003.
3. M. Stilman, P. Michel, J. Chestnutt, K. Nishiwaki, S. Kagami, and J. Kuffner, “Augmented reality for robot development and experimentation,” Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-55, November 2005.

4. Y. KIDA, S. KAGAMI, T. NAKATA, M. KOUCHI, and H. MIZOGUCHI, "Human Finding and Body Property Estimation by using Floor Segmentation and 3D Labelling," in *Proceedings of 2004 IEEE International Conference on Systems, Man & Cybernetics(SMC04)*, October 2004, pp. 2924–2929.
5. S. Kagami, Y. Takaoka, Y. Kida, K. Nishiwaki, and T. Kanade, "Online dense local 3d world reconstruction from stereo image sequences," in *Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS2005)*, August 2005, pp. 2999–3004.
6. S. Thompson and S. Kagami, "Humanoid robot localisation using stereo vision," in *Proceedings of 2005 5th IEEE-RAS International Conference on Humanoid Robots(Humanoids2005)*, December 2005, pp. 19–25.
7. Y.Takahashi, K.Nishiwaki, S.Kagami, H.Mizoguchi, and H.Inoue, "High-speed pressure sensor grid for humanoid robot foot," in *Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS2005)*, August 2005, pp. 1097–1102.
8. J. Chestnut, J. J. Kuffner, K. Nishiwaki, and S. Kagami, "Planning biped navigation strategies in complex environments," in *IEEE International Conference on Humanoid Robots (Humanoids2003)*, 10 2003.
9. S.Kagami, J.J.Kuffner, K.Nishiwaki, K.Okada, M.Inaba, and H.Inoue, "Humanoid arm motion planning using stereo vision and rrt search," *Journal of Robotics and Mechatronics*, vol. 15, no. 2, pp. 200–207, 2003.
10. M. Stilman and J. Kuffner, "Navigation among movable obstacles: Real-time reasoning in complex environments," *International Journal of Humanoid Robotics*, vol. 2, no. 4, pp. 479–504, December 2005.
11. L. Guilamo, J. Kuffner, K. Nishiwaki, and S. Kagami, "Manipulability optimization for trajectory generation," in *Proceedings of 2006 IEEE International Conference on Robotics and Automation(ICRA2006)*, May 2006, pp. 2017–2022.
12. K. Nishiwaki, M. Kuga, S. Kagami, M. Inaba, and H. Inoue, "Whole-body cooperative balanced motion generation for reaching," pp. 437–457, 2005.
13. K. Nishiwaki, Y. Fukumoto, S. Kagami, M. Inaba, and H. Inoue, "Object manipulation by hand using whole-body motion coordination," in *Proceedings of the IEEE International Conference on Mechatronics & Automation(ICMA2005)*, August 2005, pp. 1778–1783.

Android Science

– Toward a New Cross-Interdisciplinary Framework –

Hiroshi Ishiguro

Department of Adaptive Machine Systems, Osaka University
ishiguro@ams.eng.osaka-u.ac.jp

1 Android Science

Appearance and Behavior

In the evaluation of interactive robots, the performance measures are subjective impression of human subjects who interact with the robot and their unconscious reactions, such as synchronized human behaviors in the interactions and eye movements.

Obviously, both the appearance and behavior of the robots are important factors in this evaluation. There are many technical reports that compare robots with different behaviors. However nobody has focused on appearance in the previous robotics. There many empirical discussions on very simplified static robots, say dolls. Designing the robot's appearance, especially to give it a humanoid one, was always a role of the industrial designer. However we consider this to be a serious problem for developing and evaluating interactive robots. Appearance and behavior are tightly coupled with both each other and these problems, as the results of evaluation change with appearance. In our previous work, we developed several humanoids for communicating with people [3][4][5], as shown in Figure 1. We empirically know the effect of appearance is as significant as behaviors in communication. Human brain functions that recognize people support our empirical knowledge.

Android Science

To tackle the problem of appearance and behavior, two approaches are necessary: one from robotics and the other from cognitive science. The approach from robotics tries to build very humanlike robots based on knowledge from cognitive science. The approach from cognitive science uses the robot for verifying hypotheses for understanding humans. We call this cross-interdisciplinary framework *android science*.



Fig. 1. From humanoids to androids

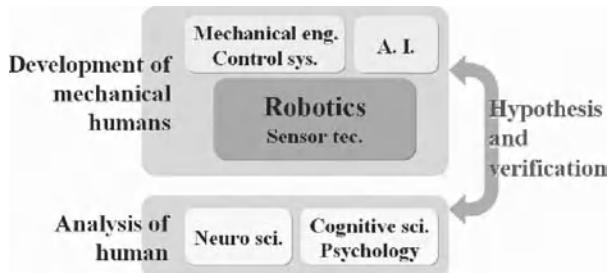


Fig. 2. The framework of android science

Previous robotics research also used knowledge of cognitive science while research in cognitive science utilized robots. However the contribution from robotics to cognitive science was not enough as robot-like robots were not sufficient as tools of cognitive science, because appearance and behavior cannot be separately handled. We expect this problem to be solved by using an android that has an identical appearance to a human. Robotics research utilizing hints from cognitive science also has a similar problem as it is difficult to clearly recognize whether the hints are given for just robot behaviors isolated from their appearance or for robots that have both the appearance and the behavior.

In the framework of android science, androids enable us to directly exchange knowledge between the development of androids in engineering and the understanding of humans in cognitive science. This conceptual paper discusses the android science from both viewing points of robotics and cognitive science.

2 Development of Androids

Very Humanlike Appearance

The main difference between robot-like robots and androids is appearance. The appearance of an android is realized by making a copy of an existing person.

The thickness of the silicon skin is 5mm in our trial manufacture. The mechanical parts, motors and sensors are covered with polyurethane and the silicon skin. Figure 3 shows the silicon skin, the inside mechanisms, the head part and the finished product of a child android made by painting colors on the silicon skin. As shown in the figure, the details are recreated very well so they cannot be distinguished from photographs of the real child.



Fig. 3. The silicon skin and inside mechanisms

Mechanisms for Humanlike Movements and Reactions

Very humanlike movement is another important factor for developing androids. For realizing humanlike movement, we developed an adult android because the child android is too small. Figure 4 shows this developed android. The android has 42 air actuators for the upper torso except fingers. We decided the positions of the actuators by analyzing movements of a real human using a precise 3D motion tracker. The actuators can represent unconscious movements of the chest from breathing in addition to conscious large movements of the head and arms. Furthermore, the android has a function for generating facial expression that is important for interactions with humans. Figure 5 shows several examples of facial expression. For this purpose, the android uses 13 of the 42 actuators.

The air actuator has several merits. First, it is very silent, much like a human. DC servomotors that require several reduction gears make un-humanlike noise. Second, the reaction of the android as against external force becomes very natural with the air dumper. If we use DC servomotors with reduction gears, they need sophisticated compliance control. This is also important for realizing safe interactions with the android.

The next issue is how to control the 42 air servo actuators for realizing very humanlike movements. The simplest approach is to directly send angular information to each joint by using a simple user interface termed a *motion editor*. However we need to specify 42 angles for creating a posture, which takes a long time. Therefore we added a function to generate smooth motions based on sinusoidal signals. This is the same idea as Perlin noise [8] used in computer graphics. This function helps especially well in making partial movements; however it is still time-consuming.

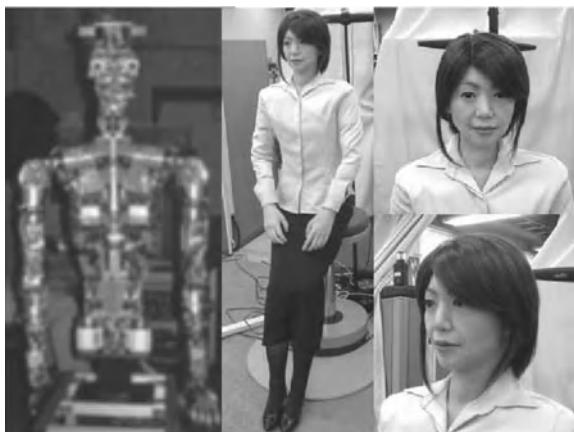


Fig. 4. Adult android developed in cooperation with Kokoro Co. Ltd.



Fig. 5. Facial expressions of the android

In addition to this problem, another difficulty is that the skin movement does not simply correspond to the joint movement. For example, the android has more than five actuators around the shoulder for humanlike shoulder

movements, with the skin moving and stretching according to the actuator motions. For solving this problem, a mapping table was required that correlates the surface movement to the actuator motions.

Our idea for solving this problem is to train a neural network. The neural network memorizes a mapping between actuator command patterns and marker 3D positions based on a large number of examples of android postures.

Toward Very Humanlike Movement

The next step after obtaining the mapping between the surface movements and actuators is implementing humanlike motions in the android. A straightforward approach for this challenge is to imitate real human motions in cooperation with the master of the android. By attaching markers of the precise 3D motion tracker on both the android and the master, the android can automatically follow human motions.

Humanlike Perception

The android requires humanlike perceptual abilities in addition to a humanlike appearance and movements. This problem has been tackled in computer vision and pattern recognition in rather controlled environments. However, the problem becomes seriously difficult when applied to the robot in other situations, as vision and audition become unstable and noisy.

Ubiquitous/distributed sensor systems solve this problem. The idea is to recognize the environment and human activities by using many distributed cameras, microphones, infrared motion sensors, floor sensors and ID tag readers in the environment. We have developed distributed vision systems [2] and distributed audition systems in our previous work. For solving this problem this work must be integrated and extended.

3 Cognitive Studies Using Androids

Total Turing Test

As discussed in the Introduction, android science has two aspects, the engineering approach and the scientific approach. The most vivid experiment where the two approaches meet is the *total Turing test*. The original was devised to evaluate the intelligence of computers under the assumption that mental capacities could be abstracted from embodiment [10]. The approach invoked many questions about the nature of intelligence. We consider intelligence as subjective phenomena among humans or between humans and robots. Obviously, the original Turing test does not cover the concept of total intelligence [1]. In contrast, the android enables us to evaluate total intelligence.

As did the original Turing test, the Total Turing test uses a time competition. We have checked how many people in preliminary experiments do not become aware within 2 sec. that they are dealing with an android. Figure 6 displays the scene. A task is given to the subject to find the colors of the cloth. The screen between the android and the subject opens for 2 sec. The subject then identifies the color. At the same time, the subject is asked whether he/she became aware the other is an android. We have prepared two types of android, one a static android and the other an android with the micro movements we call unconscious movements. Because a human does not freeze, he/she is always slightly moving even when not doing anything, such as just sitting on a chair.



Fig. 6. Total Turing test

As the result of the experiment with 20 subjects, 70% of the subjects did not become aware they were dealing with an android when the android had micro movements, but 70% became aware with the static android. This result shows the importance of the micro movements for the appearance of humanlike reality.

The 2-second experiment does not mean the android has passed the total Turing test. Nevertheless, it shows significant possibilities for the android itself and for cross-interdisciplinary studies between engineering and cognitive science.

Uncanny Valley

Why do 30% of the subjects become aware of the android? What happens if the time is longer than 2 sec.? In the experiment, the subjects felt a certain strangeness about the android's movements and appearance. Mori [7] predicted that as robots appear more human, they seem more familiar, until a point is reached at which subtle imperfections create a sensation of strangeness as shown in Figure 7. He referred to this as the *uncanny valley*.

Extension of the Uncanny Valley

Why does this uncanny valley exist? We have two hypotheses:

- If its appearance is very humanlike, the subject attempts to understand the android as being human. Therefore the subtle difference creates a strong strangeness as the uncanny valley.
- Humans expect balance between appearance and behaviors when they recognize creatures.

The second hypothesis means familiarity increases for well-balanced appearance and behavior. We refer to this as the *synergy effect*. For example, a robot should have robot-like behaviors and a human should have humanlike behaviors [9]. This differs from the uncanny valley because humans do not have sensitive mental models for recognizing robots and other toys.

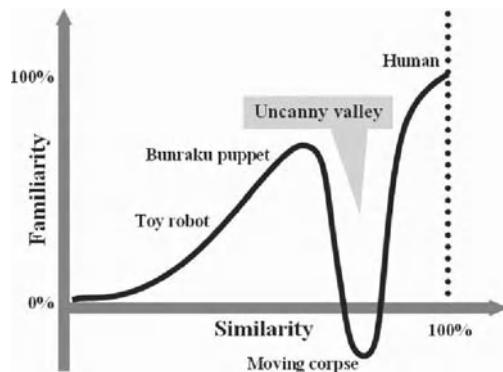


Fig. 7. Uncanny valley

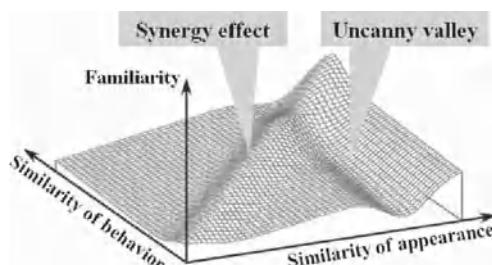


Fig. 8. The extended uncanny valley

Based on these hypotheses, we have extended the graph depicted by Mori as shown in Figure 8, which was obtained by fusing the uncanny valley pro-

vided by the first hypothesis with the synergy effect provided by the second hypothesis. This 3D graph is not exact, but rather conceptual as is Mori's graph. Nevertheless it is still a significant guide for our research. Our important role is to verify the structure of the graph through development of androids and cognitive experiments with them and obtain a more precise graph.

Age-Dependent Uncanny Valley

There is also an age-dependent relationship. One-year-old babies were attracted to the child android and were unperturbed by even jerky, robotic movements. However children between the ages of three and five were afraid of the android and refused to face it. We found this phenomenon with preliminary experiment using infants.

We consider the reasons to be as follows. If the baby's model of others is not so well-developed, the android may be able to pass itself off as human. Adults know the android is not human, so they do not expect it to fit closely a human model. However young children seem to be in the middle ground of applying a human model to the android, but finding it mismatches. This is a kind of uncanny valley. We expect to learn more about the developmental process of human recognition models of infants by verifying this age-dependent uncanny valley.

Conscious and Unconscious Recognition

Another important viewing point for the evaluation criteria is whether it is conscious or unconscious. The SD method evaluates conscious recognition of the subjects. In contrast, our previous approach evaluates the unconscious recognition. Which is more significant? In the evaluation of an android, this question is difficult to answer. In our experience, the subjects react with it as if it is a human even if they consciously recognize it as an android.

We have observed the eye movement of subjects. Figure 9 shows eye movements between a child and the child android. The child android is very eerie because of the jerky movements. As shown in the figure, the subject cannot keep gazing on the face of the human child and often looks at the upper right corner. In contrast, the subject keeps gazing at the face of the android.

Previous works in psychology suggest the following two reasons why the subject cannot keep gazing at the human face.

- Arousal reduction theory: Humans shift their gazing direction to create barriers against external signals for concentration
- Differential cortical activation theory: The eye movements are caused by brain activities.

However these theories do not fit our experiment. We consider there is the third reason as follows

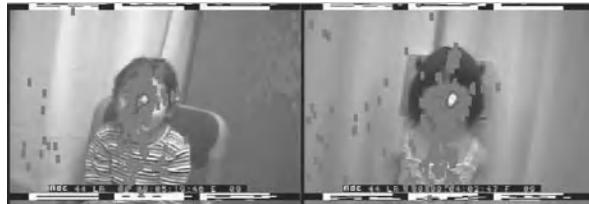


Fig. 9. Eye movements as to a human child and the android

- Social signal theory: The eye movement is a way of representing thinking [6]

We consider a human indicates he/she is social by not continually gazing at the face of another.

Possibility of an Android as a Human

Then, we have another experiment with the adult android that has humanlike behaviors. After 5 min. habituation, the subject answered questions posed by the android. During the habituation, the android talked while using humanlike body movements. Of course, the subject became aware that it was an android because 5 min. is enough long to observe the details.

We have prepared two tasks for the subject. One is to respond with either lies or the truth to questions posed by the android. The other is to answer seriously both easy and difficult questions posed by the android.

When we humans, tell a lie, it is hard to keep gazing at the face of the person to whom we are lying. For the first task, many subjects shift their gaze when they tell a lie. For the second task, almost all subjects shift their gaze when difficult questions are involved. With respect to the second task, we have compared human-human interaction and human-android interaction. Figure 10 shows the results that subjects shift their gaze in the same way for both humans and androids.



Fig. 10. Comparison between human-human interaction and human-android interaction. The gazing directions are represented by 9 areas with the numbers representing percentages.

Obviously the subjects consciously recognized the other as an android. However they unconsciously recognized it as a human and dealt with it as a social partner. Although we have discussed evaluation criteria, this finding suggests the evaluation process looks more complicated.

Through the experiment, we have reached at the following hypothesis. If a human unconsciously recognizes the android as a human, he/she will deal with it as a social partner even if he/she consciously recognizes it as a robot. At that time, the mechanical difference is not significant; and the android can naturally interact and attend to human society. Verification of this hypothesis is not easy and will take a long time. However it is an important challenge that contributes to developing deeper research approaches in both robotics and cognitive science.

This paper has been proposed android science as a new cross- interdisciplinary framework. Our purpose is not to develop the androids as commercial products, but rather to study principles of human-robot interaction. The author believes android science will contribute for it.

References

1. S. Harnad, The symbol grounding problem, *Physica D*, Vol. 42, pp. 335-346, 1990.
2. H. Ishiguro, Distributed vision system: A perceptual information infrastructure for robot navigation, *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*, pp. 36-41, 1997.
3. H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, R. Nakatsu, Robovie: An interactive humanoid robot, *Int. J. Industrial Robotics*, Vol. 28, No. 6, pp. 498-503, Nov. 2001.
4. H. Ishiguro, Toward interactive humanoid robots: a constructive approach to developing intelligent robot, *Proc. 1st Int. Joint Conf. Autonomous Agents & Multiagent Systems*, Invited talk, Part 2, pp. 621-622, 2002.
5. T. Kanda, H. Ishiguro, M. Imai, T. Ono, Development and evaluation of interactive humanoid robots, *Proceedings of the IEEE*, Vol. 92, No. 11, pp. 1839-1850, Nov. 2004.
6. A. McCarthy, K. Lee, and D. Muir. Eye gaze displays that index knowing, thinking and guessing, *Proc. Annual Conf. American Psychological Society*, 2001.
7. M. Mori, *Bukimi no tani* (the uncanny valley), *Energy*, Vol. 7, pp. 33?35, 1970.
8. K. Perlin, Real time responsive animation with personality, *IEEE Transactions on Visualization and Computer Graphics*, 1, 1, pp. 5-15, 1995.
9. T. Charminade and J. Decety, A common framework for perception and action: Neuroimaging evidence, *Behavioral & Brain Sciences*, Vol. 24, pp. 879-882, 2001.
10. A. Turing, Computing machinery and intelligence, *Mind*, Vol. 59, pp. 433-460, 1950.

Mimetic Communication Theory for Humanoid Robots Interacting with Humans

Yoshihiko Nakamura, Wataru Takano, and Katsu Yamane

Department of Mechano-Informatics, University of Tokyo, Bunkyo, Tokyo
113-8656 JAPAN nakamura@ynl.t.u-tokyo.ac.jp

Summary. The theory of behavioral communication for humanoid robots that interact with humans is discussed in this paper. For behavioral communication, it is fundamental for a humanoid robot to recognize the meaning of the whole body motion of a human. According to the previous works, it can be done in the symbolic level by adopting the proto-symbol space defined by the Hidden Markov Models based on the mimesis theory. The generation of robot motions from the proto-symbols is also to be done in the same framework. In this paper, we first introduce the meta proto-symbols that stochastically represent and become signifiants of the interaction of a robot and a human. The meta proto-symbols are a little more abstract analogy of the proto-symbols and recognize/generate the relationship of the two. A hypothesis is then proposed as the principle of fundamental communication. The experimental result follows.

Key words: Mimetic Communication, Humanoid Robot, Human Robot Interaction, Mimesis Theory, Proto Symbol Space, Hidden Markov Model.

1 Introduction

Communication is defined as a process of information exchange between social creatures through common systems such as gestures, signs, symbols or languages. Gesture or behavioral communication has much longer history than that of language for the human beings. Mimesis hypothesis suggests that the humans started the use of signs and symbols in communication through behavioral imitation [1]. The importance of behavioral communication lies in the fact that it always stays behind and enables physical interactions between two humans.

The link between a sender and receiver of messages is a necessary condition for any communication [2]. The discovery of mirror neurons [3] [4] was an epoch-making event in neuroscience. The mirror systems enabled the link between a subject and the others through gesture messages. Namely, the mirror systems are related to the development of communication [5].

In this paper, we focus on behavioral communication to support interactions between humanoid robots and humans. We discuss the fundamental theory of behavioral communication for humanoid robots that interact with humans. For behavioral communication, it is essential for a humanoid robot to recognize the meaning of the whole body motion of a human. According to the previous works [6]-[9], it can be done in the symbolic level by adopting the proto-symbol space defined by the Hidden Markov Models based on the mimesis theory. The generation of robot motions from the proto-symbols is also to be done in the same framework.

We first introduce the meta proto-symbols that stochastically represent and become signifiants of the interaction of a robot and a human. The meta proto-symbols are a little more abstract analogy of the proto-symbols and recognize/generate the relationship of the two. A hypothesis is then proposed as the principle of fundamental communication. Namely, the communication is to recognize the relationship of the two and try to maintain it, whether it is cooperative or competitive. Technical implementation of the hypothesis can be done by simply short-circuiting the output of recognition and the input of generation of the metaproto-symbols. The experimental result follows using a 20 DOF small-size humanoid robot, UT- μ 2 magnum [18].

For interaction between robots and humans, Canamero et al [10] discussed the interface of humanoid robot named *Feelix* that showed various kinds of facial expression in response to touch stimulus from a human. Breazeal [11] studied a model of social interaction between an infant and a caretaker, and then developed a robot named *Kismet* with the social model. Imitation learning is also an active field of robotics research and various kinds of approaches have been presented [12]. Samejima et al [13] [14] reported that a two-link robot could symbolize, recognize motion patterns using predicting modules, and generate motion patterns using controlling modules. Morimoto et al [15] proposed a hierarchical reinforcement learning in order to acquire motion dynamics. Not many works have been done to bridge communication and imitation learning. Billard et al [16] presented a very interesting approach to acquisition of communication skill based on the child-mother model of imitation learning. This architecture was named DRAMA, the general control Dynamic Recurrent Associate Memory Architecture [17].

2 Mimetic Communication Model of Interaction

The proto symbol space [9] [8] is a vector space approximately structured for the set of the Hidden Markov Models. A HMM is acquired from a motion pattern and to be used to recognize and generate it. In the proto symbol space, we can handle continuous transition of motion patterns. Fig.1 shows the image of bidirectional computation using the HMMs.

The recognition of transition of motion patterns or the generation of motion pattern that smoothly changes from one motion pattern to another is

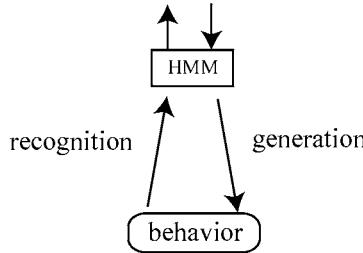


Fig. 1. Proto symbol space. Stochastic parameters of Hidden Markov Models are used for bidirectional computation of recognition and generation of motion patterns.

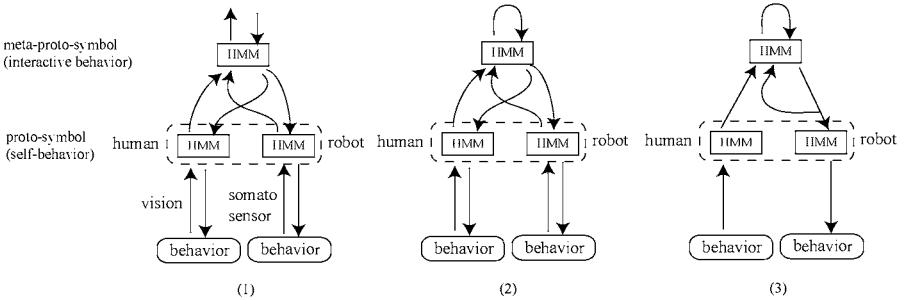


Fig. 2. Mimetic communication. A behavioral communication model for robots interacting with humans.

represented as a moving point in the proto symbol space. For the motion patterns of the point, we can define the second proto symbol space, which is called the meta proto symbol space since it represents the motion patterns of symbols.

In this paper, we propose to use the meta proto symbol space to represent the communication/interaction between a robot and a human or between the self and the partner.

Fig.2 explains the principle of mimetic communication model for interaction proposed using the meta proto symbol space.

In Fig.2 (1), a proto symbol space executes bidirectional computation of the self (robot) as well as that of the partner (human) of interaction. A meta proto symbol space is set in the second hierarchy and takes the sequences of proto symbols of the self and the partner as its behavior and executes bidirectional computation. The two recognition outputs of the self and the partner from the proto symbol space become the recognition input of the meta proto symbol space. The generation output of the meta proto symbol space separates into two and become the generation inputs of the proto symbol space. The recognition output of the meta proto symbol space implies for the self (robot) the estimated state of interaction, while the generation input of the

meta proto symbol space implies the control strategy for the interaction. The essence of interaction is in the process of computing the control strategy from the estimated states of interaction. The process should vary and be designed depending on whether the interaction is purposeful, emotional, contingent, or naturally drifting.

A hypothesis for designing fundamental interaction, namely naturally drifting interaction is to short-circuit the recognition output and the generation input of the meta proto symbol space as shown in Fig.2 (2). Because the naturally drifting interaction can be modeled to estimate the states of interaction and attempt to maintain and generate the states. Note that the naturally drifting interaction model can represent not only cooperative or friendly interactions, but also competitive or hostile interactions..

The technical implementation was done in the form of Fig.2 (3) by eliminating generation processes of the partner (human) and approximating recognition processes of the self (robot).

3 General Algorithms of Recognition and Generation

3.1 Computational Problems

For both the proto symbol space and the meta proto symbol space, the computational problems are common. In the literature [9] the computation of recognition and generation were discussed in the simplest case, namely, as an interpolation between two proto symbols. The norm of the vector space was defined by the Kullback-Leibler information modified to satisfy the symmetry property. The proto symbol space was then constructed through the multi dimensional scaling.

For the continuous recognition of motion patterns, we use stepwise moving recognition. Fig.3 shows the stepwise moving recognition for the meta proto symbol space, where W_{span} is the time width of the moving window, and W_{step} is the moving time step of the window.

3.2 Motion Recognition

We represent the observation through the moving window by $\mathbf{O}^i(t)$ where $i = \{H, R\}$ is used to indicate human (H) and robot (R). Then, $P(\mathbf{O}^i(t)|\lambda_j)$ shows the likelihood that observation $\mathbf{O}^i(t)$ is generated by the proto symbol j .

Motion recognition is to find the coordinates in the proto symbol space that is appropriate for the observation. We propose the single Gaussian model for motion recognition as shown in Fig.4. We define a Gaussian that has value $P(\mathbf{O}^i(t)|\lambda_j)$ at the coordinates of proto symbol j of i , \mathbf{x}_{psj}^i . The mean vector, $\boldsymbol{\mu}^i(t)$, and the covariance matrix, $\boldsymbol{\Sigma}^i(t)$, of the Gaussian are computed as follows:

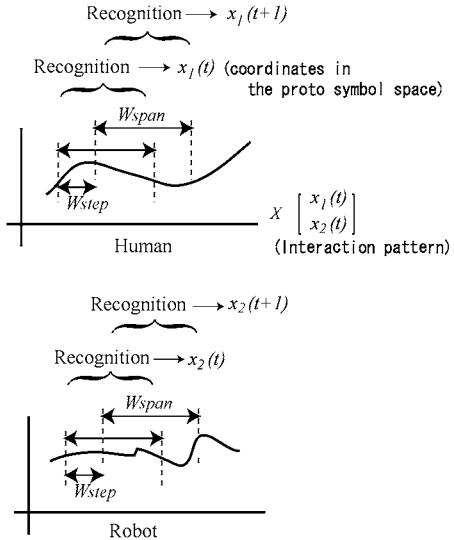


Fig. 3. Procedure for recognition of motion patterns

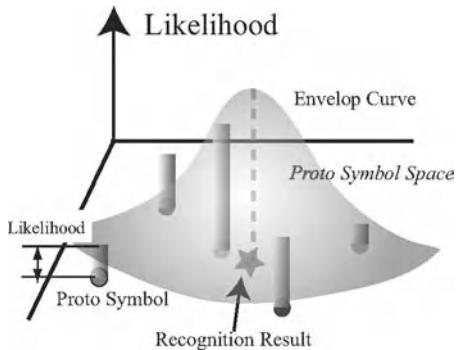


Fig. 4. The single Gaussian model for motion recognition.

$$\boldsymbol{\mu}^i(t) = \frac{1}{N_{\mathcal{PS}}^i} \sum_{j=1}^{N_{\mathcal{PS}}^i} P(\boldsymbol{O}^i(t)|\lambda_j) \boldsymbol{x}_{\mathcal{PS}}^i \quad (1)$$

$$\boldsymbol{\Sigma}^i(t) = \frac{1}{N_{\mathcal{PS}}^i} \sum_{j=1}^{N_{\mathcal{PS}}^i} \left(\mathbf{x}_{\mathcal{PS} j}^i - \boldsymbol{\mu}^i(t) \right) \left(\mathbf{x}_{\mathcal{PS} j}^i - \boldsymbol{\mu}^i(t) \right)^T \quad (2)$$

where $N_{\mathcal{PS}}^i$ is the number of proto symbols. The coordinates for the observation is given by $\mathbf{x}^i(t) = \boldsymbol{\mu}^i(t)$.

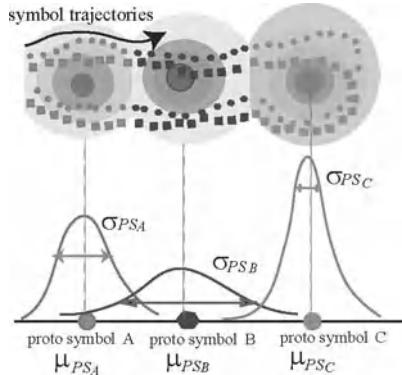


Fig. 5. Image of clustering analysis for compensating the sparseness of the proto symbols.

3.3 Likelihood of a Point in the Space

Computation of likelihood for observation is done in the previous subsection. We also need computation of likelihood of a point in the proto symbol space being associated with a proto symbol. This computation will be used for motion generation. The proto symbols are rather sparse in the proto symbol space and cannot provide a meaningful likelihood for a point distant from them.

We apply cluster analysis for the history of observations and use the result to compensate the sparseness of the proto symbols. For each observation $\mathbf{O}^i(t)$, we have a point in the proto symbol space, $\mathbf{x}^i(t) = \boldsymbol{\mu}^i(t)$. We also compute the proto symbol that provides the maximum likelihood. Namely,

$$\mathcal{R}^i = \arg \max_j P(\mathbf{O}^i(t) | \lambda_j) \quad (3)$$

where \mathcal{R}^i shows an integer indicating the proto symbol of the highest likelihood. Fig.5 shows the image of cluster analysis for compensating the sparseness of the proto symbols. The Gaussian of the proto symbol j of i is then obtained as follows:

$$\boldsymbol{\mu}_{\mathcal{P}\mathcal{S}_j^i} = \frac{1}{n_j^i} \text{SUM}\{\mathbf{x}^i(t) | \mathcal{R}^i(t) = j\} \quad (4)$$

$$\boldsymbol{\Sigma}_{\mathcal{P}\mathcal{S}_j^i} = \frac{1}{n_j^i} \text{SUM}\{(\mathbf{x}^i(t) - \boldsymbol{\mu}_{\mathcal{P}\mathcal{S}_j^i})(\mathbf{x}^i(t) - \boldsymbol{\mu}_{\mathcal{P}\mathcal{S}_j^i})^T | \mathcal{R}^i(t) = j\} \quad (5)$$

where n_j^i denotes the number of observations that are recognized as associated with proto symbol j .

3.4 Motion Generation

Using the Gaussian computed in the previous subsection, we can generate a motion pattern of the robot indicated by point $\mathbf{x}^H(t)$ in the proto symbol space as follows:

$$\mathbf{o}_G(t) = \sum_{k=1}^{N_{PS}^H} w_k \mathbf{o}_{Gk}(t) \quad (6)$$

$$w_j(t) = \frac{P(\mathbf{x}^H(t)|\lambda_j^H)}{\sum_{k=1}^{N_{PS}^H} P(\mathbf{x}^H(t)|\lambda_k)} \quad (7)$$

where $P(\mathbf{x}^H(t)|\lambda_j^H)$ is the likelihood of a point $\mathbf{x}^H(t)$ with respect to the j -th proto symbol. $\mathbf{o}_{Gk}(t)$ means a generated motion pattern by the k -th proto symbol.

In order to use the generated motion patterns for the motion of humanoid robot, we will have to consider dynamical consistency, discontinuity at switching motion patterns, and the other constraints such as work space of joints and self-collision and appropriately modify them in realtime.

4 Experiments

The experiments of mimetic communication theory were conducted. Small-size humanoid robot, UT-*mu* magnum [18] was used. The realtime motion capture system was used to measure the whole body motion of a human and their interaction was investigated. The proto symbols and the proto symbol space were developed to model the motion patterns of the humanoid robot. The same proto symbol space was used for recognition of the human subject.

The meta proto symbol space was developed by showing the typical fighting scenes of two human subjects. The fights of the humanoid robot and the human subject were demonstrated at AICHI EXPO2005 in June 2006. They did not make physical contacts, rather they fought only in the virtual screen.

The sampling time for the motion capture is 30ms. We used the model of a humanoid robot [20] with 4 active joints in each arm, 6 active joints in each leg. The motion patterns are therefore represented by sequences of 46 dimensional vectors. The window span of motion data for the recognition is 180ms, which means that the motion data includes only 6 frames of captured data.

The output motion patterns of humanoid robots were modified in realtime to consider dynamical consistency, discontinuity at switching motion patterns, and the other constraints.

The stage at the EXPO is shown in Fig.?. The overall experimental system was set up as shown in Fig.??.



Fig. 6. Realtime virtual fight between a humanoid robot and a human subject at EXPO2005.

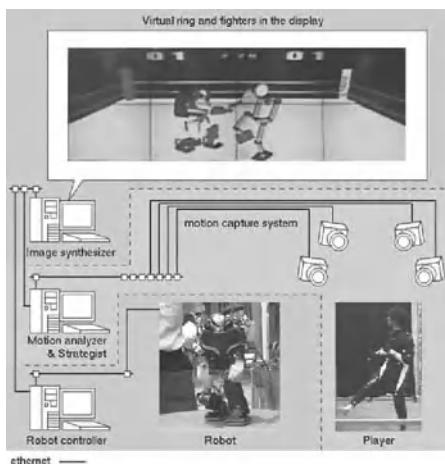


Fig. 7. Overall experimental setup for realtime virtual fight.

Fig.6 shows the virtual fighting scene between the humanoid robot and the human subject. In this figure, we see that the robot bends down against the human's punch and takes a punch at the human, and that the robot tries to protect with a leg against the human's kick and then give a kick to the human. The robot was capable of recognizing the human's behaviors and generating the suitable behaviors corresponding to the situation. These experimental results, we claim that the mimetic communication model is valid for acquiring primitive communication ability.

5 Conclusion

The mimesis model bridges the continuous motion patterns of the body of robot and the system of symbols. In this paper, we developed a fundamen-

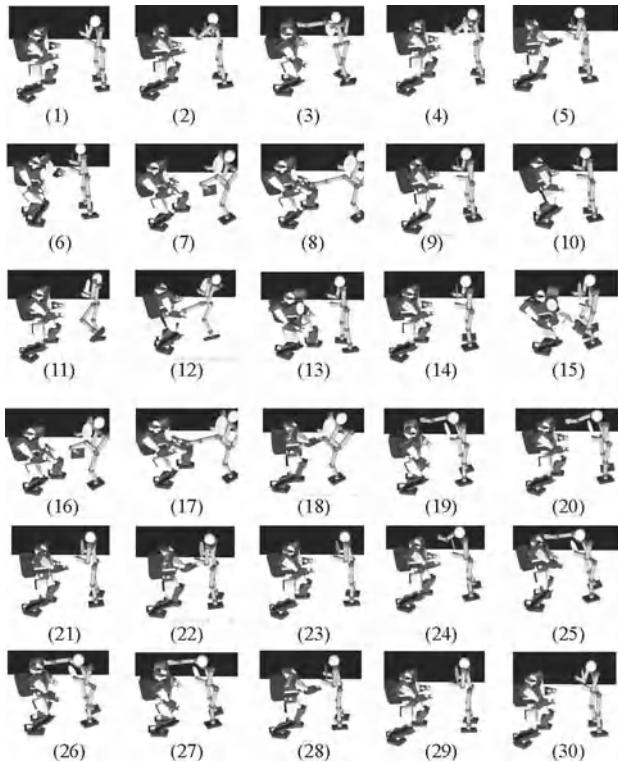


Fig. 8. Experimental result of interaction between the robot and the human.

tal theory to enable behavioral interactions between humanoid robots and humans. The interaction is supported by mimetic communication.

A hypothesis for designing fundamental interaction, namely naturally drifting interaction was established. It was to short-circuit the recognition output and the generation input of the meta proto symbol space. Because the naturally drifting interaction can be modeled to estimate the states of interaction and attempt to maintain the flow of states.

The mimetic communication theory was integrated into the realtime fighting demonstration of a humanoid robot and a human subject in the virtual screen. The experimental results showed the effectiveness of the theory.

Acknowledgment

The authors would like to express their thanks to Dr. Tomomichi Sugihara and Mr. Kou Yamamoto for their experimental supports using UT-*mu* mag-

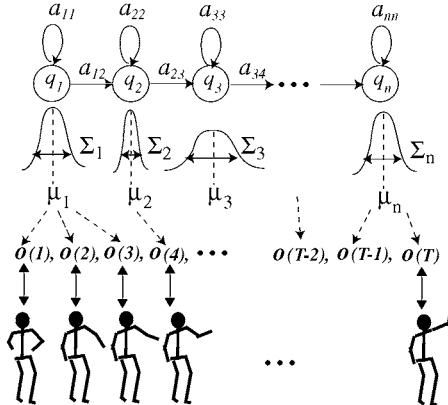


Fig. 9. A continuous Hidden Markov Model with left-to-right type state transition and continuous output vectors.

num. The experiments were conducted for the demonstration at EXPO2005, Nagoya, Japan as a part of NEDO Prototype Robot Project.

This research was supported by Category S of Grant-in-Aid for Scientific Research (2003-2007: 15100002), Japan Society for the Promotion of Science.

Appendix

Mimesis Model [9]

The mimesis model consists of a set of proto-symbols that allow bidirectional computation of recognition and generation of the whole-body motions, just like the mirror system. A set of the stochastic parameters of a Hidden Markov Model (HMM) acquired for a segmented whole-body motion is considered a proto-symbol. In the literature [9], the pseudo-distance is defined between the proto-symbols, that allows to form an Euclidean space to interpolate and extrapolate the proto-symbols. The Euclidean space is named the proto-symbol space.

The left-to-right model for state transition and the continuous HMMs were adopted to construct the mimesis model as shown in Fig.9. A HMM is defined by a set of stochastic parameters $\lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\Pi}\}$, where $\mathbf{A} = \{a_{ij}\}$ is a matrix of state transition probability from node i to node j , $\mathbf{B} = \{b_i\}$ is a vector of output probability, and $\boldsymbol{\Pi} = \{\pi_1, \pi_2, \dots, \pi_n\}$ is a set of initial node probability. The probability desity functions are assumed Gaussian as follows:

$$b_i(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^m |\boldsymbol{\Sigma}_i|}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)\right\} \quad (8)$$

where μ_i and Σ_i denote the mean vector and the covariance matrix of node i . x is an m -dimensional input vector. The parameters of HMM are computed by the Baum-Welch algorithm [19]. For computational efficiency the covariance matrix was approximated by a diagonal matrix with its diagonal elements.

Motion recognition is to find one among all the HMMs whose probability $P(O|\lambda)$ to generate the observed motion pattern O is maximum.

Triple Averaging for Motion Generation

Motion generation is to recover the motion patterns encoded by the proto symbols. This paper proposes the triple averaging method for motion generation, which is explained as follows:

step1 Compute a sequence of state transition Q_G using the transition probability A and random variables.

step2 Repeat *step1* for n_g times and obtain $Q_{G1}, Q_{G2}, \dots, Q_{Gn_g}$. Compute the mean state transition \bar{Q}_G by simply averaging them.

$$\bar{Q}_G = \{q_{s_k}\} \quad s_k = \text{int}\left(\frac{1}{N} \sum_i^N s_{k_i}\right) \quad (9)$$

where $Q_{G_i} = \{q_{s_{k_i}}\}$. s_k represents the state number at time k . If the state at time k is q_j , then $s_k = j$ and $s_{k+1} = j$ or $j + 1$. If $s_k = n$ or null, then $s_{k+1} = \text{null..}$. N is the number of s_{k_i} that are not null.

step3 Compute a sequence of output vector \hat{O}_G according to the mean state transition nodes \bar{Q}_G , using the output vector probability B and random variables.

step4 Repeat *step3* for n_o times and obtain output vector sequences $\hat{O}_{G1}, \hat{O}_{G2}, \dots, \hat{O}_{Gn_o}$. Taking their average, compute the mean output vector sequence \bar{O}_G .

step5 Repeat *step1* through *step4* for n_t times and obtain the mean output vectors $\bar{O}_{G1}, \bar{O}_{G2}, \dots, \bar{O}_{Gn_t}$. Taking their average, finally compute the generated motion pattern O_G .

Inamura's generation process [9] included double averaging of *step2* and *step4*. The third averaging in *step5* was effective to deliver a smooth output vector sequence even when the total cost of averaging was maintained constant.

References

1. M. Donald: Origin of the Modern Mind Harvard University Press, Cambridge, 1991.
2. A. M. Liberman and I. G. Mattingly: "A specialization for speech perception," Science, no. 243, pp. 489-494, 1989.

3. V. Gallese and A. Goldman: Mirror neuron and the simulation theory of mind-reading, *Trends in Cognitive Science*, vol. 2, no. 12, pp. 493-501, 1998.
4. G. Rizzolatti, L. Fogassi and V. Gallese: Neurophysiological mechanisms underlying the understanding and imitation of action, *Nature Reviews*, pp. 661-670, 2001,
5. G. Rizzolatti and M. A. Arbib: "Language within our grasp," *Trend in Neurosciences*, vol. 21, no. 5, 1998
6. T. Inamura, Y. Nakamura, H. Ezaki and I. Toshima: "Imitation and Primitive Symbol Acquisition of Humanoids by the Integrated Mimesis Loop," *Proc. of IEEE International Conference on Robotics and Automation*, Vol.4, pp.4208-4213, Seoul, Korea, May, 2001.
7. T. Inamura , I. Toshima and Y. Nakamura: "Acquisition and Embodiment of Motion Elements in Closed Mimesis Loop," *Proc. of IEEE International Conference on Robotics and Automation*, Vol.2, pp.1539-1544, Washington D.C., U.S.A., May, 2002
8. Y. Nakamura, T. Inamura and H. Tanie: "A Stochastic Model of Embodied Symbol Emergence," *Proc. of 11th International Symposium of Robotics Research*, 2003.
9. T. Inamura, I. Toshima, H. Tanie and Y. Nakamura: "Embodied Symbol Emergence based on Mimesis Theory," *International Journal of Robotics Research*, vol. 23, no. 4 pp. 363-377, 2004.
10. L. D. Canamero and J. Fredslund: "How Does It Feel? Emotional Interaction with a Humanoid LEGO Robot," In *Proc. of American Association for Artificial Intelligence Fall Symposium*, FS-00-04, 2000.
11. C. Breazeal: "A Motivational System for Regulating Human-Robot Interaction," In *Proc. of the 15th National Conference on Artificial Intelligence*, WI-54-61, 1998.
12. C. Breazeal and B. Scassellati: "Robots that imitate humans," *Trends in Cognitive Science*, Vol.6 No.11, pp.481-487, 2002
13. K. Samejima, K. Katagiri, K.Doya and M. Kawato: "Symbolization and Imitation Learning of Motion Sequence Using Competitive Modules," *IEICE Trans*, Vol.J85-D-II No1, pp.90-100, 2002.
14. K. Samejima, K. Doya, and M. Kawato: "Intra-module credit assignment in multiple model-based reinforcement learning," *Neural Networks* Vol.16 No.7, pp.985-994, 2003
15. J. Morimoto and K. Doya: "Hierarchical reinforcement learning for motion learning: learning "stand-up" trajectories," *Advanced Robotics*, vol. 13, no. 3, pp. 267-268, 1999
16. A. Billard and K. Dautenhahn: "Grounding communication in situated, social robots," In *Proc. of Towards Intelligent Mobile Robots Conference*, 1997
17. A. Billard and K. Dautenhahn: "Robot's first steps, Robot's first words...," In *Proc. of Groningen Assembly on Language Acquisition Conference*, 1997
18. T. Sugihara and Y. Nakamura: "Architectural Design of Miniature Anthropomorphic Robots Towards High-Mobility," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005
19. L. Rabiner and B. H. Juang: *Fundamentals of Speech Recognition*, Prentice Hall Signal Processing Series, 1993
20. T. Sugihara and Y. Nakamura: "Online Gait Planning with Boundary Relaxation of Biped Robot," *Proc. of 10th Robotics Symposia*, 2005 (in Japanese)

Mechanism and Design

Session Overview

Mechanisms and Design

Jean-Pierre Merlet

INRIA Jean-Pierre.Merlet@sophia.inria.fr

Robot mechanisms science must be understood as acquiring an in-depth understanding of the mechanical behavior of a robot and involve domains such as kinematics, dynamics and singularity analysis. Two issues must be addressed:

- *analysis*: determine all the mechanical properties of a given robot that are necessary to control it and to verify that its behavior will satisfy a given set of requirements
- *synthesis*: being given a set of requirements determine what should be the mechanical arrangement and the dimensioning of the robot. Synthesis is in general a much more complex issue than analysis

The study of robot mechanisms and of their design is a fundamental and exciting part of robotic science as the mechanical part of the robot will, at the end, condition what the robot can perform in term of tasks and will drastically influence control issues.

It may be believed that this part of robotics is well mastered now, especially for serial industrial robots. Even in that case they are still many open issues. For example for the analysis part, a consequence of manufacturing tolerances is that a real robot will always differ from its theoretical model: managing these uncertainties to certify some robot properties is a complex problem that is far from being solved and involves sophisticated mathematics. It may then be understood that the synthesis of serial industrial arm is also an open problem, especially if manufacturing tolerances are taken into account.

Robotics is also a rapidly evolving field in which new application fields and materials renew and enlarge the mechanisms that must be studied. For example recent moves in robotics toward closed-chain mechanism and nano-robots must be emphasized. In the first case the closed structure allows to reach very good performance level for load, accuracy and stiffness, that open loop mechanisms cannot rival. As for micro-robot the change of scale implies that some mechanical effects, that are usually neglected at the macro scale, become preponderant: it is hence necessary to completely revisit the analysis

and synthesis of such mechanism. Furthermore new materials with astounding properties (e.g. Ghz motion) may be used, although their use and integration in a robotic system is still an open problem.

The papers presented in this session illustrates perfectly such evolution:

1. the paper presented by M. Uchiyama and co-authors describes the use of a closed-loop mechanism for a 6-dof haptic device. Here the stiffness and force/torque capacities of such type of mechanism is a key advantage for the application but requires a careful analysis of the dimensioning
2. B.J. Nelson and co-authors address the building of nanoelectromechanical systems (NEMS). They combine the top-down (direct fabrication) and bottom-up (assembly) approaches to design actuators and sensors with carbon nanotubes and Si nanocoils and present experimental, theoretical and design perspectives
3. in the last paper of the session J-P. Merlet investigates performance indices based on the Jacobian matrix that are used for design purposes and shows that they are not appropriate for closed-loop chains.

Design of a Compact 6-DOF Haptic Device to Use Parallel Mechanisms

Masaru Uchiyama¹, Yuichi Tsumaki², and Woo-Keun Yoon³

¹ Department of Aerospace Engineering, Tohoku University

uchiyama@space.mech.tohoku.ac.jp

² Department of Intelligent Machines and System Engineering, Hirosaki University

tsumaki@cc.hirosaki-u.ac.jp

³ Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST) wk.yoon@aist.go.jp

Summary. We present design of a compact haptic device in which parallel mechanisms are utilized. The design realizes a large workspace of orientational motion in a compact volume of the device. The device is a parallel-serial mechanism consisting of a modified DELTA mechanism for translational motion and a spatial five-bar gimbal mechanism for orientational motion. We derive an analytical model of stiffness for the modified DELTA mechanism which we utilize for the design of a stiff platform for translational motion. The model shows that the compliance matrix is a function of kinematic parameters as well as elastic parameters of each mechanical element. Configuration dependency of the compliance matrix is therefore an important point to be noticed.

1 Introduction

A device to make a bridge between human haptic sense and data space is called a haptic device. It displays the sense of touch to a human. It transfers human haptic sense in the real world to signals in data space. Those devices include a tactile display, a force/torque display, etc. A master arm in a master/slave system is a type of haptic device that displays force/torque information at a slave arm. This paper discusses on the design of such a haptic device of master arm type.

For a haptic device of master arm type, PHANTOM of SensAble Technologies, Inc. [1] is well known. But this has not sufficient force/torque capacity and is unable to display a very rigid feeling. A non-holonomic haptic device to display a rigid contact using a wheel has been proposed [2]. However, it is difficult to realize a haptic device of six DOF (Degrees Of Freedom). Fast six-DOF motion is realized by a haptic device of magnetic levitation type [3]. However, the workspace of the device is limited. A parallel wire system [4] may realize fast motion but requires a large place for itself.

To solve those problems, this paper proposes to apply parallel mechanisms to the design. The target is to realize a compact six-DOF device with large workspace for orientational motion and with capability of high-frequency motion. A compact six-DOF parallel mechanism like [5] may be used as a haptic device. However, the haptic device built by such a parallel mechanism only, will have limitation in orientational workspace. It is difficult to realize a large orientational workspace by such a parallel mechanism. A redundant parallel mechanism [6] may enlarge the workspace to some extent but need extra motors.

A stiff and light-weight mechanism is needed to increase the bandwidth of frequency response. For this purpose, a parallel mechanism is a good selection. Stiffness analysis of a parallel mechanism has been studied by, for example, Gosselin [7]. However, he considers only the stiffness of each actuator. Huang [8] proposed a method of stiffness analysis for a parallel mechanism, in which elastic components are considered. However, his analysis does not deal with bearings at free joints that are often used in the parallel mechanism.

We present a design of a compact haptic device in which parallel mechanisms are utilized and a large orientational workspace is realized in a compact volume of the device [9]. The device is a parallel-serial mechanism consisting of a modified DELTA mechanism for translational motion and a spatial five-bar gimbal mechanism for orientational motion. We derive an analytical model of stiffness for the modified DELTA mechanism to design a stiff platform for translational motion [10].

The paper is organized as follows: In Section 2, the design of a mechanism for the haptic device is presented. In Section 3, a model for stiffness analysis is derived, based on which the design is elaborated in Section 4 to yield a mechanism with well-balanced stiffness. The paper is concluded in Section 5

2 Synthesis of a Compact 6-DOF Mechanism

In this section, we present synthesis of a compact six-DOF mechanism for a haptic device of a master arm type. Design requirements and a six-DOF mechanism to meet the requirements are presented.

2.1 Design Requirements

Design requirements for a mechanism of the targeted haptic device are listed as follows:

1. Capability of six-DOF motion,
2. Capability of high-frequency motion,
3. Compact space for placing, and
4. Large workspace.

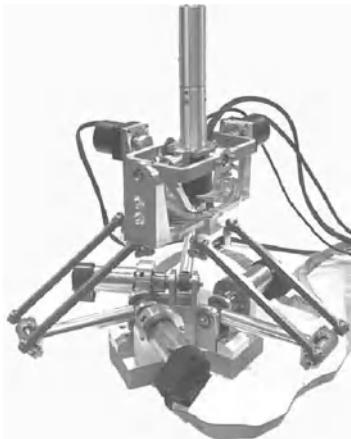


Fig. 1. Overview of the haptic device

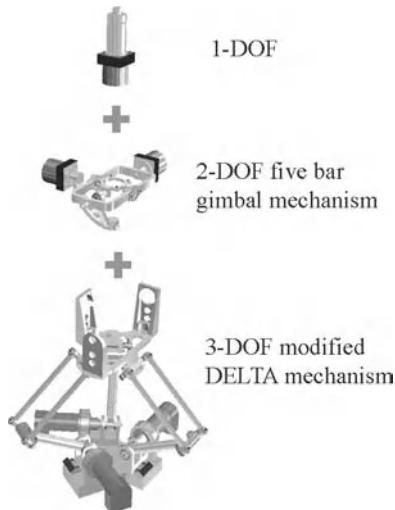


Fig. 2. DOF arrangement

To meet the requirements 1, 2, and 3, the parallel mechanism [11], [12] will be a good candidate. However, the requirement 4 for orientational motion will not be met by a parallel mechanism only, since orientational workspace of the parallel mechanism is usually very limited. In this paper, we solve the problem by applying two parallel mechanisms connected serially to translational motion and to orientation motion, separately.

2.2 A Compact 6-DOF Mechanism

The overview of the mechanism that we synthesize is shown in Figure 1. Architecture of the mechanism is shown in Figure 2, diagrammatically. As shown in the figure, the mechanism consists of three parts, two of which are parallel mechanisms, connected serially. The remaining one is a serial mechanism of one DOF. Thus, the mechanism is a parallel-serial mechanism.

The root of the mechanism is for three-DOF translational motion. Its overview is shown in Figure 3. It is a type of the DELTA mechanism invented by Clavel [13]. However, it is slightly different from the Clavel's DELTA. Difference is shown in Figure 4. The conventional DELTA uses ball joints to connect the rod to the arm on one end and to the traveling plate on the other, while the mechanism proposed in this paper uses ball bearings for those connections. We call this mechanism a modified DELTA mechanism. With the modification in the mechanism, we have larger movable range for the joints between arm and rod and between rod and traveling plate, respectively. This is shown in Figure 5. A similar mechanism has been proposed by Tsai [14].

The middle and the top parts of the mechanism are for orientational motion. The middle part is a five-bar gimbal mechanism [15] as shown in Figure 6.

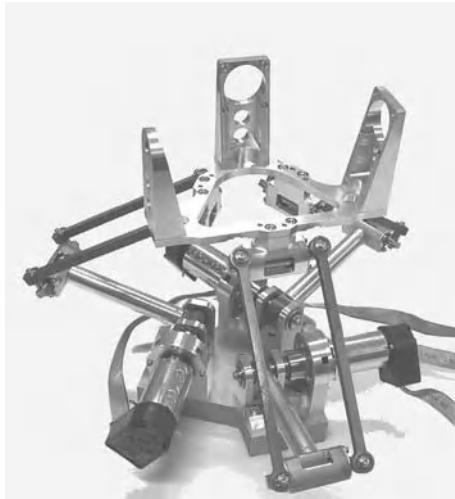


Fig. 3. The modified DELTA mechanism

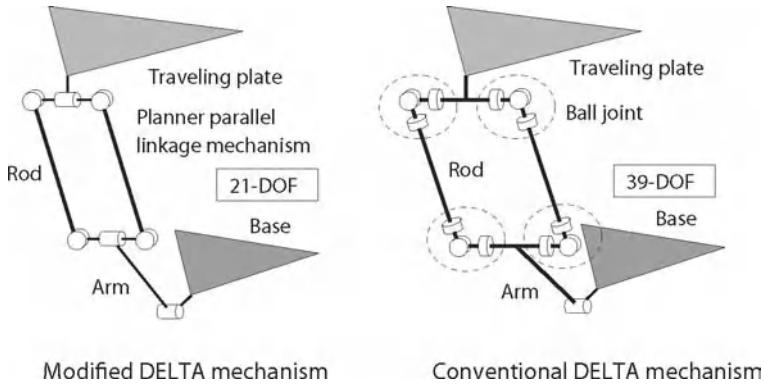


Fig. 4. Arrangement of DOF of the modified DELTA mechanism

This realizes two orientational motions, that is roll θ_r and pitch θ_p . Yaw motion θ_y is realized by the top part of the mechanism. An assembly drawing of the gimbal mechanism is shown in Figure 7. The axes for roll and pitch motions are supported by two bearings grounded on the rigid frame. It is noted that a parallel mechanism to implement the three orientational motions simultaneously has been proposed in [16], but we do not employ this mechanism because its movable range for yaw is small.

To meet the design requirements, the modified DELTA mechanism at the root has to be sufficiently stiff because it has to carry the mechanism for orientational motion. In the following sections we present procedure to design a stiff mechanism for this part.

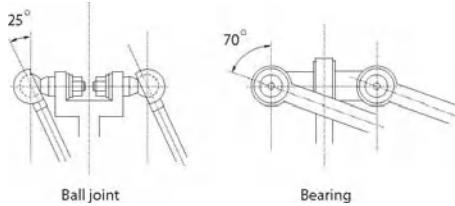


Fig. 5. Difference between limits of ball joint and bearing

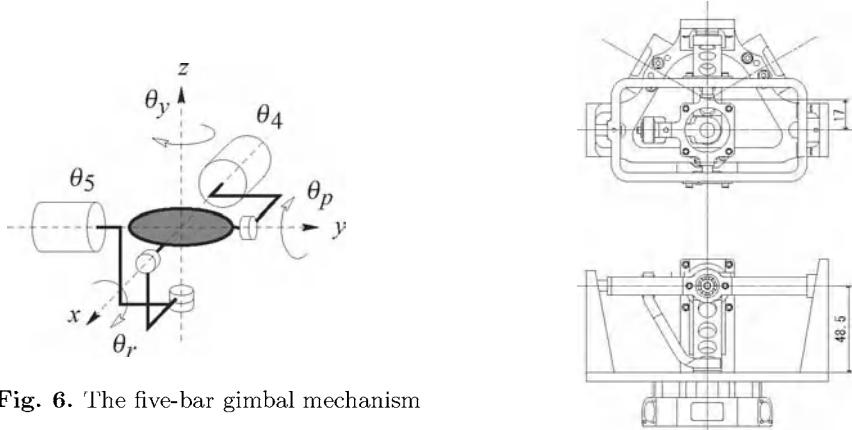


Fig. 6. The five-bar gimbal mechanism

Fig. 7. An assembly drawing of the subsystem for orientational motion

3 Stiffness Analysis of the Modified DELTA Mechanism

In this section, we present a model for the analysis of stiffness of a parallel mechanism. Then, we apply this model to the modified DELTA mechanism presented in the previous section, and point out that the stiffness depends on the kinematic parameters, and therefore on the configuration of the mechanism, even when the same mechanical components are used.

3.1 A Stiffness Model of a Parallel Mechanism

A parallel mechanism is a closed-loop mechanism consisting of a base plate, a traveling plate and elementary chains that connect the two plates. Its stiffness is determined by the stiffness of each elementary chain. We assume the base and traveling plates are rigid. We begin with the analysis of the elementary chain and derive a compliance matrix of the target mechanism.

Tip Compliance of an Elementary Chain

The stiffness of each elementary chain is represented by its tip compliance [17], which we are going to derive here. Svinin and Uchiyama [18] studied the static compliant motion of a serial manipulator with elastic deformations in its structure. Let us suppose that the elementary chain consists of m elastic elements and n joints as shown in Figure 8. Forces and moments at each elastic element cause its elastic deformations of translation and rotation:

$$\mathbf{e}_i = [\delta_{xi} \ \delta_{yi} \ \delta_{zi} \ \phi_{xi} \ \phi_{yi} \ \phi_{zi}]^T \quad (1)$$

where \mathbf{e}_i is an elastic deformation vector of the i th element. δ_{xi} , δ_{yi} and δ_{zi} are the translational deformations, and ϕ_{xi} , ϕ_{yi} and ϕ_{zi} are the orientational deformations, respectively. Assembling the all \mathbf{e}_i for $i = 1, 2, \dots, m$, we have an elastic deformation vector for the elementary chain:

$$\mathbf{e} = [\mathbf{e}_1^T \ \mathbf{e}_2^T \ \cdots \ \mathbf{e}_m^T]^T \quad (2)$$

which is determined by forces and moments on each element. If we suppose linear elasticity, we have

$$\mathbf{e} = \mathbf{C}_e [\mathbf{f}_1^T \ \mathbf{f}_2^T \ \cdots \ \mathbf{f}_m^T]^T \quad (3)$$

where

$$\mathbf{C}_e = \text{diag} [\mathbf{C}_{e1} \ \mathbf{C}_{e2} \ \cdots \ \mathbf{C}_{em}] \quad (4)$$

is the compliance matrix of the all elastic elements, \mathbf{C}_{ei} is the local compliance matrix of the i th elastic element, and \mathbf{f}_i is the forces and moments acting on the i th element. The tip compliance matrix \mathbf{C}_s of the elementary chain which relates the tip deformations of the elementary chain to the forces and moments applied at the tip is given by

$$\mathbf{C}_s = \mathbf{J}_e(\boldsymbol{\theta}, \mathbf{0}) \mathbf{C}_e \mathbf{J}_e^T(\boldsymbol{\theta}, \mathbf{0}) \quad (5)$$

assuming that the elastic deformation \mathbf{e} is small, namely $\mathbf{e} = \mathbf{0}$ in $\mathbf{J}_e(\boldsymbol{\theta}, \mathbf{e})$, where $\mathbf{J}_e(\boldsymbol{\theta}, \mathbf{e})$ is the Jacobian matrix consisting of the Jacobian matrices for each elastic element defined by

$$\mathbf{J}_e(\boldsymbol{\theta}, \mathbf{e}) = [\mathbf{J}_{e1}(\boldsymbol{\theta}, \mathbf{e}) \ \mathbf{J}_{e2}(\boldsymbol{\theta}, \mathbf{e}) \ \cdots \ \mathbf{J}_{em}(\boldsymbol{\theta}, \mathbf{e})]. \quad (6)$$

$\mathbf{J}_{ei}(\boldsymbol{\theta}, \mathbf{e})$ is the Jacobian matrix for each elastic element. $\mathbf{J}_e(\boldsymbol{\theta}, \mathbf{e})$ is a function of both $\boldsymbol{\theta}$ and \mathbf{e} . $\boldsymbol{\theta}$ is a joint angle vector.

Tip Compliance of a Parallel Mechanism

Using the compliance matrix of the elementary chain given by Equation (5) we derive a tip compliance matrix of the parallel mechanism shown in Figure 9.

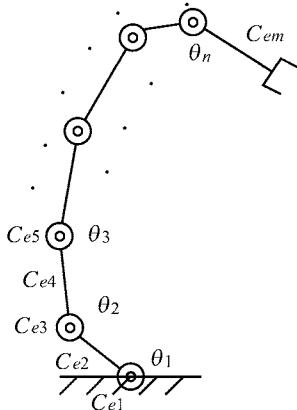


Fig. 8. A model of a serial mechanism

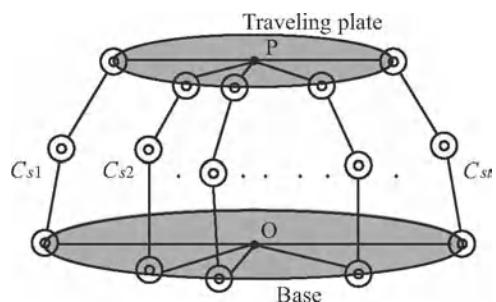


Fig. 9. A model of a parallel mechanism

This parallel mechanism consists of t elemental chains. The point O is the origin of the base plate and the point P on the traveling plate is the output point of the mechanism. Each elementary chain connects the points O and P. The tip compliance matrix of the parallel mechanism is given by

$$\mathbf{C}_p^{-1} = \mathbf{C}_{s1}^{-1} + \mathbf{C}_{s2}^{-1} + \cdots + \mathbf{C}_{st}^{-1} \quad (7)$$

where \mathbf{C}_{sj} ($j = 1, 2, \dots, t$) is the compliance matrix of the j th elementary chain. It should be noted that the elastic deformations of both traveling plate and base plate are ignored.

Now, we have an equation to calculate the tip compliance matrix of the parallel mechanism. To calculate \mathbf{C}_p by Equation (7), we need to have \mathbf{C}_{ei} in Equation (4), that is a model for the i th elastic element. Typical elastic elements in a parallel mechanism are a link and a bearing. We present models for them in the following sections.

Modeling of a Link

Suppose that the i th elastic element is a link of a slender beam. Forces and moments on the beam cause elastic deformations. The relation between the forces and moments and the elastic deformations at the end of the beam is well known. It is expressed by

$$\mathbf{C}_{ei} = \begin{bmatrix} \frac{L^3}{3EI_x} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{L^3}{3EI_y} & 0 & 0 & 0 & -\frac{L^2}{2EI_z} \\ 0 & 0 & \frac{L^3}{3EI_z} & 0 & \frac{L^2}{2EI_y} & 0 \\ 0 & 0 & 0 & \frac{L}{GI_p} & 0 & 0 \\ 0 & 0 & \frac{L^2}{2EI_z} & 0 & \frac{L}{EI_y} & 0 \\ 0 & -\frac{L^2}{2EI_y} & 0 & 0 & 0 & \frac{L}{EI_z} \end{bmatrix} \quad (8)$$

where L is the link length, E is the modulus of the longitudinal elasticity, G is the modulus of the transverse elasticity. I_x , I_y and I_z are the geometrical moments of inertia. I_p is the polar moment of inertia.

Modeling of a Bearing

A bearing is a machine element often used in the parallel mechanism. When the i th element is a bearing, the compliance matrix is given by

$$\mathbf{C}_{ei} = \begin{bmatrix} \frac{1}{k_a} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{k_r} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{k_r} & 0 & 0 & 0 \\ 0 & 0 & 0 & \Phi & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{k_m} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{k_m} \end{bmatrix} \quad (9)$$

where k_a is the coefficient of elasticity in the axial direction, k_r is the coefficient of elasticity in the radial direction, $1/\Phi$ is the coefficient of rotational elasticity in the axial direction, and k_m is the coefficient of rotational elasticity in the radial direction. The direction of the x axis is chosen to be the rotation axis.

If the axial rotation is free, which is usually the case for a bearing, the coefficient of rotational elasticity $1/\Phi$ is nearly zero and Φ is close to infinity. However, if Φ is chosen close to infinity, the numerical calculation becomes unstable. Therefore, Φ should be chosen large enough but not close to infinity. In this paper, the value of 10^8 rad/Nm is used. This value is much larger than any other matrix elements.

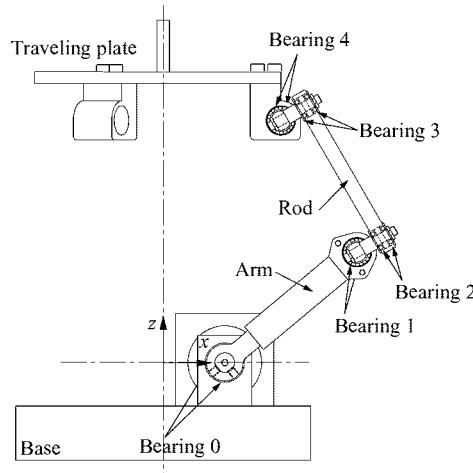


Fig. 10. An assembly drawing of the modified DELTA mechanism

3.2 Application of the Model to the Modified DELTA Mechanism

We apply the stiffness model derived in the previous section to the modified DELTA mechanism in order to obtain a compliance matrix for the mechanism. A schematic diagram of this mechanism is shown in Figure 10. This mechanism is made of a base, bearings 0, three arms, bearings 1 and 2, three rod parts, bearings 3 and 4 and a traveling plate. The output shaft of the motor is supported by the bearings 0. The rod part which consists of a planar parallel mechanism is made of the bearings 2, two parallel rods and the bearings 3 (see also Figures 3 and 4). The passive joints are equipped with conventional ball bearings that are mounted in pairs. We derive the compliance matrix for this mechanism, first deriving a model of the bearing pair, then a model of the rod part, and finally assembling those models.

Modeling of a Pair of Bearings

The connection between the rod part and the arm and between the rod part and the traveling plate is through a pair of bearings as shown in Figure 11. The coefficients of elasticity in the axial and radial directions of this part are obtained as those for a bearing multiplied by two. The coefficient of rotational elasticity in the axial direction is also obtained by the same way. However, the coefficient of rotational elasticity in the radial direction cannot be obtained simply by this way. This is obtained using a model of deformation shown in Figure 12. The moment M in the figure is obtained by

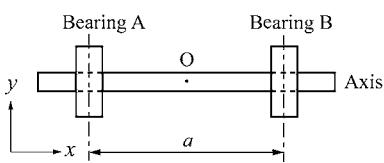


Fig. 11. A two bearing system

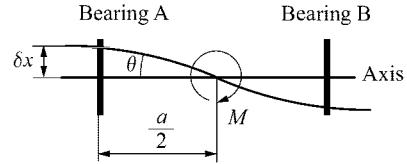


Fig. 12. Elastic deformation of the two bearing system

$$\begin{aligned} M &= 2 \left(k_m \theta + \frac{a}{2} \delta x k_r \right) = 2 \left(k_m \theta + \frac{a}{2} \frac{a}{2} \theta k_r \right) \\ &= 2 \left\{ k_m + \left(\frac{a}{2} \right)^2 k_r \right\} \theta \end{aligned} \quad (10)$$

where δx is the elastic deformation in the radial direction, θ is the rotation angle, and a is the distance between the two bearings as shown in the figure. k_m and k_r are elastic coefficients. Therefore, we have the compliance matrix of the pair of bearings as follows:

$$C_{ei} = \begin{bmatrix} \frac{1}{2k_a} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2k_r} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2k_r} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\phi}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2 \left\{ k_m + \left(\frac{a}{2} \right)^2 k_r \right\}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2 \left\{ k_m + \left(\frac{a}{2} \right)^2 k_r \right\}} \end{bmatrix}. \quad (11)$$

Modeling of the Rod Part by a Parallel Mechanism

The rod part is made of a planar parallel mechanism. This parallel mechanism consists of two parallel rods and the bearings 2 and 3 as shown in Figure 10 (see also Figures 3 and 4). We consider the two rods (Rod L and R) separately as shown in Figure 13 and calculate the compliance matrix of each rod, first, and then, the compliance matrix of the whole rod system. According to Equation (4), the compliance matrices C_{eL} and C_{eR} for the rods L and R, respectively, shown in Figure 13, are given by

$$C_{eL} = \text{diag} [C_{b2L} \ C_{rL} \ C_{b3L}] \quad (12)$$

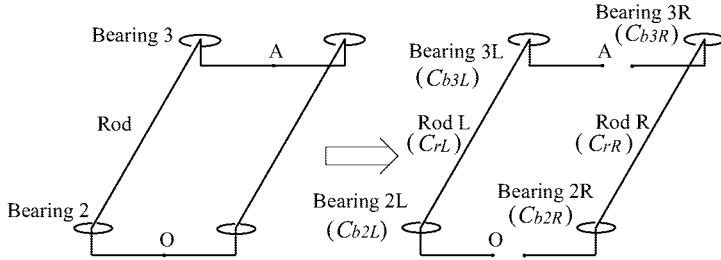


Fig. 13. Modeling of the rod part

and

$$\mathbf{C}_{eR} = \text{diag} [\mathbf{C}_{b2R} \mathbf{C}_{rR} \mathbf{C}_{b3R}] \quad (13)$$

where \mathbf{C}_{b2L} , \mathbf{C}_{b2R} , \mathbf{C}_{b3L} and \mathbf{C}_{b3R} are the compliance matrices of the bearings 2L, 2R, 3L and 3R, respectively. \mathbf{C}_{rL} and \mathbf{C}_{rR} are the compliance matrices of the rods L and R, respectively. The Jacobian matrices $\mathbf{J}_{eL}(\boldsymbol{\theta}, \mathbf{0})$ and $\mathbf{J}_{eR}(\boldsymbol{\theta}, \mathbf{0})$ for the rods L and R are written as

$$\mathbf{J}_{eL}(\boldsymbol{\theta}, \mathbf{0}) = [\mathbf{J}_{b2L}(\boldsymbol{\theta}, \mathbf{0}) \mathbf{J}_{rL}(\boldsymbol{\theta}, \mathbf{0}) \mathbf{J}_{b3L}(\boldsymbol{\theta}, \mathbf{0})] \quad (14)$$

and

$$\mathbf{J}_{eR}(\boldsymbol{\theta}, \mathbf{0}) = [\mathbf{J}_{b2R}(\boldsymbol{\theta}, \mathbf{0}) \mathbf{J}_{rR}(\boldsymbol{\theta}, \mathbf{0}) \mathbf{J}_{b3R}(\boldsymbol{\theta}, \mathbf{0})], \quad (15)$$

respectively, where $\mathbf{J}_{b2L}(\boldsymbol{\theta}, \mathbf{0})$, $\mathbf{J}_{b2R}(\boldsymbol{\theta}, \mathbf{0})$, $\mathbf{J}_{b3L}(\boldsymbol{\theta}, \mathbf{0})$ and $\mathbf{J}_{b3R}(\boldsymbol{\theta}, \mathbf{0})$ are the Jacobian matrices of the bearings 2L, 2R, 3L and 3R, respectively. $\mathbf{J}_{rL}(\boldsymbol{\theta}, \mathbf{0})$ and $\mathbf{J}_{rR}(\boldsymbol{\theta}, \mathbf{0})$ are the Jacobian matrices of the rods L and R, respectively. Therefore, the compliance matrices $\mathbf{C}_{\text{rod}L}$ and $\mathbf{C}_{\text{rod}R}$ for the rods L and R can be written as

$$\mathbf{C}_{\text{rod}L} = \mathbf{J}_{eL}(\boldsymbol{\theta}, \mathbf{0}) \mathbf{C}_{eL} \mathbf{J}_{eL}^T(\boldsymbol{\theta}, \mathbf{0}) \quad (16)$$

and

$$\mathbf{C}_{\text{rod}R} = \mathbf{J}_{eR}(\boldsymbol{\theta}, \mathbf{0}) \mathbf{C}_{eR} \mathbf{J}_{eR}^T(\boldsymbol{\theta}, \mathbf{0}), \quad (17)$$

respectively. Consequently, the compliance matrix of the rod part \mathbf{C}_{2r3} is obtained as

$$\mathbf{C}_{2r3}^{-1} = \mathbf{C}_{\text{rod}L}^{-1} + \mathbf{C}_{\text{rod}R}^{-1}. \quad (18)$$

Stiffness of the Modified DELTA Mechanism

The modified DELTA mechanism consists of three elementary chains as shown in Figure 3. Each elementary chain is connected to the same traveling plate

which does not deform elastically. A point on the traveling plate can be a common tip for the three elementary chains. Therefore, we first derive the compliance matrices \mathbf{C}_{sj} ($j = 1, 2, 3$) for the j th elementary chain using Equation (5). Then, using Equation (7), we obtain the compliance matrix \mathbf{C}_p of the whole mechanism.

The compliance matrix \mathbf{C}_{ej} ($j = 1, 2, 3$) defined by Equation (4) for each elementary chain is given by

$$\mathbf{C}_{ej} = \text{diag} [\mathbf{C}_{b0j} \mathbf{C}_{aj} \mathbf{C}_{b1j} \mathbf{C}_{2r3j} \mathbf{C}_{b4j}] \quad (19)$$

where \mathbf{C}_{b0j} , \mathbf{C}_{b1j} and \mathbf{C}_{b4j} are the compliance matrices of the bearings 0, 1, and 4, respectively. \mathbf{C}_{aj} is the compliance matrix of the arm. \mathbf{C}_{2r3j} is the compliance matrix of the rod part. The Jacobian matrix $\mathbf{J}_{ej}(\boldsymbol{\theta}, \mathbf{0})$ is written as

$$\mathbf{J}_{ej}(\boldsymbol{\theta}, \mathbf{0}) = [\mathbf{J}_{b0j}(\boldsymbol{\theta}, \mathbf{0}) \mathbf{J}_{aj}(\boldsymbol{\theta}, \mathbf{0}) \mathbf{J}_{b1j}(\boldsymbol{\theta}, \mathbf{0}) \mathbf{J}_{2r3j}(\boldsymbol{\theta}, \mathbf{0}) \mathbf{J}_{b4j}(\boldsymbol{\theta}, \mathbf{0})] \quad (20)$$

where $\mathbf{J}_{b0j}(\boldsymbol{\theta}, \mathbf{0})$, $\mathbf{J}_{b1j}(\boldsymbol{\theta}, \mathbf{0})$ and $\mathbf{J}_{b4j}(\boldsymbol{\theta}, \mathbf{0})$ are the Jacobian matrices of the bearings 0, 1 and 4, respectively. $\mathbf{J}_{aj}(\boldsymbol{\theta}, \mathbf{0})$ is the Jacobian matrix of the arm. $\mathbf{J}_{2r3j}(\boldsymbol{\theta}, \mathbf{0})$ is the Jacobian matrix of the rod part. Therefore, the compliance matrix of the j th elementary chain \mathbf{C}_{sj} is obtained by

$$\mathbf{C}_{sj} = \mathbf{J}_{ej}(\boldsymbol{\theta}, \mathbf{0}) \mathbf{C}_{ej} \mathbf{J}_{ej}^T(\boldsymbol{\theta}, \mathbf{0}) . \quad (21)$$

Combining the three matrices for the three elementary chains, the compliance matrix of the whole mechanism \mathbf{C}_p is obtained by

$$\mathbf{C}_p^{-1} = \mathbf{C}_{s1}^{-1} + \mathbf{C}_{s2}^{-1} + \mathbf{C}_{s3}^{-1} . \quad (22)$$

As has been seen in the derivation, the compliance matrix \mathbf{C}_p is a function of the joint angles $\boldsymbol{\theta}$, kinematic parameters of the structure, and elastic parameters of the components such as links, bearings, etc. The model obtained here in this section gives a tool to optimize those parameters through evaluation of the matrix \mathbf{C}_p .

4 Detailed Design of the Modified DELTA Mechanism

We discuss on the design of a modified DELTA mechanism utilizing the stiffness model derived in the previous section. Using this model we elaborate the stiffness of the modified DELTA mechanism to decide its parameters in details. We assume a specification that the workspace be around a sphere of 75 mm radius. The procedure of the design is listed as follows:

1. First, we consider the singular configuration to obtain a set of kinematic parameters and realize a singularity-free workspace. This part does not use the stiffness model but uses only a kinematic model.

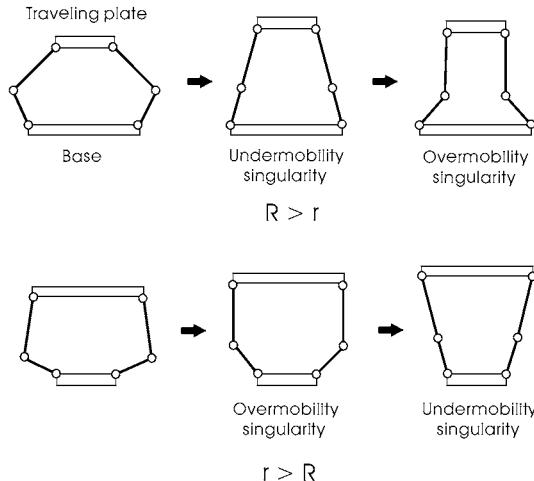


Fig. 14. Singular configurations

2. We discuss how each elastic element influences the tip stiffness and identify the elastic elements having a large influence on the reduction of the tip stiffness. Then, we use the results to improve the tip stiffness.
3. We discuss on kinematic parameters that influence on the tip stiffness and that may be used as a design index for a well-balanced tip stiffness.
4. Finally, we propose an index for the design of a modified DELTA mechanism and give a design example.

4.1 Singular Configuration

Two types of singular configuration are considered. They are undermobility and overmobility singularities [12]. Figure 14 shows the two types of singularity for the modified DELTA mechanism, diagrammatically on a plane. This figure suggests that the case where the base radius R is equal to or larger than the traveling plate radius r be more recommendable than the case where $r > R$ because the former case does not have overmobility singularity in the workspace.

In the following discussion, we set both the traveling plate radius and the base radius equal to 40 mm. Also, since the workspace is given around a sphere of 75 mm radius, we set the sum of the arm length and of the rod length 220 mm, the minimum height 50 mm in order to avoid the undermobility, and the maximum height 200 mm in order to avoid the overmobility.

4.2 Parameters of the Modified DELTA Mechanism

Kinematic parameters of the modified DELTA mechanism are shown in Figure 15. Point O is the origin and point T is the tip position. L is the arm

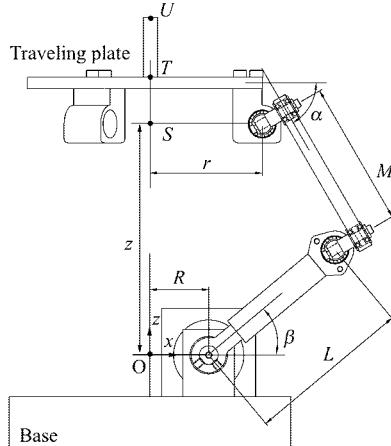


Fig. 15. Kinematic parameters of the modified DELTA mechanism

Table 1. Values of kinematic parameters

Parameter	[mm]
Rod length M	110
Arm length L	110
Base radius R	40
Traveling plate radius r	40

length, M is the rod length, R is the base radius, r is the traveling plate radius and z is the traveling plate height which is the distance between points O and S . Here, we deal with the case where the distance between points S and T is 15 mm, the distance between points T and U is 63.5 mm, and the distance between the two parallel rods at the rod part is 31 mm. The values of M , L , R , and r are given in Table 1.

It is assumed that the arms, rods, motor axes, bearings 0, 1, 2, 3 and 4 deform elastically. More specific details on the parts of the modified DELTA mechanism are given below:

- The arm is a hollow pole, made of A7075 material, with an internal diameter of 8 mm and an external diameter of 12 mm.
- The rod is a prismatic solid beam, made of SUS304 material, one side measure of which is 5 mm and the other 6 mm.
- Bearing 0 is an NSK model F688A. Bearings 1 and 4 are NSK model MR128.
- Bearings 2 and 3 are NSK model F684.
- The motor is a Maxon model A-max 26.

From the values of Table 1 and the elastic parameters of the above parts, we calculate the compliance matrices: C_{aj} for the arms, C_{rL} and C_{rR} for

the rods, \mathbf{C}_{b0j} , \mathbf{C}_{b1j} , \mathbf{C}_{b2L} , \mathbf{C}_{b2R} , \mathbf{C}_{b3L} , \mathbf{C}_{b3R} , and \mathbf{C}_{b4j} for the bearings. All bearings are used in pairs.

The stiffness at the tip position (point U, see Figure 15) of the modified DELTA mechanism changes largely depending on the traveling plate position. Therefore, it is necessary to design the mechanism taking into consideration the tip stiffness at all points in the workspace. However, it is very difficult to evaluate all the 6×6 elements of the tip compliance matrix at all points. Therefore, we simplify the evaluation by limiting the point only in the z direction, with no motion in the (x, y) plane. In this case, \mathbf{C}_p is given by

$$\mathbf{C}_p = \begin{bmatrix} A & 0 & 0 & 0 & B & 0 \\ 0 & A & 0 & -B & 0 & 0 \\ 0 & 0 & C & 0 & 0 & 0 \\ 0 & -B & 0 & D & 0 & 0 \\ B & 0 & 0 & 0 & D & 0 \\ 0 & 0 & 0 & 0 & 0 & E \end{bmatrix} \quad (23)$$

where A , B , C , D and E are non-zero elements determined by kinematic and elastic parameters.

It should be noted that the value of Φ for bearing 0 is measured directly in the real setup and made 0.0058 rad/Nm . In the bearing 0 a motor axis is inserted. Therefore, the compliance around this axis depends on the performance of the motor, control law, etc.

4.3 Influence of Each Elastic Element on the Tip Stiffness

Evaluating \mathbf{C}_p in Equation (23) for each of the compliance matrices \mathbf{C}_{aj} for the arms, \mathbf{C}_{rL} and \mathbf{C}_{rR} for the rods, \mathbf{C}_{b0j} , \mathbf{C}_{b1j} , \mathbf{C}_{b2L} , \mathbf{C}_{b2R} , \mathbf{C}_{b3L} , \mathbf{C}_{b3R} , and \mathbf{C}_{b4j} for the bearings, with the rest of them being zero, we know influences of each elastic element on the tip compliance matrix. Through this numerical analysis we find that the influence on the element A and B of the bearings 2 and 3 is large, and we decrease the influence by replacing the bearings by ceramic bearings. Also, to decrease the influence of the arm on the elements C , D and E , we change the arm internal diameter to 10 mm and its external diameter to 14 mm. Like this way, the compliance matrix is improved.

4.4 Relation Between Tip Stiffness and α

The changes of the compliance matrix for the tip (point U) under elastic deformation of all the elements together (arms, rods, motor axes, bearings 0, 1, 2, 3, and 4) are shown in Figure 16, where α is the angle between the traveling plate and the rod as has been shown in Figure 15. According to Figure 16, when α increases, each of the elements A , B , C , D and E of Equation (23) changes as follows:

- The compliance in the direction of x - and y -axes (element A) increases.

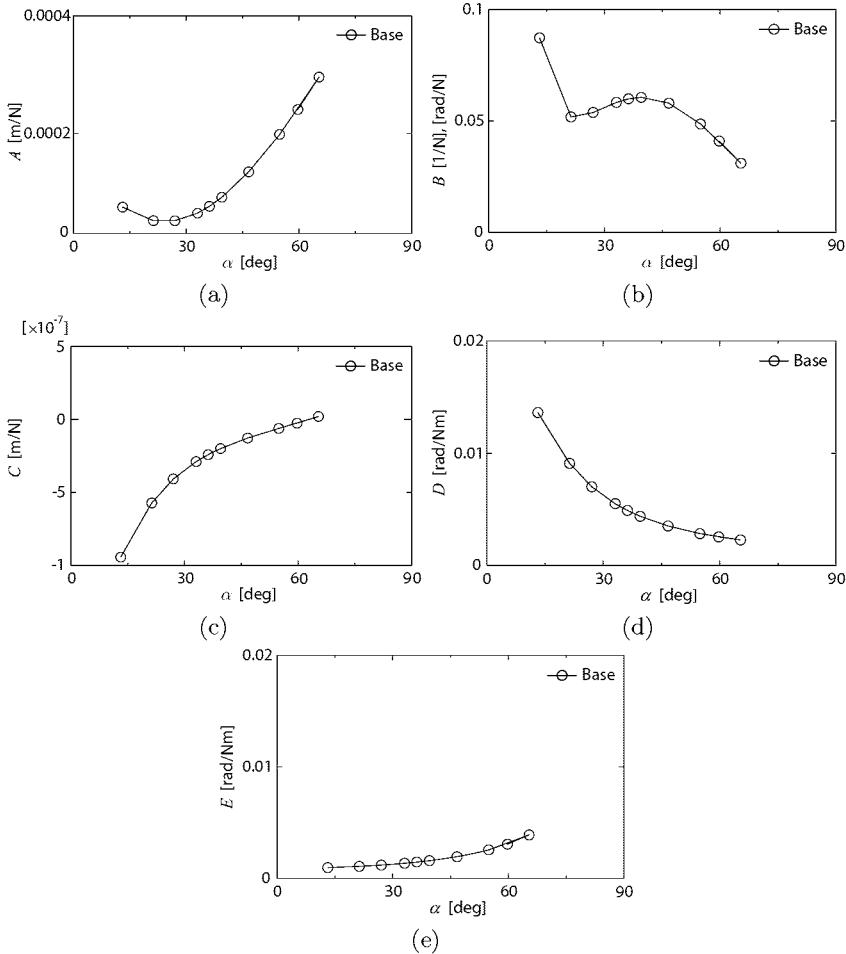


Fig. 16. Elements of the tip compliance matrix as a function of the parameter α

- The compliance of rotation around x - and y -axes against y and x forces, respectively, (element B) decreases for the most part of α , although it increases for a while at about 50 degrees.
- The compliance in the direction of x - and y -axes against y and x moments, respectively, (element B) changes in the same manner.
- The compliance in the direction of z -axis (element C) increases.
- The compliance of rotation around x - and y -axes (element D) decreases.
- The compliance of rotation around z -axis (element E) increases.

Therefore, in order to obtain a well-balanced stiffness, it is necessary to limit the value of α properly in the workspace.

4.5 A Design Index for the Modified DELTA Mechanism

From the above results, we find that the tip stiffness of the modified DELTA mechanism changes largely depending on the configuration, represented by the parameter α , even if elastic parameters are fixed. Therefore, we may use α as a design index. We should notice that the value of α depends on the base radius, the traveling plate radius, the arm length and the rod length. According to the results of numerical calculation, the value of α in the range from 40 to 70 degrees is best for realizing a well-balanced tip stiffness. If the value of α is outside this range, the stiffness of many elements decreases.

Based on the discussion, we decide that both the traveling plate and the base radii are 40 mm, the arm length is 93 mm and the rod length is 127 mm, in order to obtain a good balance of stiffness in the specified workspace (around a sphere of 75 mm radius).

5 Conclusions

We have presented a design of a compact haptic device in which parallel mechanisms are utilized and a large workspace of orientational motion is realized. The device is a parallel-serial mechanism consisting of a modified DELTA mechanism for translational motion and a spatial five-bar gimbal mechanism for orientational motion. We have derived an analytical model of stiffness for the modified DELTA mechanism, which we have utilized for the design of stiff platform for translational motion. The model shows that the compliance matrix is a function of kinematic parameters as well as elastic parameters of each element. Configuration dependency of the compliance matrix is an important point to be noticed. Key points newly proposed in the stiffness model are:

- Exploitation of stiffness analysis method for a flexible arm (manipulator) to obtain stiffness of the elementary chains in deriving the tip stiffness of the parallel mechanism.
- Modeling of the free motion around the axis of rotation in a bearing using a very small value of the elasticity coefficient.

We have obtained the following results regarding the design of the modified DELTA mechanism:

- The angle α can be a design index to optimize the stiffness of the modified DELTA mechanism.
- The stiffness of the bearings 2 and 3 should be sufficiently large.

From these results, we have found that α be restricted within the value between 40 to 70 degrees in order to obtain a well-balanced stiffness.

Future research will be directed to design of a more compact haptic device with higher frequency response using an actuator with faster response and with less friction.

References

1. Massie TH, Salisbury JK (1994) The PHANToM haptic interface: a device for probing virtual objects. In: Proc. 1994 ASME Int. Mechanical Engineering Exposition and Congress, Chicago, Illinois. pp 295–302
2. Colgate JE, Peshkin MA, Wannasuphoprasit W (1996) Nonholonomic haptic display. In: Proc. 1996 IEEE Int. Conf. on Robotics and Automation, Minneapolis, Minnesota. pp 539–544
3. Berkelman PJ, Hollis RL, Salcudean SE (1995) Interacting with virtual environments using a magnetic levitation haptic interface. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Pittsburgh, Pennsylvania. pp 117–122
4. Hirata Y, Sato M (1992) 3-dimensional interface device for virtual workspace. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Raleigh, North Carolina. pp 889–896
5. Inoue H, Tsusaka Y, Fukuzumi T (1986) Parallel manipulator. In: Faugeras O, Giralt G (eds) Robotics research, the third international symposium. The MIT Press. pp 321–327.
6. Hayward V (1995) Toward a seven axis haptic device. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Pittsburgh, Pennsylvania. pp 133–139
7. Gosselin C (1990) Stiffness mapping for parallel manipulator. *IEEE Trans. on Robotics and Automation* 6:377–382
8. Huang T, Zhao X, Whitehouse DJ (2002) Stiffness estimation of a tripod-based parallel kinematic machine. *IEEE Trans. on Robotics and Automation* 18:50–58
9. Tsumaki Y, Naruse H, Nenchev DN, Uchiyama M (1998) Design of a compact 6-DOF haptic interface. In: Proc. 1998 IEEE Int. Conf. on Robotics and Automation, Leaven, Belgium. pp 2580–2585
10. Yoon WK, Suehiro T, Tsumaki Y, Uchiyama M (2004) Stiffness analysis and design of a compact modified delta parallel mechanism. *ROBOTICA* 22:463–475
11. Merlet JP (2000) Parallel robots. Kluwer Academic Publishers
12. Uchiyama M (1994) Structures and characteristics of parallel manipulators. *Advanced Robotics* 8:545–557
13. Clavel R (1988) DELTA, a fast robot with parallel geometry. In: Proc. 18th Int. Symp. on Industrial Robots, Lausanne, Switzerland. pp 91–100
14. Tsai LW (1995) Multi-degree-of-freedom mechanisms for machine tools and like. U.S. Patent No. 5656905
15. Ouerfelli M, Kumar V (1994) Optimization of a spherical five-bar parallel drive linkage. *Trans. ASME, J. of Mechanical Design* 116:166–173
16. Gosselin CM, Hamel JF (1994) Development and experimentation of a fast three-degree-of-freedom spherical parallel manipulator. In: Jamshidi M, Yuh J, Nguyen C, Lumia R (eds) Proc. First World Automation Congress, Maui, Hawaii. TSI Press. 2:229–234
17. Komatsu T, Uenohara M, Iikura S, Miura H, Shimoyama I (1990) Compliance control for a two-link flexible manipulator. *Trans. JSME, Series C* 56:2642–2648 (in Japanese)
18. Svinin MM, Uchiyama M (1994) Contribution to inverse kinematics of flexible robot arms. *JSME Int. J., Series C: Dynamics, Control, Robotics, Design and Manufacturing* 37:755–764

Hybrid Nanorobotic Approaches to NEMS

B. J. Nelson¹, L. X. Dong², A. Subramanian³, and D. J. Bell⁴

¹ Swiss Federal Institute of Technology (ETH), Zurich bnelson@ethz.ch

² Swiss Federal Institute of Technology (ETH), Zurich ldong@ethz.ch

³ Swiss Federal Institute of Technology (ETH), Zurich arun@ethz.ch

⁴ Swiss Federal Institute of Technology (ETH), Zurich dbell@ethz.ch

Robotic manipulation at the nanometer scale is a promising technology for structuring, characterizing and assembling nano building blocks into nanoelectromechanical systems (NEMS). Combined with recently developed nanofabrication processes, a hybrid approach to building NEMS from individual carbon nanotubes (CNTs) and SiGe/Si nanocoils is described. Nanosensors and nanoactuators are investigated from experimental, theoretical, and design perspectives.

1 Introduction

Despite the claims of many "futurists," the form nanorobots of the future will take and what tasks they will actually perform remain unclear. However, it is clear that nanotechnology is progressing towards the construction of intelligent sensors, actuators, and systems that are smaller than 100nm. These nanoelectromechanical systems (NEMS) will serve as both the tools to be used for fabricating future nanorobots as well as the components from which these nanorobots may be developed. Shrinking device size to these dimensions presents many fascinating opportunities such as manipulating nanoobjects with nanotools, measuring mass in femto-gram ranges, sensing forces at pico-Newton scales, and inducing GHz motion, among other new possibilities waiting to be discovered. These capabilities will, of course, drive the tasks that future nanorobots constructed by and with NEMS will perform.

The design and fabrication of NEMS is an emerging area being pursued by an increasing number of researchers. Just as with MEMS, NEMS design is inextricably linked to available fabrication techniques. However, though the development of microfabrication processes has become somewhat stable over the past decade, nanofabrication processes are still being actively pursued, and the design constraints generated by these processes are relatively unexplored. Two approaches to nanofabrication, top-down and bottom-up, have been identified by the nanotechnology research community, and the topic of

this paper is how these trends can be integrated through robotics resulting in new classes of NEMS devices.

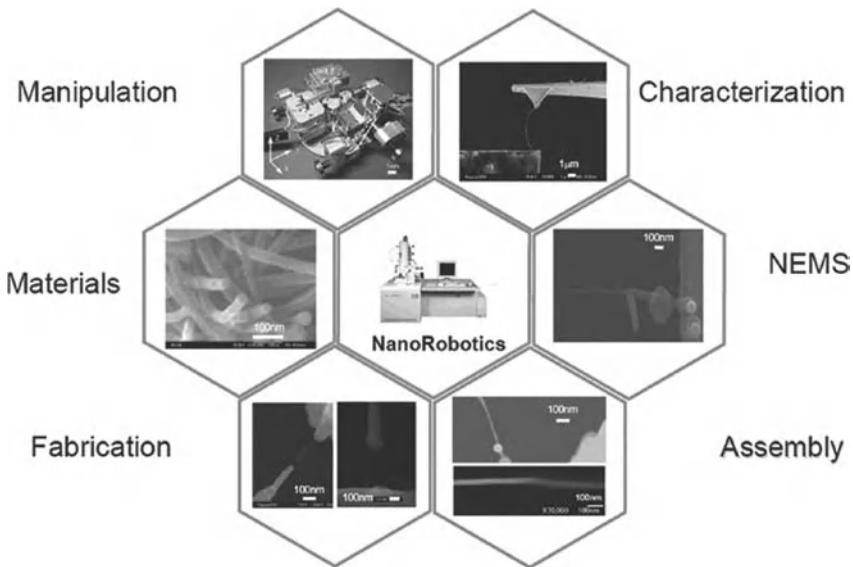


Fig. 1. A nanorobotic manipulation approach to NEMS

Top-down and bottom-up nanofabrication strategies are being independently investigated by various researchers. Top-down approaches are based on microfabrication and include technologies such as nano-lithography, nano-imprinting, and chemical etching. Presently, these are 2D fabrication processes with relatively low resolution. Bottom-up strategies are assembly-based techniques. Currently these strategies include techniques such as self-assembly, dip-pen lithography, and directed self-assembly. These techniques can generate regular nano patterns at large scales. With the ability to position and orient nanometer scale objects, nanorobotic manipulation is an enabling technology for structuring, characterizing and assembling many types of nanosystems (shown in Fig. 1) [1]. By combining top-down (Fig. 2(a)) and bottom-up processes (Fig. 2(b)), a hybrid nanorobotic approach (Fig. 2(c)) based on nanorobotic manipulation provides a third way to fabricate NEMS by structuring as-grown nanomaterials or nanostructures. In this system, nanofabrication based top-down processes and nanoassembly based bottom-up processes can be performed in an arbitrary order. Consider nanofabrication processes in which nanomaterials or nanostructures can be fabricated into nano building blocks by removing unwanted parts. These building blocks can then be assembled into NEMS. Conversely, nanoassembly can be performed first and nanomaterials or nanostructures can be assembled into higher-level (i.e. more

complex, 3D, arrays, etc.) structures, and then the high-level structures can be further modified into NEMS by nanofabrication.

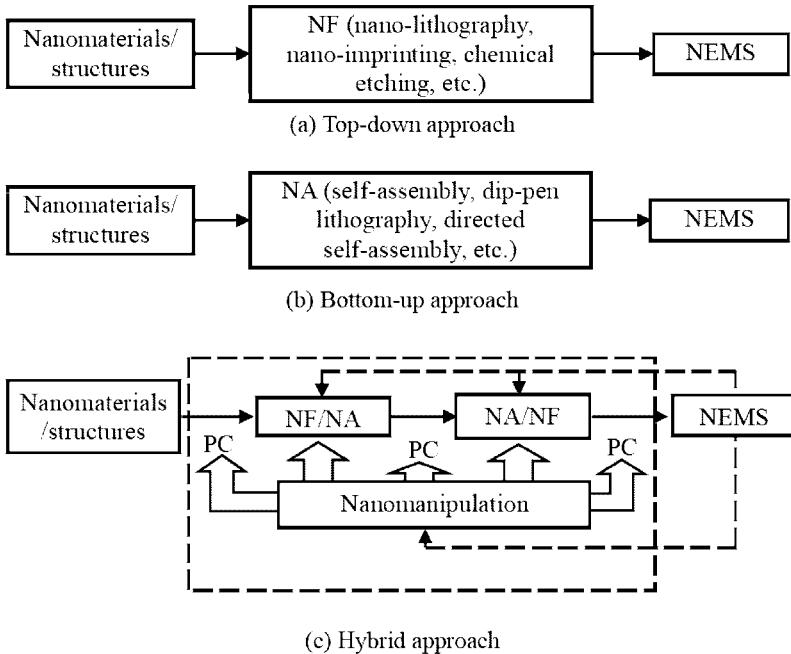


Fig. 2. Approaches to NEMS (PC: Property Characterization, NF: Nano Fabrication, NA: Nano Assembly)

Nanorobotic manipulation enables this hybrid approach for creating NEMS that can attain a higher functionality because they possess more complex structures. Moreover, property characterization can be performed after intermediate processes, and *in situ* active characterization can be performed using manipulation rather than conventional static observations. The impact of the hybrid approach on robotics is twofold: it expands the lower limit of robotic exploration further into the nanometer scale, and it will provide nanoscale sensors and actuators and assembly technology for building nanorobots. Nanomaterial science, bionanotechnology, and nanoelectronics will also benefit from advances in this new nanomanufacturing technique from the perspectives of property characterization, fabrication and assembly. This paper introduces carbon nanotubes (CNTs) and nanocoils in Section 2. In Sections 3 and 4, the assembly of individual nanotubes and nanocoils into NEMS are presented along with characterization results.

2 Carbon Nanotubes and Nanocoils for NEMS

Carbon nanotubes [2] and nanocoils have been used as base materials and structures because of their exceptional properties and unique structures. For NEMS, some of the most important characteristics of nanotubes include their nanometer diameter, large aspect ratio (10-1000), TPa scale Young's modulus, excellent elasticity, ultra-small interlayer friction, sensitivity of conductance to various physical or chemical changes, and charge-induced bond-length change. Helical 3-D nanostructures, or nanocoils, have been synthesized from different materials including helical carbon nanotubes [3] and zinc oxide nanobelts [4]. A new method of creating structures with nanometer-scale dimensions has recently been presented [5] and can be fabricated in a controllable way [6]. The structures are created through a top-down fabrication process in which a strained nanometer thick heteroepitaxial bilayer curls up to form 3-D structures with nanoscale features. Helical geometries and tubes with diameters between 10nm and 10 μ m have been achieved. Because of their interesting morphology, mechanical, electrical, and electromagnetic properties, potential applications of these nanostructures in NEMS include nanosprings [7], electromechanical sensors [8], magnetic field detectors, chemical or biological sensors, generators of magnetic beams, inductors, actuators, and high-performance electromagnetic wave absorbers. NEMS based on individual single- or multi-walled carbon nanotubes (SWNTs, MWNTs) and nanocoils are of increasing interest, indicating that capabilities for incorporating these individual building blocks at specific locations on a device must be developed.

Random spreading [9], direct growth [10], self-assembly [11], dielectrophoretic assembly [12] and nanomanipulation [13] have been demonstrated for positioning as-grown nanotubes on electrodes for the construction of these devices. However, for nanotube-based structures, nanorobotic assembly is still the only technique capable of *in situ* structuring, characterization and assembly. Because the as-fabricated nanocoils are not free-standing from their substrate, nanorobotic assembly is virtually the only way to incorporate them into devices at present.

3 Individual Nanotube Based NEMS

Basic techniques for the nanorobotic manipulation of carbon nanotubes are shown in Fig. 3 [1]. These serve as the basis for handling, structuring, characterizing and assembling NEMS. Configurations of nanotools, sensors, and actuators based on individual nanotubes that have been experimentally demonstrated are summarized as shown in Fig. 4.

For detecting deep and narrow features on a surface, cantilevered nanotubes (Fig. 3(a), [15]) have been demonstrated as probe tips for an atomic force microscope (AFM) [16], a scanning tunneling microscope (STM) and

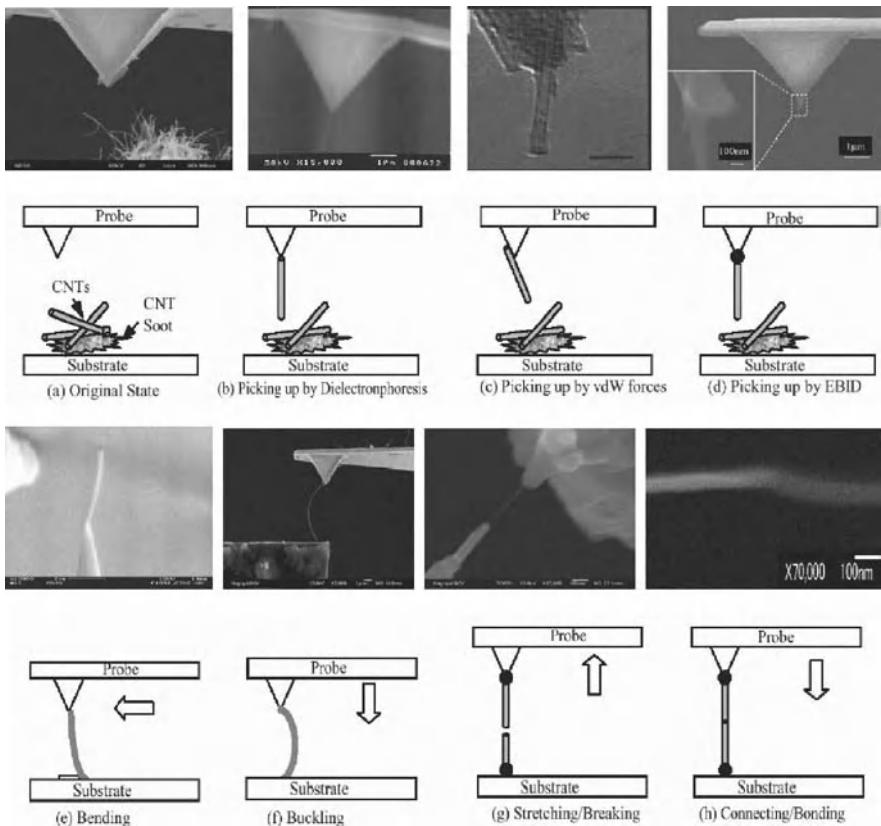


Fig. 3. Nanorobotic manipulation of CNTs. The basic technique is to pick up an individual tube from CNT soot (as in (a)) or from an oriented array; (b) shows a free-standing nanotube picked up by dielectrophoresis generated by a non-uniform electric field between the probe and substrate, (c) (from [14]) and (d) show the same manipulation by contacting a tube with the probe surface or fixing (e.g. with EBID) a tube to the tip. Vertical manipulation of nanotubes includes bending (e), buckling (f), stretching/breaking (g), and connecting/bonding (h). All examples with the exception of (c) are from the authors' work.

other types of scanning probe microscopes (SPM). Nanotubes provide ultra-small diameters, ultra-large aspect ratios, and excellent mechanical properties. Manual assembly, direct growth and nanoassembly have proven effective for their construction. Cantilevered nanotubes have also been demonstrated as probes for the measurement of ultra-small physical quantities, such as femtogram mass [17], mass flow sensors [18], and pico-Newton order force sensors [18] on the basis of their static deflections or change of resonant frequencies detected within an electron microscope. Deflections cannot be measured from micrographs in real-time limiting the application of these types of sensors.

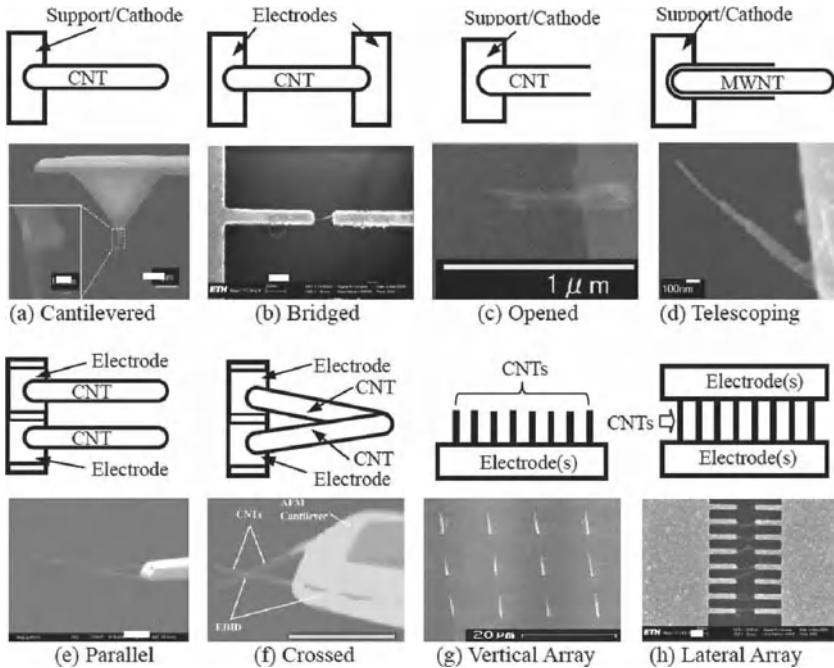


Fig. 4. Configurations of individual nanotube-based NEMS. Scale bars: (a) $1\mu\text{m}$ (inset: 100nm), (b) 200nm , (c) $1\mu\text{m}$, (d) 100nm , (e) and (f) $1\mu\text{m}$, (g) $10\mu\text{m}$, and (h) 300nm . All examples are from the authors' work.

Inter-electrode distance changes cause emission current variation of a nanotube emitter and may serve as a candidate to replace microscope images [18].

Bridged individual nanotubes (Fig.3(b), [19]) have been the basis for electric characterization. A nanotube based gas sensor design has adopted this configuration [20].

Opened nanotubes (Fig.3(c), [21]) can serve as an atomic or molecular container. A thermometer based on this structure has been demonstrated by monitoring the height of the gallium inside the nanotube using transmission electron microscopy (TEM) [22].

Bulk nanotubes can be used to fabricate actuators based on charge injection induced bond-length change [23], and, theoretically, individual nanotubes also work on the same principle. Electro-static deflection of a nanotube has been used to construct a relay [24]. A new family of nanotube actuators can be constructed by taking advantage of the ultra-low inter-layer friction of a multi-walled nanotube. Linear bearings based on telescoping nanotubes have been demonstrated [25],[18]. Recently, a micro actuator with a nanotube as a rotation bearing has been demonstrated [26]. A preliminary experiment on

a promising nanotube linear motor with field emission current serving as position feed back has been shown with nanorobotic manipulation (Fig. 3(d), [21]).

Cantilevered dual nanotubes have been demonstrated as nanotweezers [27] and nanoscissors (Fig. 3(e)) [13] by manual and nanorobotic assembly, respectively.

Based on electric resistance change under different temperatures, nanotube thermal probes (Fig. 3(f), [18]) have been demonstrated for measuring the temperature at precise locations. These thermal probes are more advantageous than nanotube based thermometers because the thermometers require TEM imaging. The probes also have better reproducibility than devices based on dielectrophoretically assembled bulk nanotubes [28]. Gas sensors and hot-wire based mass/flow sensors can also be constructed in this configuration rather than a bridged one.

The integration of the above mentioned devices can be realized using the configurations shown in Fig. 3(g) [29] and (h) [12]. The arrays of individual nanotubes can also be used to fabricate nanosensors, such as position encoders [30].

Nanotube based NEMS remains a rich research field with a large number of open problems. New materials and effects at the nanoscale will enable a new family of sensors and actuators for the detection and actuation of ultra-small quantities or objects with ultra-high precision and frequencies. Through random spreading, direct growth, and nanorobotic manipulation, proto-types have been demonstrated. However, for integration into NEMS, self-assembly processes will become increasingly important. Among them, we believe that dielectrophoretic nanoassembly will play a significant role for large scale production of 2D regular structures [31].

4 NEMS Made from Nanocoils

The construction of NEMS using nanocoils involves the assembly of as-grown or as-fabricated nanocoils, which is a significant challenge from a fabrication standpoint. Focusing on the unique aspects of manipulating nanocoils due to their helical geometry, high elasticity, single end fixation, and strong adhesion of the coils to the substrate for wet etching, a series of new processes is presented using a manipulator (MM3A, Kleindiek) installed in an SEM (Zeiss DSM962). As-fabricated SiGe/Si bilayer nanocoils are shown in Fig. 5. Special tools have been fabricated including a nanohook prepared by controlled "tip-crashing" of a commercially available tungsten sharp probe (Picoprobe T-1-10-1mm and T-1-10) onto a substrate, and a "sticky" probe prepared by tip dipping into a double-sided SEM silver conductive tape (Ted Pella, Inc.). As shown in Fig. 6, experiments demonstrate that nanocoils can be released from a chip by lateral pushing, picked up with a nanohook or a "sticky" probe, and placed between the probe/hook and another probe or

an AFM cantilever (Nano-probe, NP-S). Axial pulling/pushing, radial compressing/releasing, and bending/buckling have also been demonstrated. These processes have shown the effectiveness of manipulation for the characterization of coil-shaped nanostructures and their assembly for NEMS, which have been otherwise unavailable.

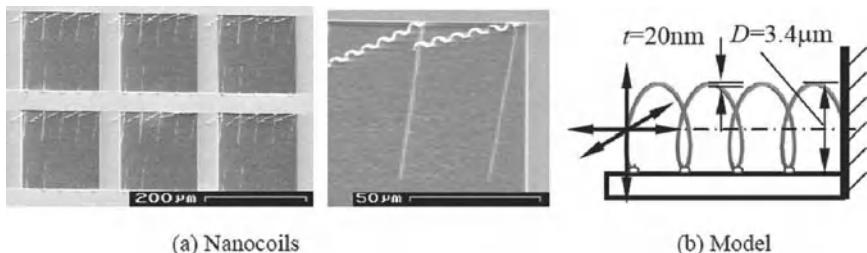


Fig. 5. As-fabricated nanocoils (Thickness: $t=20\text{nm}$ (without Cr layer) or 41nm (with Cr layer). Diameter: $D=3.4\mu\text{m}$)

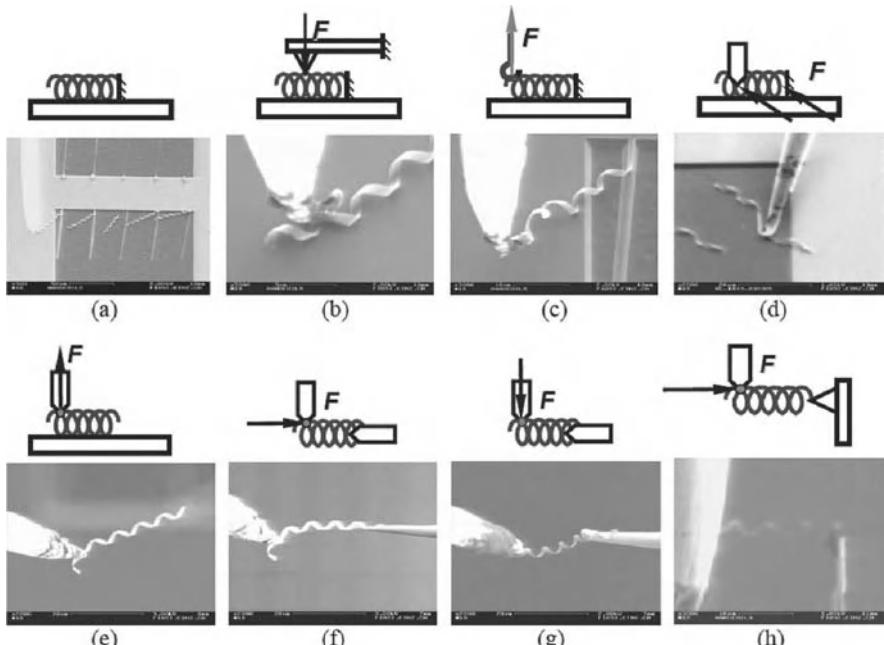


Fig. 6. Nanorobotic manipulation of nanocoils (a) original state, (b) compressing/releasing, (c) hooking, (d) lateral pushing/breaking, (e) picking, (f) placing/inserting, (g) bending, and (h) pushing and pulling

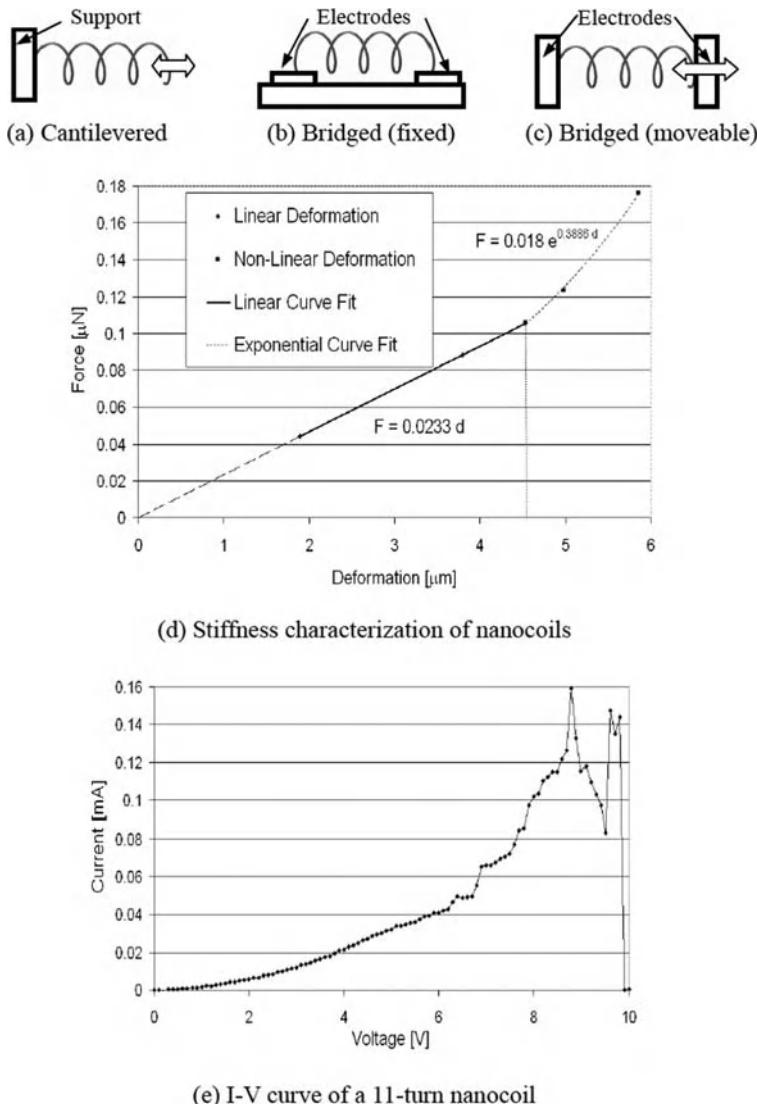


Fig. 7. Nanocoil based devices. Cantilevered nanocoils (a) can serve as nanosprings. Nanoelectromagnets, chemical sensors, and nanoinductors use nanocoils bridged between two electrodes (b). Electromechanical sensors can use a similar configuration but with one end connected to a moveable electrode. Mechanical stiffness (d) and electric conductivity (e) are basic properties of interest for these devices.

Configurations of nanodevices based on individual nanocoils are shown in Fig. 7. Cantilevered nanocoils as shown in Fig. 7(a) can serve as nanosprings.

Nanoelectromagnets, chemical sensors and nanoinductors involve nanocoils bridged between two electrodes as shown in Fig. 7(b). Electromechanical sensors can use a similar configuration but with one end connected to a moveable electrode as shown in Fig. 7(c). Mechanical stiffness and electric conductivity are fundamental properties for these devices that must be further investigated.

As shown in Fig. 6(h), axial pulling is used to measure the stiffness of a nanocoil. A series of SEM images are analyzed to extract the AFM tip displacement and the nanospring deformation, i.e. the relative displacement of the probe from the AFM tip. From this displacement data and the known stiffness of the AFM cantilever, the tensile force acting on the nanospring versus the nanospring deformation was plotted. The deformation of the nanospring was measured relative to the first measurement point. This was necessary because the proper attachment of the nanospring to the AFM cantilever must be verified. Afterwards, it was not possible to return to the point of zero deformation. Instead, the experimental data as presented in Fig. 7(d) has been shifted such that with the calculated linear elastic spring line begins at zero force and zero deformation. From Fig. 7(d), the stiffness of the spring was estimated to be 0.0233 N/m. The linear elastic region of the nanospring extends to a deformation of 4.5 μm . An exponential approximation was fitted to the nonlinear region. When the applied force reached 0.176 μN , the attachment between the nanospring and the AFM cantilever broke. Finite element simulation (ANSYS 9.0) was used to validate the experimental data [8]. Since the exact region of attachment cannot be identified according to the SEM images, simulations were conducted for 4, 4.5, and 5 turns to obtain an estimate of the possible range according to the apparent number of turns of the nanospring of between 4 and 5. The nanosprings in the simulations were fixed on one end and had an axial load of 0.106 μN applied on the other end. For the simulation results for the spring with 4 turns, the stiffness from simulation is 0.0302 N/m. For the nanospring with 5 turns it is 0.0191 N/m. The measured stiffness falls into this range with 22.0% above the minimum value and 22.8% below the maximum value, and very close to the stiffness of a 4.5-turn nanospring that has a stiffness of 0.0230 N/m according to simulation.

Fig. 7(e) shows the results from electrical characterization experiments on a nanospring with 11 turns using the configuration as shown in Fig. 6(g). The I-V curve is non-linear, which may be caused by the resistance change of the semiconductive bilayer due to ohmic heating. Another possible reason is the decrease in contact resistance caused by thermal stress. The maximum current was found to be 0.159 mA under an 8.8 V bias. Higher voltage causes the nanospring to "blow off." From the fast scanning screen of the SEM, an extension of the nanospring on probes was observed around the peak current so that the current does not drop abruptly. At 9.4 V, the extended nanospring is broken down, causing an abrupt drop in the I-V curve.

From fabrication and characterization results, the helical nanostructures appear to be suitable for inductors. They would allow further miniaturization compared to state-of-the-art micro inductors. For this purpose, higher dop-

ing of the bilayer and an additional metal layer would result in the required conductance. Conductance, inductance, and quality factor can be further improved if, after curling up, additional metal is electroplated onto the helical structures. Moreover, a semiconductive helical structure, when functionalized with binding molecules, can be used for chemical sensing under the same principle as demonstrated with other types of nanostructures [32]. With bilayers in the range of a few monolayers, the resulting structures would exhibit very high surface-to-volume ratio with the entire surface exposed to an incoming analyte.

5 Conclusions

A hybrid nanofabrication approach based on nanorobotic manipulation has been investigated for building NEMS. Processes for manipulating carbon nanotubes and SiGe/Si bilayer nanocoils have been developed, demonstrating their effectiveness for handling, structuring, and characterizing nanomaterials and nanostructures, and for assembling them into NEMS. An overview of NEMS made from individual nanotubes and nanocoils has been presented. A hybrid approach based on nanorobotic manipulation provides the possibility for *in situ* active property characterization, structuring and assembly of nanomaterials and nanostructures. The approach enables the construction of NEMS sensors and actuators and, eventually, nanorobots.

References

1. Dong LX (2003) Nanorobotic manipulations of carbon nanotubes, Ph.D Dissertation, Nagoya University, Nagoya
2. Iijima S (1991) Nature 354: 56-58
3. Zhang X B, Bernaerts D, Van Tendeloo G, Amelincks S, Van Landuyt J, Ivanov V, Nagy J B, Lambin Ph, Lucas A A (1994) Europhys Lett 27: 141-146
4. Kong X Y, Wang Z L (2003) Nano Lett 3: 1625-1631
5. Golod S V, Prinz V, Mashanov V I, Gutakovskiy A K (2001) Semicond Sci Technol 16: 181-185
6. Zhang L, Deckhardt E, Weber A, Schonenberger C, Grutzmacher D (2005) Nanotechnology 16: 655-663
7. Bell D J, Dong L X, Nelson B J (2005) Manipulation of nanocoils for nanoelectromagnets, Proc. of the 5th IEEE Conference on Nanotechnology, Nagoya, July 11-15
8. Bell D J, Sun Y, Zhang L, Dong L X, Nelson B J, Grutzmacher D (2005) Three-dimensional nanosprings for electromechanical sensors, In: Proc. of the 13th Int. Conf. on Solid-State Sensors, Actuators and Microsystems, Seoul, pp.15-18, June 5-9
9. Martel R, Schmidt T, Shea H R, Hertel T, Avouris Ph (1998) Appl Phys Lett 73: 2447-2449

10. Franklin N R, Li Y M, Chen R J, Javey A, Dai H J (2001) *Appl Phys Lett* 79: 4571-4573
11. Rueckes T, Kim K, Joselevich E, Tseng G Y, Cheung C-L, Lieber C M (2000) *Science* 289: 94-97
12. Subramanian A, Vikramaditya B, Dong L X, Bell D J, Nelson B J (2006) Micro and nanorobotic assembly using dielectrophoresis, In: *Robotics: Science and Systems I*, MIT Press,
13. Fukuda T, Arai F, Dong L X (2003) *Proceedings of the IEEE* 91: 1803-1818
14. Hafner J H, Cheung C-L, Oosterkamp T H, Lieber C M (2001) *J. Phys. Chem. B* 105: 743-746
15. Dong L X, Arai F, Fukuda T (2002) *Appl Phys Lett* 81: 1919-1921
16. Dai H J, Hafner J H, Rinzler A G, Colbert D T, Smalley R E (1996) *Nature* 384: 147-150
17. Poncharal P, Wang Z L, Ugarte D, de Heer W A (1999) *Science* 283: 1513-1516
18. Fukuda T, Arai F, Dong L X, Imaizumi Y (2004) Prospective of nanotube sensors and nanotube actuators, In: *Proc. of the 4th IEEE Int. Conf. on Nanotechnology*, Munich, Aug.17-19
19. Subramanian A, Nelson B J, Dong L X, Bell D J (2005) Dielectrophoretic nano-assembly of nanotube-based NEMS with nanoelectrodes, In: *Proc. of the 6th IEEE Int. Symp. on Assembly and Task Planning*, Montreal, July 19-21
20. Kong J, Franklin N R, Zhou C W, Chapline M G, Peng S, Cho K J, Dai H J (2000) *Science* 287: 622-625
21. Dong L X, Nelson B J, Fukuda T, Arai F (2005) Towards linear nano servomotors with integrated position sensing, In: *Proc. of the 2005 IEEE Int'l Conf. on Robotics and Automation*, pp.867-872, Barcelona, April 18-22
22. Gao Y H, Bando Y (2002) *Nature* 415: 599
23. Baughman R H, Cui C X, Zakhidov A A, Iqbal Z, Barisci J N, Spinks G M, Wallace G G, Mazzoldi A, De Rossi D, Rinzler A G, Jaschinski O, Roth S, Kertesz M (1999) *Science* 284: 1340-1344
24. Lee S W, Lee D S, Morjan R E, Jhang S H, Svenningsson M, Nerushev O A, Park Y W, Campbell E E B (2004) *Nano Lett* 4: 2027-2030
25. Cumings J, Zettl A (2000) *Science* 289: 602-604
26. Fennimore A M, Yuzvinsky T D, Han W-Q, Fuhrer M S, Cumings J, Zettl A (2003) *Nature* 424: 408-410
27. Kim P, Lieber C M (1999) *Science* 286: 2148-2150
28. Fung C K M, Wong V T S, Chan R H M, Li W J (2004) *IEEE Trans Nanotech* 3: 395-403
29. Subramanian A, Dong L X, Nelson B J (2005) Selective eradication of individual nanotubes from vertically aligned arrays, In: *Proc. of 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey, pp.105-110, Monterey, July 24- 28
30. Dong L X, Subramanian A, Nelson B J (2005) Nano encoders based on arrays of single nanotube emitters, In: *Proc. of the 5th IEEE Conference on Nanotechnology*, Nagoya, July 11-15
31. Nelson B J, Subramanian A, Vikramaditya B, Dong L X , Bell D J (2005) Micro and Nano Assembly Using Dielectrophoretic Forces, In: *Proc. of the 3rd Int. Conf. on Microchannels and Minichannels (ICMM2005)*, Toronto, Ontario, Canada, June 13-15 (Keynote speech)
32. Cui Y, Wei Q Q, Park H K, Lieber C M (2001) *Science* 293: 1289-1292

SLAM

Jacobian, Manipulability, Condition Number and Accuracy of Parallel Robots

J.-P. Merlet¹

INRIA, BP 93, 06902 Sophia-Antipolis, France

Jean-Pierre.Merlet@sophia.inria.fr

1 Introduction

Parallel robots are nowadays leaving academic laboratories and are finding their way in an increasingly larger number of application fields such as telescopes, fine positioning devices, fast packaging, machine-tool, medical application. A key issue for such use is optimal design as performances of parallel robots are very sensitive to their dimensioning. Optimal design methodologies have to rely on kinetostatic performance indices and accuracy is clearly a key-issue for many applications. It has also be a key-issue for serial robots and consequently this problem has been extensively studied and various accuracy indices have been defined. The results have been in general directly transposed to parallel robots. We will now review how well these indices are appropriate for parallel robots.

2 Jacobian and Inverse Jacobian Matrix

Let \mathbf{X}_a denotes the generalized coordinates of the end-effector composed of parameters describing the available n d.o.f. of the end-effector while \mathbf{X} denotes all the generalized coordinates of the end-effector. We will impose no constraints on the choice of \mathbf{X} (e.g. for a Gough robot with a planar platform the pose may be represented by the 9 coordinates of 3 particular points on the end-effector).

The geometry of the robot is described by its joints variables vector Θ . The twist \mathbf{W} of the end effector is composed of its translational and angular velocities and the *restricted twist* \mathbf{W}_a is defined as the restriction of \mathbf{W} to the available d.o.f. of the robot. It is well known that for robot having at least 2 rotational d.o.f. \mathbf{W} is not the time-derivative of \mathbf{X} as there is no representation of the orientation whose derivatives corresponds to the angular velocities. However there exists usually a matrix H such that

$$\mathbf{W} = \mathbf{H}\dot{\mathbf{X}} \quad (1)$$

In the usual approach the jacobian matrix J_k linearly relates the actuated joint velocities Θ_a to \mathbf{W}_a :

$$\mathbf{W}_a = J_k \dot{\Theta}_a \quad (2)$$

In this paper we consider only non-redundant robots so that matrix J_k is square and we will call it the *kinematic jacobian*. A feature of parallel robots is that it is usually easy to establish an analytical form for J_k^{-1} while it is often impossible to obtain J_k .

But we may also define other jacobian matrices by first changing the parameters in Θ . Indeed parallel robots differ from their serial counterpart by a larger number of passive joints and it may thus be interesting to include the m passive joints variables Θ_p . If Θ is defined as (Θ_a, Θ_p) we may then define write the l inverse kinematics equations as $\mathbf{F}(\Theta, \mathbf{X}_a) = 0$ from which we derive

$$\frac{\partial \mathbf{F}}{\partial \Theta} \dot{\Theta} + \frac{\partial \mathbf{F}}{\partial \mathbf{X}_a} \dot{\mathbf{X}}_a = \mathbf{U} \dot{\Theta} + \mathbf{V}_a \dot{\mathbf{X}}_a = 0 \quad (3)$$

where \mathbf{U} is $(l \times (n+m))$ and \mathbf{V}_a is $(l \times n)$. This relation allows to quantify the influence of the measurement errors on the passive and actuated joints variables on the positioning errors $\Delta \mathbf{X}_a$ on the n d.o.f. of the end-effector by using (1).

Although we say that some robot have $n < 6$ d.o.f., still the end-effector is a 6 d.o.f. rigid body and positioning errors on *all* d.o.f. should be examined. It is thus interesting to determine an inverse jacobian that involves the full twist \mathbf{W} of the end-effector. In that case we write the kinematics equations as $\mathbf{G}(\Theta, \mathbf{X}) = 0$. If we fix \mathbf{X} we know that these kinematics equations have a finite number of solutions, which implies that the number of equations in \mathbf{G} should be $n+m$. By differentiation we get:

$$\frac{\partial \mathbf{G}}{\partial \Theta} \dot{\Theta} + \frac{\partial \mathbf{G}}{\partial \mathbf{X}} \dot{\mathbf{X}} = \mathbf{A} \dot{\Theta} + \mathbf{B} \dot{\mathbf{X}} = 0 \quad (4)$$

where \mathbf{A} is a square $n+m \times n+m$ matrix while \mathbf{B} is $n+m \times 6$. Provided that \mathbf{H} is square and not singular we may now derive an inverse jacobian such that

$$\dot{\Theta} = -\mathbf{A}^{-1} \mathbf{B} \mathbf{H}^{-1} \mathbf{W} = \mathbf{J}^{-1} \mathbf{W} \quad (5)$$

where \mathbf{J}^{-1} is $n+m \times 6$. In most cases however a velocity analysis allows one to obtain a simpler inverse jacobian matrix through a relation that involves only Θ_a :

$$\begin{pmatrix} \dot{\Theta}_a \\ \mathbf{0} \end{pmatrix} = \mathbf{J}_{fk}^{-1} \mathbf{W} \quad (6)$$

where \mathbf{J}_{fk}^{-1} is $n+m \times 6$ and will be called the *full inverse kinematics jacobian*.

We may further extend this approach to take into account the design parameters \mathcal{P} of the robot (e.g. the location of the anchor points of the legs

in a Gough platform). For that purpose the kinematics equations will be written as $\mathbf{G}(\mathcal{P}, \boldsymbol{\Theta}, \mathbf{X}) = 0$ and the matrix of the partial derivatives of \mathbf{G} with respect to \mathcal{P} will allow one to quantify the influence of the errors on \mathcal{P} on the positioning error of the end-effector.

As may be seen there is not a single inverse jacobian matrix but a multiplicity of them. Note however an important property of the inverse jacobian J^{-1} of (5) with respect to J_{fk}^{-1} : the rank of J_{fk}^{-1} is the same than the rank of J^{-1} .

It is also important to note that any inverse jacobian involving the full twist of the end-effector \mathbf{W} will not be homogeneous in terms of units. This will be true also for the inverse kinematic jacobian for robot involving both translation and rotational d.o.f. for the end-effector. Consequently many matrix properties (such as the trace, determinant) will not be invariant under a change of units.

In this paper we will focus on the influence of $\Delta\boldsymbol{\Theta}_a$ on the positioning errors of the end-effector through J_{fk}^{-1} . The necessity of using the full inverse kinematic jacobian will be emphasized on an example.

2.1 Example: The 3 – *UPU* Robot

Tsai [10] has proposed this robot as a 3 d.o.f. translation robot (figure 1). Each leg of this robot is constituted, starting from the base, by a U joint followed by an extensible leg terminated by another U joint *whose axis are the same than the U joint on the base*. This constraint allows theoretically to obtain only translation for the end-effector. This example will allow us to establish a methodology for determining the full inverse kinematic jacobian. But it will also enable to show the importance of this matrix. The story is that such a robot was designed at Seoul National University (SNU) and that is was

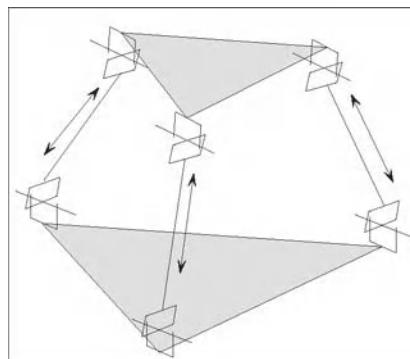


Fig. 1. The 3 – *UPU* robot

exhibiting a strange behavior: although the prismatic actuators were locked, the end-effector was exhibiting significant orientation motion. This phenomena was explained by Bonev and Zlatanov [1] and later in [2, 11]. Furthermore motion sensitivity to manufacturing tolerances has been studied [5, 8] and has shown that this robot was very sensitive.

We will denote by B_1, B_2, B_3 the center of the U joints on the platform and will now calculate the full inverse kinematic jacobian matrix. The velocity \mathbf{V}_B of the B points is $\mathbf{V}_B = \mathbf{V} + \mathbf{BC} \times \boldsymbol{\Omega}$. Let us define \mathbf{n} as the unit vector of the leg and compute the dot product of the right and left terms of the previous equation:

$$\mathbf{V}_B \cdot \mathbf{n} = \dot{\rho} \mathbf{n} = \mathbf{V} \cdot \mathbf{n} + (\mathbf{BC} \times \boldsymbol{\Omega}) \cdot \mathbf{n} = \mathbf{V} \cdot \mathbf{n} + (\mathbf{CB} \times \mathbf{n}) \cdot \boldsymbol{\Omega} \quad (7)$$

Now let us define $\mathbf{u}_i, \mathbf{v}_i$ the unit vectors of the two joint axis of the U joint at B_i . These vectors are the same for the base and platform. The angular velocity of the leg $\boldsymbol{\omega}_l$ with respect to the base and the angular velocity of the platform $\boldsymbol{\omega}_p$ with respect to the leg are

$$\boldsymbol{\omega}_l = \dot{\theta}_A^i \mathbf{u}_i + \dot{\alpha}_A^i \mathbf{v}_i \quad \boldsymbol{\omega}_p = \dot{\theta}_B^i \mathbf{u}_i + \dot{\alpha}_B^i \mathbf{v}_i$$

The angular velocity of the platform is

$$\boldsymbol{\Omega} = \boldsymbol{\omega}_l + \boldsymbol{\omega}_p = K_1^i \mathbf{u}_i + K_2^i \mathbf{v}_i$$

where K_1^i, K_2^i can be obtained from the previous equations. Now define $\mathbf{s}_i = \mathbf{u}_i \times \mathbf{v}_i$ and compute the dot product of the right and left terms of the previous equation by \mathbf{s}_i :

$$\mathbf{s}_i \cdot \boldsymbol{\Omega} = 0 \quad (8)$$

Combining equations (7, 8) we get the full velocities equations involving the twist \mathbf{W} as

$$\begin{pmatrix} \dot{\rho}_i \\ \mathbf{0} \end{pmatrix} = \mathbf{J}_{fk}^{-1} \mathbf{W} = \begin{pmatrix} \mathbf{n}_i & (\mathbf{CB}_i \times \mathbf{n}_i) \\ \mathbf{0} & \mathbf{s}_i \end{pmatrix} \mathbf{W} \quad (9)$$

which establish the full inverse kinematic jacobian. The inverse kinematic jacobian may be extracted from \mathbf{J}_{fk}^{-1} as the 3×3 matrix whose rows are the \mathbf{n}_i vectors. But an important point for accuracy analysis is to consider the lower part of \mathbf{J}_{fk}^{-1} which shows that if $\mathbf{s}_1 \cdot (\mathbf{s}_2 \times \mathbf{s}_3) = 0$ the platform may exhibit orientation motion that may be infinitesimal or finite according to the geometry of the U joint. It happens that the design of the SNU robot was in the latter category.

3 Manipulability

It is realistic to assume that the joint errors are bounded and consequently so will be the positioning errors. The norm of the bound may be chosen arbitrary

as (6) is linear so that a simple scaling will allow to determine the positioning error from the errors obtained for a given bound. A value of 1 for the bound is usually chosen:

$$\|\Delta\Theta\| \leq 1 \quad (10)$$

which leads to

$$\Delta X^T J^{-T} J^{-1} \Delta X \leq 1 \quad (11)$$

A classical geometrical interpretation of this relation is presented for the 2D case in figure 2. If the Euclidean norm is used (10) represents a circle in the joints errors space. This circle is mapped through matrix $J^{-T} J^{-1}$ into an ellipse in the generalized coordinates error space. More generally the mapping transform the hyper-sphere of the joints errors space into an ellipsoid, usually called the *manipulability ellipsoid*.

In fact the use of the Euclidean norm is not realistic: it implies for example that if one of the joint error is 1, then by some mysterious influence all the other joint errors are 0. The appropriate norm is the infinity norm that states that the absolute value of the joint errors are independently bounded by 1. With this norm (10) represents a n -dimensional square in the joints errors space that is mapped into the *kinematics polyhedron*, that includes the manipulability ellipsoid, in the generalized coordinates errors space. Figure 2 illustrates this mapping in the 2D case. It must be noted that, apart of being more realistic, the previous mapping leads to geometrical object that can

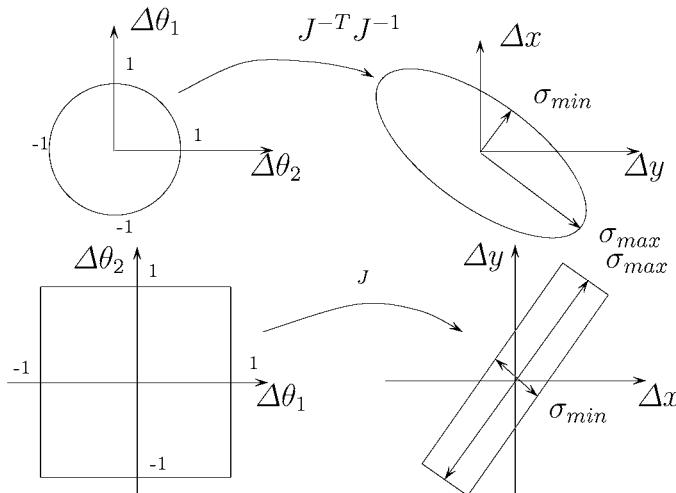


Fig. 2. The mapping between the joints errors space and the generalized coordinates error space induced by $J^{-T} J$ according to the norm: on top the Euclidean norm and on bottom the infinity norm.

be more easily manipulated than the ellipsoid. For example assume that one want to determine what are all the possible end-effector velocities that can be obtained in 2 different poses of the end-effector. For that purpose we will have to calculate the intersection of the 2 polyhedra obtained for the 2 poses, a well known problem of computational geometry, that can be much more easily solved than computing the intersection of 2 ellipsoids.

4 Condition Number

A large dimension along a given axis of the kinematics polyhedron indicates a large amplification error. It is therefore necessary to quantify this amplification factor. Let us consider the linear system:

$$\mathbf{J}^{-1} \Delta \mathbf{X} = \Delta \boldsymbol{\Theta},$$

where \mathbf{J}^{-1} is a $n \times n$ inverse kinematic jacobian matrix. The error amplification factor in this system expresses how a relative error in $\boldsymbol{\Theta}$ gets multiplied and leads to a relative error in \mathbf{X} . It characterize in some sense the dexterity of the robot and will be used as a performance index. We now use a norm such that

$$\|\mathbf{J}^{-1} \Delta \mathbf{X}\| \leq \|\mathbf{J}^{-1}\| \|\mathbf{J}\| \|\Delta \mathbf{X}\|,$$

and obtain

$$\frac{\|\Delta \mathbf{X}\|}{\|\mathbf{X}\|} \leq \|\mathbf{J}^{-1}\| \|\mathbf{J}\| \frac{\|\Delta \boldsymbol{\Theta}\|}{\|\boldsymbol{\Theta}\|};$$

The error amplification factor, called the *condition number* κ , is therefore defined as

$$\kappa(\mathbf{J}^{-1}) = \|\mathbf{J}^{-1}\| \|\mathbf{J}\|.$$

The condition number is thus dependent on the choice of the matrix norm. The most used norms are:

- the 2-norm defined as the square root of the largest eigenvalue of matrix $\mathbf{J}^{-T} \mathbf{J}^{-1}$: the condition number of \mathbf{J}^{-1} is thus the square root of the ratio between the largest and the smallest eigenvalues of $\mathbf{J}^{-T} \mathbf{J}^{-1}$,
- the Euclidean (or Frobenius) norm defined for the $m \times n$ matrix \mathbf{A} by:

$$\|\mathbf{A}\| = \sqrt{\sum_{i=1}^{i=m} \sum_{j=1}^{j=n} |a_{ij}|^2}$$
 or equivalently as $\sqrt{\text{tr}(\mathbf{A}^T \mathbf{A})}$: if λ_i denotes the eigenvalues of $\mathbf{J}^{-T} \mathbf{J}^{-1}$, then the condition number is the ratio between $\sum \lambda_i^2$ and $\prod \lambda_i$. Note that sometime is also used a weighted Frobenius norm in which $\mathbf{A}^T \mathbf{A}$ is substituted by $\mathbf{A}^T \mathbf{W} \mathbf{A}$ where \mathbf{W} is the weight matrix

In these two cases, the smallest possible value of the condition number is 1. The inverse of the condition number, which has a value in [0,1], is also often used. A value of 0 will indicate that the inverse jacobian matrix is singular.

The condition number is quite often used as an index to describe first the accuracy/dexterity of a robot and, second, the closeness of a pose to a singularity. For the later point it is in general not possible to define a mathematical distance to a singularity for robots whose d.o.f. is a mix between translation and orientation: hence the use of the condition number is as valid an index than any other one. But it has the advantage of being a single number for describing the overall kinematic behavior of a robot.

The definition of the condition number makes clear that we cannot calculate its analytical form as a function of the pose parameters except for very simple robot. But robust linear algebra software allows to calculate it numerically for a given pose.

But for robot having both translation and orientation d.o.f. there is a major drawback of the condition number: the matrix involved in its calculation are not homogeneous in terms of units. Hence the value of the condition number for a given robot and pose will change according to the unit choice, while clearly the kinematic accuracy is constant. Ma and Angeles [6] suggested to define a *normalized inverse jacobian matrix* by dividing the rotational elements of the matrix by a length such as the length of the links in a nominal position, or the *natural length* defined as that which minimizes the condition number for a given pose. Still the choice of the length remains arbitrary as it just allows to define a correspondence between a rotation and a translation and as mentioned by Park [9] "this arbitrariness is an unavoidable consequence of the geometry of SE(3)".

To evaluate the efficiency of the condition number for accuracy evaluation we just use our Gough robot and chooses three reference poses defined by the coordinates of the center and the Euler angles as $P_1=x=y=0, z=53$ cm, $\psi=0, \theta=0, \phi=0$ (roughly the pose obtained for the mid-stroke of the actuator), $P_2=x=y=0, z=53$ cm, $\psi=30^\circ, \theta=0, \phi=0$ (whose orientation is roughly 1/3 of the possible rotation around the z axis) and $P_3=x=y=10, z=53$ cm, $\psi=0, \theta=0, \phi=0$. (close to the border of the translation workspace for this orientation). We then computed the absolute value of the maximal positioning error at these poses, obtained as the sum of the absolute value of the elements of the rows of the kinematic jacobian, as indicated in the following table.

Pose	ΔX_x	ΔX_y	ΔX_z	ΔX_{θ_x}	ΔX_{θ_y}	ΔX_{θ_z}
P_1	0.1184	0.1268	0.010087	0.1185	0.1184	0.697
P_2	0.1189	0.1274	0.01266	0.1333	0.1429	0.808
P_3	0.123	0.1309	0.0372	0.15	0.1663	0.7208

It can be seen in this table that the positioning errors are significantly larger for P_2 and P_3 compared to P_1 . As for P_3 the errors are usually larger compared to P_2 except for the rotation around z . Hence as far as accuracy is concerned the ordering of the poses from the most to the least accurate is P_1, P_2, P_3 and we expect to obtain a similar ordering for the condition number.

For this robot we define the normalized inverse jacobian matrix J_n^{-1} obtained by dividing the orientation components of the J_k^{-1} by 53 i.e. roughly the legs lengths at pose P_1 . The considered accuracy indices will be

- C_d : the determinant of J_k^{-1}
- C_2, C_2^n : the 2-norm condition number of J_k^{-1}, J_n^{-1}
- C_F, C_F^n : the Frobenius-norm condition number of J_k^{-1}, J_n^{-1}
- C_2^3, C_F^3 : the 2-norm and Frobenius norm condition number of the inverse jacobian matrix obtained when the inverse kinematics equations are based on the coordinates of 3 points of the end-effector. The chosen points will be all possible triplets in the set B_i : hence we will provide ranges for these indices.

The results are presented in the following table:

	C_d	C_2	C_2^n	C_F	C_F^n	C_2^3	C_F^3
P_1	-29.22	75.14	63.9	152.8	70.2	[9.55,55.47]	[258.8,3204.9]
P_2	-24.64	75.16	73.8	154	80.9	[9.62,43.84]	[218.8,2383.6]
P_3	-23.93	80.65	68.4	158.3	74.7	[10.06,58.95]	[286.5,3618]

For C_2 it may be seen that the difference is surprisingly very small between P_1, P_2 and significant between P_3, P_2 . The ordering between P_2, P_3 is not respected for C_2^n, C_F^n although these indices are coherent when considering P_1 . For C_F, C_d the ordering is respected although the changes in the index are relatively small for C_F . On the other hand there is a surprisingly decrease of C_2^3, C_F^3 between P_2 and P_1 while there is a significant increase between P_1 and P_3 . Hence none of this condition numbers exhibits a completely coherent behavior with respect to the accuracy of this robot.

This simple example shows clearly that the concept of condition number has to be carefully considered when talking about optimal design for robot.

5 Isotropy

An *isotropic* pose of a robot is defined as a pose where κ is equal to 1 and a robot which has only isotropic poses in its workspace is coined an *isotropic robot*. Designing an isotropic parallel robot is often considered as a design objective [3, 12]. A trivial example of isotropic robot is a serial Cartesian X-Y-Z robot whose kinematic jacobian matrix is the identity. But this is a surprising denomination as stricto sensu isotropy indicates that the performances of a robot should be the same whatever is the motion direction. Now if we assume that all the actuator velocities of a X-Y-Z robot are bounded to 1, then the maximal velocity of the end-effector lie in the range $[1, \sqrt{3}]$: as far as velocity is considered such robot is far from isotropy. Still the concept may have some interest: for example any Cartesian robot whose actuator axis are not mutually orthogonal will exhibit a ratio between its maximal velocities over its workspace that will be larger than $\sqrt{3}$. Hence, instead of using the name "isotropic robot" we may consider using the name "maximally regular robot".

6 Global Conditioning Index

The condition number is a local indication for the dexterity of a robot. To evaluate the dexterity of a robot over a given workspace W Gosselin [4] has introduced the *global conditioning index* (GCI) as:

$$\text{GCI} = \frac{\int_W \left(\frac{1}{\kappa} \right) dW}{\int_W dW} .$$

which correspond to the average value of $1/\kappa$. Clearly this concept makes sense for the optimal design of robot for which the extremal and average value of any performance are important design factors. But apart of the validity of the condition number that has been discussed in a previous section the problem with the GCI is its calculation. Clearly we cannot expect to obtain its closed-form and we must rely on a numerical evaluation. The usual method is to sample the workspace using a regular grid, compute $1/\kappa_i$ at each node N_i and approximate the GCI as GCI_a , the sum of the $1/\kappa_i$ divided by the number of nodes. This calculation may be computer intensive as its complexity is exponential with respect to the number of d.o.f. of the robot. Furthermore this method does not allow to get a bound on $|\text{GCI} - \text{GCI}_a|$. To deal with this error problem it is sometimes assumed that if the result with m_1 sampling points is close to the result obtained with m_2 points, m_2 being significantly larger than m_1 , then the later result is a good approximation of the index. This assumption will be true only if the condition number is smooth enough, a claim that is difficult to support. Consider for example a simple planar serial 2R robot: its GCI can be computed almost exactly as it depends only on a single parameter. We sample this parameter using $10, 20, \dots, m_1$, $m_2 = m_1 + 10$ points and stop the calculation when the relative error between $\text{GCI}_a(m_1), \text{GCI}_a(m_2)$ is lower than 0.5% and assumes $\text{GCI} \approx \text{GCI}_a(m_2)$. For $m_1 = 50$ the relative error is 0.377% while the relative error on the GCI is still 1.751%. It may be assumed that such error will even be larger for more complex robot.

A better evaluation will probably be obtained by using Monte-Carlo integration (with an error that decreases as $1/\sqrt{n}$ where n is the number of sampling nodes) or quasi-Monte Carlo. In the previous example (which is not favorable for Monte-Carlo method as there is only one parameter) we found out that by using the same stop criteria the relative error on the GCI was reduced to 0.63%. A certified evaluation of the global conditioning index is therefore an open problem but nevertheless the calculation of such index will probably be computer intensive.

7 Conclusion

Classical dexterity indices such as the condition number are not very adequate for parallel robots. In our opinion the most appropriate accuracy indices are

the determination of the maximal positioning errors, their average values and their variance. We have presented in a recent paper a computer intensive method for finding the largest maximal positioning errors, up to an arbitrary accuracy, of a 6 d.o.f. robot [7]. A real challenge is to design algorithms for calculating the average and variance of the maximal positioning errors over a given workspace. An important point is that there is no need to calculate these values *exactly* as soon as it is possible to impose a bound on the calculation error. Indeed for comparison purposes an approximate value with a guaranteed error will be sufficient.

References

1. Bonev I.A. and Zlatanov D. The mystery of the singular SNU translational parallel robot.
www.parallelmechanic.org/Reviews/Review004.html, June, 12, 2001.
2. Di Gregorio R. and Parenti-Castelli V. Mobility analysis of the 3-UPU parallel mechanism assembled for a pure translational motion. *ASME J. of Mechanical Design*, 124(2):259–264, June 2002.
3. Fattah A. and Hasan Ghasemi A.M. Isotropic design of spatial parallel manipulators. *Int. J. of Robotics Research*, 21(9):811–824, September 2002.
4. Gosselin C. *Kinematic analysis optimization and programming of parallel robotic manipulators*. Ph.D. Thesis, McGill University, Montréal, June, 15, 1988.
5. Han C. and others . Kinematic sensitivity analysis of the 3-UPU parallel manipulator. *Mechanism and Machine Theory*, 37(8):787–798, August 2002.
6. Ma O. and Angeles J. Optimum architecture design of platform manipulator. In *ICAR*, pages 1131–1135, Pise, June, 19-22, 1991.
7. Merlet J-P. and Daney D. Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace. In *IEEE Int. Conf. on Robotics and Automation*, Barcelona, April, 19-22, 2005.
8. Parenti-Castelli V. and Di Gregorio R. Influence of manufacturing errors on the kinematic performance of the 3-UPU parallel mechanism. In *2nd Chemnitzer Parallelkinematik Seminar*, pages 85–99, Chemnitz, April, 12-13, 2000.
9. Park M.K. and Kim J.W. Kinematic manipulability of closed chains. In *ARK*, pages 99–108, Portoroz-Bernardin, June, 22-26, 1996.
10. Tsai L-W. Kinematics of a three-dof platform with three extensible limbs. In *ARK*, pages 401–410, Portoroz-Bernardin, June, 22-26, 1996.
11. Wolf A., Shoham M., and Park F.C. Investigation of singularities and self-motions of the 3-UPU robot. In *ARK*, pages 165–174, Caldes de Malavalla, June 29- July 2, 2002.
12. Zanganeh K.E. and Angeles J. On the isotropic design of general six-degree-of-freedom parallel manipulators. In J-P. Merlet B. Ravani, editor, *Computational Kinematics*, pages 213–220. Kluwer, 1995.

Session Overview

Simultaneous Localisation and Mapping

Paul Newman¹ and Henrik I. Christensen²

¹ University of Oxford pnewman@robots.ox.ac.uk

² Royal Institute of Technology, Sweden hic@nada.kth.se

1 Introduction

The Simultaneous Localisation and Mapping (SLAM) problem remains a prominent area of research in the mobile robotics community. The ISRR symposia have borne witness to marked progress of the field since its conception almost 20 years ago. This year, once again, the question "is the SLAM problem now solved?" was posed. Well the answer to that question probably lies in the definition of "solved". We still do not have the persistent SLAM-enabled machines that we strive for, so in that sense, perhaps it isn't solved, but we do have a firm understanding of the problem now. We do appreciate the limits of performance, we can handle uncertainties in a principled way and recognize the penalties for failing to do so. We also have several solutions to the scaling problem that so dogged the field for several years. To these probabilistic frameworks we are able to attach any of several representational schemes to represent both maps and vehicle trajectories. We run these "solutions" on vehicles equipped with various sensors, cameras, radars, sonars and of course the ubiquitous laser range finder.

One crucial missing component is that of operational robustness. Broadly, the issue can be split into two categories: firstly robustness in the face of erroneous manipulation and insufficient representations of the underlying pdfs and secondly robustness in presence of perceptual ambiguity. The later problem is receiving substantial attention under the guise of the "data association" and "loop closing" problems within the SLAM context. Failing to obtain persistent, long-term SLAM deployments because of accumulating errors in pdf representations is, of course, a closely related problem (bad data association can be caused by incorrect probabilistic representations). A common, although not blanket, criticism of contemporary SLAM techniques is their lack of introspection, they tend to be passive both in data acquisition and data processing. There seems to be a significant scope for planning, acting, and perceiving to aid the SLAM estimation process itself and be more pro-active in assessing the quality of the estimation results.

Perhaps the greatest challenges to contemporary SLAM techniques become clear when trying to apply them in the great outdoors. The benign, distinct surfaces of the flat indoor domain are no more, the world is now truly 3D and single-plane laser scanners are inadequate. The local scene is frequently orders of magnitude larger and may need multiple sensor modalities to access it - cameras, radar, 3D laser and in the underwater domain, beam-steerable sonars. Then there is the issue of performing SLAM in highly dynamic environments that outdoor settings typically demand. The overwhelming majority of SLAM research has relied upon the static world assumption - with varying but typically small degrees of tolerance to scene dynamics. This begs the question how should a principled SLAM system cope with substantial and unexpected scene changes - how can it differentiate this from a catastrophic estimation failure?

2 Summary of Papers Presented at ISRR

The paper by Bowling et al. addresses the problem of localisation without an *a-priori* choice of representation or specification of process and observation models. The paper hinges on the concept of Action Respecting Embedding a technique similar to Local Linear Embedding, that learns a low dimension manifold within a high dimensional measurement input space. Crucially this operation preserves the local topology originally present when the measurement sequence was gathered. While not addressing the SLAM problem in a familiar way, the paper does illustrate the opportunities that techniques being established in the machine learning domain offer the SLAM research community.

The paper by Wang et al. is a presentation of decoupling in SLAM. Traditionally there is a correlation between robot motion and sensory readings which results in a correlation of all data in a SLAM model. The correlation results in an overall complexity of SLAM which is $O(N^2)$, where N is the number of map features. Various approaches to address the scalability problem have presented in the literature, including the C-EKF by Nebot et al [3], FastSLAM by Montemerlo [2] and the Atlas framework by Bosse et al [4]. In this paper it is demonstrated how a careful relative formulation of the problem, combined with the information filter framework allows decoupling of mapping and localisation — providing a SLAM algorithm with good scaling properties that still allows each feature estimate to be improved with each observation.

Another approach which addresses the scaling problem is presented by Walter et al. The paper again uses the information formulation of the SLAM problem and, like the SEIF proposal [1] manages the scaling problem by maintaining an active set of features with substantial correlations to the vehicle. The suggestion here is to use the act of deleting and re-initialising the vehicle states to create and manage this active subset of features in a consistent

fashion. The paper analyses the new proposal (ESEIF) and compares it to the SEIF formulation concluding with a side by side comparison of the two algorithms working on two well known data sets.

3 Wrap-Up

So it seems that while it is indisputable that the state of the art SLAM has moved on substantially over the past decade there is still interest research going on, much to do and many interesting questions left un-answered. It is not a solved problem but we do know what questions we should be asking.

References

1. S. Thrun, and Y. Liu. Multi-robot SLAM with sparse extended information filters. "Proceedings of the 11th International Symposium of Robotics Research", Dario and Chatila (Eds.), Springer Verlag, Sienna, October 2003.
2. M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit. FastSLAM A Factored Solution to the Simultaneous Localization and Mapping Problem, "Proceedings of the AAAI National Conference on Artificial Intelligence", Edmonton, Canada, 2002.
3. J. Guivant and E. Nebot. Optimization of the Simultaneous Localization and Map Building Algorithm for Real Time Implementation. *IEEE Transactions on Robotic and Automation*, vol. 17, No. 3, pp. 242-257, June 2001.
4. M. Bosse and P. Newman and J.J. Leonard and S. Teller. SLAM in Large-scale Cyclic Environments using the Atlas Framework. *International Journal of Robotics Research*. vol. 23, pp. 1113-1139, 2004

Subjective Localization with Action Respecting Embedding

Michael Bowling¹, Dana Wilkinson², Ali Ghodsi³, and Adam Milstein²

¹ Department of Computing Science, University of Alberta
bowling@cs.ualberta.ca

² School of Computer Science, University of Waterloo
{d3wilkin,ahpmilst}@uwaterloo.ca

³ Department of Statistics and Actuarial Science, University of Waterloo
aghodsib@uwaterloo.ca

Summary. Robot localization is the problem of how to estimate a robot’s pose within an objective frame of reference. Traditional localization requires knowledge of two key conditional probabilities: the motion and sensor models. These models depend critically on the specific robot as well as its environment. Building these models can be time-consuming, manually intensive, and can require expert intuitions. However, the models are necessary for the robot to relate its own subjective view of sensors and motors to the robot’s objective pose. In this paper we seek to remove the need for human provided models. We introduce a technique for *subjective localization*, relaxing the requirement that the robot localize within a global frame of reference. Using an algorithm for action-respecting non-linear dimensionality reduction, we learn a subjective representation of pose from a stream of actions and sensations. We then extract from the data natural motion and sensor models defined for this new representation. Monte Carlo localization is used to track this representation of the robot’s pose while executing new actions and receiving new sensor readings. We evaluate the technique in a synthetic image manipulation domain and with a mobile robot using vision and laser sensors.

1 Introduction

A key problem in mobile robotics is localization: estimating a robot’s pose while it moves and senses in the world. Knowledge of a robot’s position in its environment is one of the most basic requirements for many autonomous tasks. The majority of localization techniques focus on *objective localization*, where the pose is estimated in terms of a human defined global frame of reference. For example, pose may be defined as the position and orientation on a two-dimensional Cartesian map with units in meters. In this paper, we seek to relax this notion of localization.

One of the most successful approaches to objective localization uses probabilities to model all aspects of a robot’s uncertainty, including the current pose estimate, the effect of actions, and the information provided by sensors. Rules of probabilistic inference can then be applied in a straightforward fashion to maintain an estimate of the robot’s location. Approaches of this type often restrict the form of the models

(e.g., Gaussian distributions in Kalman filters [Kal60]) or use various approximation techniques (e.g., sampling in Monte Carlo localization [FBDT99]), to allow inference, and thus localization, to be computationally feasible.

A key prerequisite for all probabilistic approaches are models of the uncertainty in the robot's motion and sensors. Classical kinematics defines the expected global motion of the robot when a particular control is applied to it. But kinematics requires many assumptions in its deterministic calculations (e.g., infinite friction) that do not hold in practice. Hence, robot motion is uncertain. Likewise, there are many uncontrollable and unpredictable factors (e.g., acoustic reflectance of a surface with sonar, or ambient lighting with vision) that effect readings from sensors. Hence, robot sensing is also uncertain. Probabilistic models of these uncertainties form the basis for inference (which drives the localization). Unfortunately, these models are often not easy to build. They can require extensive knowledge of the robot's kinematics or sensors, which may not be known or easily described. They may require time-consuming manual measurements to estimate characteristics of noise or to build a map of sensor readings over the environment. Finally, by definition, a well constructed model must be specific to the particular hardware used. Modifying the robot platform invalidates these laboriously constructed models and new models must be created. For example, changing from a wheeled robot to a legged robot obviously invalidates the motion model. Changing from a sonar to a laser, or from a laser to a camera will require replacement of the sensor model. Even minor changes, such as inflating the tires on the robot, or replacing its camera with one of a different model, will require expert modifications to the various models. Recent work has examined techniques for automatically calibrating some of these models (e.g., [RT99], [MTKW02], [EP04], [SS05]), but no current method exists to calibrate these models for objective localization without considerable expert knowledge.⁴

This paper examines the problem of *subjective localization*. We relax the requirement that the robot must estimate its pose in terms of a global frame of reference. Instead, the choice of representation is left as part of the localization problem. This relaxation allows the robot to learn both motion and sensor models as the models can be defined purely in terms of its own subjective motor and sensor values. Although objective localization may be necessary for certain tasks, not all tasks require knowledge of an objective position. Delivery tasks, for example, need only recognize location with respect to locations visited in the past. A robot can be given a guided tour of its environment ("getting its bearings") and informed of salient locations along the tour which can then be labeled in its subjective map.

The problem of subjective localization will be tackled with a four-step process. We first gather data of the robot moving and sensing in its world. We then use this data to learn both an appropriate frame of reference for localization as well as the actual trajectory the robot followed during the data gathering. We then learn motion and

⁴ Special mention should be made of the work of Stronger and Stone [SS05], which learns motion and sensor models starting with only an inaccurate motion model. Their approach is still quite knowledge intensive, using a human-defined preprocessing step to simplify the complex image sensor down to a single estimate of distance to a beacon.

sensor models in this frame of reference from the training data and the learned trajectory. Finally, we incorporate these models into Monte Carlo Localization (MCL), a probabilistic localization technique. The cornerstone of this approach is extracting a subjective frame of reference from a trace of sensorimotor data. This is solved with Action Respecting Embedding (ARE) [BGW05], a technique for non-linear dimensionality reduction which finds low-dimensional descriptions of the robot's pose in a frame where actions correspond to simple transformations.

The rest of this paper is organized as follows. Section 2 provides an overview of Monte Carlo localization. Section 3 summarizes the Action Respecting Embedding algorithm, which extracts the subjective representation. Section 4 describes the learning of motion and sensor models in this new frame of reference, which can then be used in MCL. Section 5 demonstrates the effectiveness of this approach, both in a synthetic image manipulation domain and with a mobile robot using first a camera and then a laser as the primary sensor. Section 6 concludes.

2 Monte Carlo Localization

Monte Carlo Localization (MCL) [FBDT99] is a method for estimating the posterior distribution of the robot's pose conditioned on the robot's actions and sensor readings. It relies on the Markovian assumption that the past and the future are conditionally independent given the present. MCL is an implementation of a recursive Bayes filter. If x_t is the location at time t , z_t is the sensor data at time t , and u_t is the motion data at time t then the posterior distribution becomes:

$$\text{Bel}(x_t) = p(x_t|z_T, u_T) \quad (1)$$

where $z_T = z_1, \dots, z_t$ and, similarly, $u_T = u_1, \dots, u_{t-1}$. For objective localization the sensor data is usually in the form of range data, such as laser range-finder readings, however any type of sensor for which the proper kind of model exists is admissible. The motion data is usually the report from the robot's odometers, but again, any data with an appropriate model will satisfy the equation.

For a recursive Bayes' filter, a recursive formula is necessary, so Equation 1 is converted, using a combination of Bayes' rule and the Markovian assumption, into:

$$\text{Bel}(x_t) = (1/Z) p(z_t|x_t) \int p(x_t|x_{t-1}, u_t) \text{Bel}(x_{t-1}) dx_{t-1}, \quad (2)$$

where Z is a normalization term. $p(x_t|u_t, x_{t-1})$ is called the motion model, the probability of a resulting pose given a starting pose and an action. $p(z_t|x_t)$ is called the sensor model, the probability of receiving a particular sensor reading given the robot's pose. If these two models exist then MCL can be performed.

Unfortunately, virtually all robots operate in a continuous space, so the integral in Equation 2 is impossible to compute directly. In order to solve the problem, MCL approximates the continuous space with a finite set of samples or "particles". At each time-step the set of samples is moved probabilistically according to the motion model. The samples are then annotated with a weight determined by the sensor model. The weight of each sample is the probability of receiving the observed sensor reading given that the robot is at the location represented by the particle. Finally, the

particles are resampled according to their weight. Resampling generates the new set of samples by choosing a particle with probability proportional to its weight with replacement. Although MCL is obviously only correct as the number of samples approaches infinity, it is often accurate for a relatively small number of samples.

MCL is a common technique for objective localization, where the motion model and sensor model are constructed by hand or through experimentation. It can be used equally well for subjective localization if one has appropriate motion and sensor models in a subjective frame of reference. Section 3 deals with finding such a frame of reference, while Section 4 details the learning of the required models.

3 Action Respecting Embedding

High-dimensional data sets, such as a sequence of images or scans from a laser range-finder, can usually be characterized by a low-dimensional representation that is related to the process generating the data. For example, one low-dimensional representation for image data might correspond to the degrees of freedom of the platform moving the camera which gathered the data. Such a low-dimensional representation of the sensor readings might be an ideal frame of reference for subjective localization. The goal, then, is to take a temporal sequence of sensor readings z_1, \dots, z_n with associated control actions, u_2, \dots, u_n , and find a low-dimensional representation for z_1, \dots, z_n that would be appropriate for localization.

Recently, *non-linear manifold-learning* techniques have been used to map high-dimensional datasets, such as sensor readings, into smaller dimensional spaces. Semidefinite Embedding (SDE) [WS04] is one such technique. SDE learns a kernel matrix, which represents a non-linear projection of the input data into a more linear representation. It then uses Kernel PCA [SS02], a generalization of principle components analysis to feature spaces represented by kernels, to extract out a low-dimensional representation of the data. The kernel matrix, K , is learned in SDE by solving a semidefinite program with a simple set of constraints. The most important constraints encode the common requirement in dimensionality reduction that the non-linear embedding should preserve local distances. In other words, nearby points in the original input space should remain nearby in the resulting feature representation. Therefore, SDE requires a distance metric $\|\cdot\|$ on the original input space, and uses this metric to construct a k -nearest neighbors graph. It then adds constraints into the semidefinite program to ensure that the distance between neighbors is preserved. The optimization maximizes the trace of K , i.e., the variance of the learned feature representation, which should minimize its dimensionality.

SDE, though, ignores two important pieces of knowledge about our data: the temporal ordering of the input vectors z_i , and the action labels u_i . Therefore, SDE doesn't require temporally nearby input points to be spatially nearby in the feature representation. Also, SDE won't enforce the extracted space to be one where the robot's actions have a simple interpretation. The recent Action Respecting Embedding (ARE) algorithm uses the aforementioned knowledge to address these issues.

Formally, ARE takes a set of D -dimensional input vectors, z_1, \dots, z_n (i.e., sensor readings, in temporal order) along with associated discrete actions u_1, \dots, u_{n-1} (where action u_i was executed between input z_i and input z_{i+1}), and computes a

set of d -dimensional output vectors x_1, \dots, x_n in one-to-one correspondence with the input vectors that provide a meaningful embedding in $d < D$ dimensions. ARE is similar to SDE but extends it in two key ways. First, it exploits the knowledge that the sensor readings are given in a temporal sequence by building an improved neighborhood graph based on each input's distances to its temporal neighbors using an arbitrary local distance metric⁵. Second, it constrains the embedding to respect the action labels that are associated with adjacent pairs of observations. This ensures that the actions have a simple interpretation in the resulting feature space.

This second enhancement of ARE is the critical feature for subjective localization. ARE constrains the learned manifold to be a space where the actions correspond to transformations consisting only of rotation and translation in that space⁶—in other words, every action is required to be a distance-preserving transformation for all inputs in the learned feature space. Letting $\Phi(z_i)$ denote input z_i 's representation in this learned feature space, we require u 's transformation, f_u , to satisfy:

$$\forall i, j \quad \|f_u(\Phi(z_i)) - f_u(\Phi(z_j))\| = \|\Phi(z_i) - \Phi(z_j)\|. \quad (3)$$

Now, let $u = u_i = u_j$, so $f_u(\Phi(z_i)) = \Phi(z_{i+1})$ and $f_u(\Phi(z_j)) = \Phi(z_{j+1})$. Hence, constraint 3 becomes:

$$\|\Phi(z_{i+1}) - \Phi(z_{j+1})\| = \|\Phi(z_i) - \Phi(z_j)\|. \quad (4)$$

In terms of the kernel matrix, this can be written as:

$$\begin{aligned} \forall i, j \quad u_i = u_j \Rightarrow K_{(i+1)(i+1)} - 2K_{(i+1)(j+1)} + K_{(j+1)(j+1)} \\ = K_{ii} - 2K_{ij} + K_{jj}. \end{aligned} \quad (5)$$

Add constraint 5 to the SDE optimization problem to get the ARE algorithm shown in Table 1.

Table 1. Algorithm: Action Respecting Embedding (ARE).

Algorithm: ARE($\ \cdot\ $, (z_1, \dots, z_n) , (u_2, \dots, u_n))
Construct neighbor graph, N, according to [BGW05].
Maximize $\text{Tr}(K)$ subject to $K \succeq 0$, $\sum_{ij} K_{ij} = 0$, $\forall ij \quad N_{ij} > 0 \vee [N^T N]_{ij} > 0 \Rightarrow$ $K_{ii} - 2K_{ij} + K_{jj} \leq \ z_i - z_j\ ^2$, and $\forall ij \quad u_i = u_j \Rightarrow K_{(i+1)(i+1)} - 2K_{(i+1)(j+1)} + K_{(j+1)(j+1)}$ $= K_{ii} - 2K_{ij} + K_{jj}$
Run Kernel PCA with learned kernel, K.

⁵ We have found that ARE is fairly robust to the choice of distance metrics, and use simple Euclidean distance for all of the experiments in this paper.

⁶ Notice this is not requiring the actions in the objective space to be rotations and translations, since ARE is learning a non-linear feature representation.

4 Subjective Localization

Recall that the subjective localization problem involves both determining an appropriate subjective frame of reference for localization and then tracking the robot's position in that representation. In the work presented in this paper, ARE is used to learn the frame of reference. A completely unsupervised stream of data from a robot acting in the world (consisting of a stream of sensor readings, z_1, \dots, z_n , and associated actions, u_1, \dots, u_{n-1} , which are elements of some set of discrete actions) will be used as input to ARE in order to learn an appropriate subjective representation.

In order to perform localization with this representation, a motion model and sensor model must be computed. ARE, though, provides more than just a coordinate system. It also provides the actual d -dimensional embedded points, x_1, \dots, x_n , that correspond to the trajectory the robot followed in the data-gathering phase. This trajectory—along with the robot's sensations, z_1, \dots, z_n , and actions, u_1, \dots, u_n —can be used to learn the models from the training data. Both models will be learned in a similar fashion. We will first estimate the expectation of the model and then use the error to estimate a noise component. We begin with the motion model.

4.1 Motion Model

The motion model is the posterior distribution $p(x_t|u_t, x_{t-1})$. Since this model will be used in a particle filter, it is only necessary to be able to draw a sample, \hat{x} , from the model, given a u_t and x_{t-1} , i.e.:

$$\hat{x} \sim p(x_t|u_t, x_{t-1}).$$

First, separate the model into an expectation plus a noise component.

$$\hat{x} \sim E(x_t|u_t, x_{t-1}) + \eta(x_t|u_t, x_{t-1})$$

Now make the simplifying assumption that the noise depends only on the action and not on the previous pose. This gives the form:

$$\hat{x} \sim E(x_t|u_t, x_{t-1}) + \eta(x_t|u_t) \tag{6}$$

We can now learn the model by learning the expectation component, then using the sample errors to estimate the noise component.

Consider some action u . Every t where $u_t = u$ gives one sample, x_t and x_{t-1} , from the distribution $p(x_t|u, x_{t-1})$. Using these sample points, a function of x_{t-1} is desired that gives a close estimate of x_t . ARE explicitly includes constraints that ensure such a function exists and is a simple rotation plus a translation in the learned representation. We can recover these functions by solving an optimization problem to find the corresponding rotation matrix A_u and translation vector b_u such that $f_u(x) = A_u x + b_u$. Formally,

$$\text{Min } \sum_{t:u_t=u} \|A_u x_{t-1} + b_u - x_t\|^2 \text{ s.t. } A_u^T A_u = I \text{ (i.e., } A \text{ is a rotation)}$$

This problem is similar to the extended orthonormal Procrustes problem [SC70] and has a closed form solution. Let X_u be a matrix whose columns are x_{t-1} for all t such that $u_t = u$, and let Y_u be a matrix whose columns are x_t for the same t . The following is the solution to this optimization problem:

$$A_u = VW^T \quad \text{where} \quad VSW^T = \text{svd} \left(Y_u^T \left(I - \frac{ee^T}{d} \right) X_u \right) \quad (7)$$

$$b_u = (Y_u - A_u X_u)^T e / d, \quad (8)$$

where $\text{svd}(\cdot)$ is the singular value decomposition and e is a column vector with d ones. Now the expected motion can be defined as:

$$E(x_t | u_t, x_{t-1}) = A_{u_t} x_{t-1} + b_{u_t}. \quad (9)$$

Since we only included the top d principal components of the output of ARE, this model of the expected motion won't be exact. The errors in the learned transformation can be used to build a model of the motion noise. Consider again some action u , let ξ_t be the residual error for action u on x_t :

$$\xi_t = A_u x_{t-1} + b_u - x_t \quad \text{where} \quad \sum_{t:u_t=u} \xi_t = \mathbf{0}.$$

The motion noise can be modeled as a zero-mean multivariate Gaussian, where the covariance matrix can be estimated directly from the samples ξ_t . Formally:

$$\eta(x_t | u_t) \sim N(\mathbf{0}, \Sigma_{u_t}), \quad (10)$$

where:

$$\Sigma_{u_t}(i, j) = \sum_{t:u_t=u} \xi_t(i) \xi_t(j).$$

Combining Equations 6, 9, and 10 gives the complete motion model.

4.2 Sensor Model

The sensor model is the probability distribution $p(z_t | x_t)$. In the context of a particle filter, the density of the distribution at z_t must be provided for a given x_t . In estimating this model from the data a few assumptions must be made. Notice that ARE doesn't take the images directly as its input, but rather uses an image's distance to every other image as a kind of feature representation. We will use the same representation for new observations, computing a feature vector:

$$\bar{z}_t(i) = \|z_t - z_i\| \quad \forall i = 1 \dots n.$$

The best way to view this feature vector is that it provides a crude estimate of the "distance" of the robot's pose to the previous poses, x_1, \dots, x_n . The additional assumption is required that each of the components of the feature vector are independently distributed⁷. That is, each is an independent estimate of the "distance" to a past pose. The final assumption is that this probability only depends upon the *distance* to the specific past pose in the subjective representation⁸, i.e., $\|x_t - x_i\|$. These assumptions combine to give the following form for the model. Let $d_{ti} = \|x_t - x_i\|$:

⁷ This assumption, while almost certainly incorrect, is similar to the common MCL assumption (often necessary for tractability) that sensor readings are independent.

⁸ This is not an unreasonable assumption, since ARE explicitly constrains distances in the subjective representation $\|x_i - x_j\|$ by observed image distances $\|z_i - z_j\|$.

$$p(z_t|x_t) = p(\bar{z}_t|x_t) = \prod_{i=1}^n p(\bar{z}_t(i)|x_t) = \prod_{i=1}^n p(\bar{z}_t(i)|d_{ti}). \quad (11)$$

Now to estimate a Gaussian model for the conditional random variable $\bar{z}_t(i)|d_{ti}$. Consider again the training trajectory, each x_t gives one sample for this joint distribution: $\bar{z}_t(i)$ and d_{ti} . To build a Gaussian model, for each landmark, i , use regression to fit a low-degree polynomial determining the distribution mean as a function of distance ($\mu_i(d_{ti})$)⁹. Then take the mean of the squared errors to estimate distribution variance (σ_i^2). This gives the following Gaussian density function:

$$\bar{z}_t(i)|d_{ti} \sim N(\mu_i(d_{ti}), \sigma_i^2). \quad (12)$$

Combining Equations 11 and 12 gives the sensor model.

4.3 Using the Models

The final step of the technique is to use the motion and sensor models with Monte Carlo localization to track the robot's position in the learned subjective space. The only detail left to be addressed is the initial distribution for localization. Since we processed the data after a single training run, we know our exact position in the subjective representation, x_n . All the samples in MCL are initialized to this point.

In the end, the subjective localization procedure has three configurable parameters: the dimensionality of the subjective representation, d , the degree of polynomial used in the sensor model, and the number of particles used by MCL. Overall, the procedure has a small number of parameters and, as seen in the next section, can actually localize in a number of different situations with a variety of parameter settings.

5 Results

Here, the algorithm from Section 4 is applied to two different domains. The first is IMAGEBOT a synthetic image manipulation domain. The second is a mobile robot, demonstrating localization with first a camera, then a laser range-finder as the primary sensor. First the domains are described followed by the experiments with the results of localization. Then a measure of accuracy is presented that is appropriate for subjective localization, showing accuracy across a variety of experiments. Finally, we show the robustness of the algorithm to the choice of its few parameters.

5.1 The Domains

We explored subjective localization in two different domains.

Image based (IMAGEBOT).

Given an image, imagine a virtual robot that observes a small patch on that image and takes actions to move this patch around the larger image. This “image robot” provides an excellent domain in which subjective localization can be rigorously tested while having obvious corollaries to mobile robotics.

For these experiments, IMAGEBOT will always be viewing a 100 by 100 patch of a 2048 by 1536 image. All the experiments use the image from Figure 1. IMAGEBOT has four translation actions and two zoom actions. The allowed translations are forward, backward, left and right, each by 25 pixels. The zoom changes the scale of

⁹ This is very similar to the sensor model construction by Stronger and Stone [SS05].

the underlying image by a factor of $2^{1/8}$ or $2^{-1/8}$. Since we are interested in noisy actions, zero-mean Gaussian noise is added to the magnitude of the change of any of the actions with a standard deviation of one-tenth of the mean change.



Fig. 1. IMAGEBOT’s world.



Fig. 2. A 45-action IMAGEBOT trajectory.

Mobile Robot.

Experiments were performed on an ActivMedia Pioneer 3 DX8 robot equipped with an ordinary web camera and a laser range-finder. A series of predefined actions were used to move the robot up and down a corridor with data being collected after each action. Additionally, after each action was performed the robot’s position was manually measured to discover actual error. We performed experiments using the camera as the only sensor, then using the laser as the only sensor.

5.2 Experiments

In all experiments, a dataset is gathered by executing a sequence of actions and receiving the associated sequence of sensor readings. After each action, measurement of objective location is taken—used later to compute a measure of accuracy. The sequence is split into two sets, training and test. The training set is used by ARE to extract a subjective representation and associated trajectory. Motion and sensor models are learned as described in Section 4. Finally, the models are used in MCL to localize given the test set. The mean of the particles after every given action and observation is used as the estimated position in the subjective frame of reference.

In order to extract a model of noise, the training data needs to contain examples of executing the same action from approximately the same location. Since the points after taking this action will be in various locations, the noise of the motion model can then be reconstructed. Therefore, each dataset begins by taking repeated short sequences of actions (such as going forward three steps then backward three steps), ensuring the training data includes a representation of noise in the robot’s actions.

Image Based (IMAGEBOT).

In the IMAGEBOT domain three different paths were examined, each path was generated three times, each different due to noise. The first path was a simple line, where IMAGEBOT executed forward and backward actions. The second was an “A” shaped path using forward, backward, left, and right (an example of this trajectory is shown

Figure 2). The last was the same “A” shaped path where the right and left actions were replaced by zoom-in and zoom-out actions, respectively. In all cases, the test data involved retracing IMAGEBOT’s steps back over its path. This involves different observations, though, as the actions are noisy.

Figure 3 shows an example “A” shaped path (the “A” is tilted to the right) in objective coordinates. The dotted line shows the training data with the trajectory starting in the upper left. The solid line shows the test data, a reversed “A” starting from the bottom left. Note that noise prevents the two paths from exactly lining up.

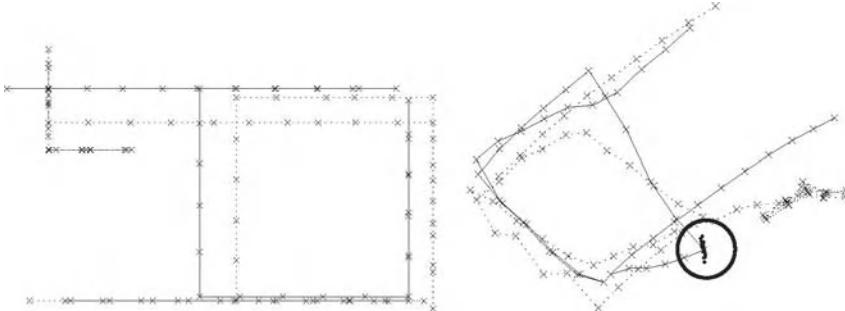


Fig. 3. IMAGEBOT’s “A” shaped trajectory in objective coordinates. The dotted line is the trajectory starting in the upper left. The solid line is the test data, a reversed “A” starting from the bottom left.

Fig. 4. Subjective localization on the “A” on training data, solid line is predicted location using MCL on test data.

Figure 4 shows the results of using the data from Figure 3 with our subjective localization technique. The dotted line shows the trajectory that resulted from running ARE on the training data in the learned frame of reference. The solid line is the predicted points from MCL while receiving images and actions from the test data. The circled cloud of points point shows the set of 100 particles in MCL at that point in the trajectory. The learned trajectory corresponds strongly with the objective trajectory, and the localized trajectory follows along appropriately. In the next section we investigate a quantitative measure of localization accuracy showing the results on this and the other trajectories.

Mobile Robot.

There was one simple path studied with the Pioneer robot, but two experiments were performed with it. In the first, observations were 160x120 pixel images from the camera. In the second, observations were the 180 distance estimates from the laser range-finder. Training and test paths were the same as the first path of IMAGEBOT: a simple forward and backward trajectory. Figure 5 shows the consecutive images taken as the robot traversed this path. The top row (left to right) shows images as the robot moves forward down its path. The bottom row (right to left) shows the continuation of the trajectory as the robot moves back up the path.

Figure 6 shows ARE’s learned trajectory (dotted line) and the predicted trajectory from localization on the test set (solid line) using the camera as sensor input. The objective space corresponds to a single primary dimension, and the trajectories

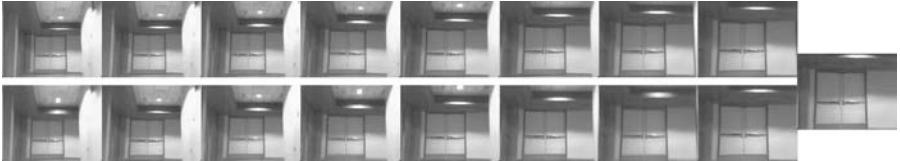


Fig. 5. Images gathered from the robot moving forward (top) then backward (bottom).

correctly capture this as the x dimension in the plot. The learned and predicted trajectories look qualitatively similar when the laser is used as sensor input and so are not shown here.



Fig. 6. Subjective localization of the robot using the camera. Dotted line is ARE's trajectory on training data, solid line is predicted location using MCL on test data.

5.3 Accuracy

Accuracy is a measure for evaluating localization performance. In objective localization, this amounts to comparing the predicted position to hand-measured objective positions and reporting the mean error. For subjective localization, this is not possible as the robot's location is only known in a subjective frame of reference. This makes it difficult to measure the accuracy of an algorithm. As mentioned in the introduction, one use for a subjective representation is for recognizing locations visited in the past. In particular, a new position in the subjective frame of reference can be compared to previous training points. Distance in the subjective space can be used to estimate which training point we expect to be closest to in objective space.

A method for evaluating subjective localization now becomes clear. For a given predicted subjective location, find the closest point in the training data to this location and consider this an objective prediction. The error is simply the distance in objective coordinates between the robot's true (measured) location and this prediction from the training set. This gives a measure of accuracy in objective terms (Note, it is generally impossible to achieve zero error). For comparison, an oracle score can also be computed. Look at the actual objective positions of each point in the test data and determine the closest training point, using this distance as the oracle error. Any measure of accuracy can be compared to this oracle's accuracy. As another baseline, compute the error of a random subjective localization algorithm that chooses a random training point as the prediction of its location. These two baselines, oracle and random, can be used to evaluate the accuracy of any subjective localization method.

Table 2 shows the accuracy results for all three paths in IMAGEBOT averaged over three datasets each with ten complete runs of MCL. Table 3 shows the accuracy

results for both camera and laser-based robot experiments, averaged over ten complete runs of MCL. In IMAGEBOT, the accuracies are within approximately 10 pixels, a vast improvement over the random baseline and not far from the oracle's accuracy. With the mobile robot, the accuracy with the camera is approximately 150mm, an order of magnitude improvement over random and about an order of magnitude behind the oracle performance. The performance with the laser range-finder is not quite as strong, but still demonstrates effective localization.

Table 2. IMAGEBOT accuracy.

	Mean Error (Pixels)		
	Oracle	ARE	Random
Straight line	4.82	10.39	86.83
"A" with translation	3.62	14.81	104.56
"A" with zoom	1.71	19.58	84.67

Table 3. Mobile robot accuracy.

	Mean Error (mm)		
	Oracle	ARE	Random
Robot with camera	14.25	149.10	1482.83
Robot with laser	16.25	287.93	1450.50

5.4 Robustness

Finally, we consider the robustness of this technique. The results in the previous section demonstrate one aspect of robustness—the ability to subjectively localize in two very different domains. Even more compelling, the primary sensor was switched from camera to laser and the robot was still able to successfully localize. The algorithm, entirely unchanged, found a new subjective representation, and new motion and sensor models without requiring time-consuming manual creation of these new models.

Parameters.

Another aspect is the robustness of the algorithm to the setting of its various parameters. There was no tuning of the parameters for any of the results presented here. All results used simple Euclidean distance as ARE's local distance metric over observations. All used a degree three polynomial when computing the sensor model. The final two parameters are the choice of the number of dimensions d in the subjective representation and the number of particles used in MCL. Varying the choice of d from two to eight dimensions in the IMAGEBOT line example affects the resulting accuracy by no more than 2 pixels. Varying the number of particles used in MCL from 50 to 500 caused no difference in the resulting trajectories. In summary, the presented technique has surprisingly few parameters and is quite robust to their choice.

Leaving the Map.

As a final consideration of robustness, an IMAGEBOT trajectory was examined where the test data included objective locations far outside the gathered training data. This means that the synthetic robot left its map for portions of its trajectory. The accuracy measure on the trajectory, averaged over ten runs, was 57 pixels, where the oracle was 37, and random was 158. The high errors for all techniques is due to the fact that for many test points no point in the training data was objectively close. Despite

this, the subjective localization based on ARE continues to perform only marginally worse than the oracle.

6 Conclusion

In summary we examined the problem of subjective localization, where the algorithm can choose an appropriate frame of reference in which to localize. We proposed a technique for solving this problem by (i) extracting a subjective representation from training data using Action Respecting Embedding, (ii) learning a motion model and sensor model for this representation, and (iii) using these models with Monte Carlo localization to track the robot's location in the subjective frame of reference. We evaluated this technique in both a synthetic image manipulation domain and with a mobile robot. The algorithm successfully extracted subjective representations and localized on new test data with substantial accuracy. These results were consistent, with no changes to the algorithm, across a variety of different experiments, including changing the robot's primary sensor from camera to laser. We also showed that the algorithm was robust to the few parameters that it depends upon.

References

- [BGW05] Michael Bowling, Ali Ghodsi, and Dana Wilkinson. Action respecting embedding. In *Proceedings of the Twenty-Second International Conference on Machine Learning*, pages 65–72, 2005.
- [EP04] Austin I. Eliazar and Ronald Parr. Learning probabilistic motion models for mobile robots. In *Proceedings of the Twenty-First International Conference on Machine Learning*. ACM Press, 2004.
- [FBDT99] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 1999.
- [Kal60] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, pages 35–45, 1960.
- [MTKW02] Michael Montemerlo, Sebastian Thrun, Koller Koller, and Ben Wegbreit. Fast-SLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 593–598, 2002.
- [RT99] Nicholas Roy and Sebastian Thrun. Online self-calibration for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999.
- [SC70] P. H. Schoenemann and R. Carroll. Fitting one matrix to another choice of a central dilation and a rigid motion. *Psychometrika*, 35:245–255, 1970.
- [SS02] B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.
- [SS05] Daniel Stronger and Peter Stone. Simultaneous calibration of action and sensor models on a mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.
- [WS04] K. Weinberger and L. Saul. Unsupervised learning of image manifolds by semidefinite programing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 988–995, 2004.

D-SLAM: Decoupled Localization and Mapping for Autonomous Robots

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. The main contribution of this paper is the reformulation of the simultaneous localization and mapping (SLAM) problem for mobile robots such that the mapping and localization can be treated as two concurrent yet separated processes: D-SLAM (decoupled SLAM). It is shown that SLAM can be decoupled into solving a non-linear static estimation problem for mapping and a low-dimensional dynamic estimation problem for localization. The mapping problem can be solved using an Extended Information Filter where the information matrix is shown to be exactly sparse. A significant saving in the computational effort can be achieved for large scale problems by exploiting the special properties of sparse matrices. An important feature of D-SLAM is that the correlation among landmarks are still kept and it is demonstrated that the uncertainty of the map landmarks monotonically decrease. The algorithm is illustrated through computer simulations and experiments.

1 Introduction

Simultaneous localization and mapping (SLAM) has been the subject of extensive research in the past few years with a number of robotics research groups contributing to make substantial progress in this area (see for example, [1], [2], [3], [4], [5], [6], [7] and the references therein). Traditionally, SLAM uses a state vector incorporating the location of the robot, all the landmarks and maintains the associated full covariance matrix. This, however, leads to a heavy computational burden when solving large scale SLAM problems.

Many researchers have exploited the special structure of the covariance matrix in order to reduce the computational effort required in SLAM. One notable result in the recent past has been that of Thrun et al. [7] which uses the Extended Information Filter to exploit the relative sparseness of the information matrix to reduce the computational effort required in SLAM. Frese [8] provided a proof for the approximate sparseness of the information matrix. However, Eustice et al. [9] demonstrated that the process of sparsification proposed in [7] leads to inconsistent estimates.

In a recent development, Eustice et al. [10] show that the inclusion of the robot trajectory in the form of past robot poses in the state vector leads to an exactly sparse information matrix. The resulting Exactly Sparse Delayed State Filter (ESDSF)

provides clear computational advantages when a view-based map representation is used. In the example presented the “map” is not represented within the state vector and is therefore not directly updated.

Another way to reduce the computation complexity is to delete the robot in the map state vector. A variety of attempts have been made to achieve this by constructing relative maps using the observation information. For example, Newman [3] introduced a relative map in which the map state contains the relative locations among the landmarks. Csorba et al. [11] and Martinelli et al. [12] have used relative maps where the map state contains distances (and angles) among the landmarks, which are invariants under shift and rotation. The structure of the covariance matrix is kept sparse by maintaining a state vector with redundant elements. As the relationships between these elements are not enforced, for large scale problems the map becomes complex and difficult to use. However, if the constraints that enforce these relationships are applied, the simple structure of the covariance matrix is destroyed, leading to an increased computational complexity [3].

This paper presents an extension of the decoupled SLAM algorithm, D-SLAM, proposed by the authors in [15] [16], where SLAM is reformulated as a static estimation problem for mapping and a three dimensional dynamic estimation problem for localization. The landmark locations are maintained using either a compact relative map [15] or an absolute Cartesian map [16]. The new formulation retains the significant advantage of being able to improve the location estimates of all the landmarks from one local observation, yet results in an exactly sparse information matrix with the number of nonzero elements related to the range of the sensor on board the robot. The main assumption in [15] [16] is that the robot can observe at least two previously seen landmarks in each observation. This paper provides a strategy to relax the above assumption by merging a set of observations to construct admissible measurements. An improved localization process based on a local SLAM is also presented.

The paper is organized as follows. The mapping and the localization algorithms in D-SLAM are stated in Sections 2 and 3, respectively. The computational complexity is addressed in Section 4. Section 5 provides simulation and experiments results of D-SLAM algorithm. Section 6 concludes the paper by providing a discussion and addressing future research directions.

2 Mapping in D-SLAM

In D-SLAM, the robot location is not included in the state vector in the mapping process. The state vector only contains the Cartesian coordinate of the locations of all the observed landmarks:

$$X = (X_1, \dots, X_n)^T = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)^T. \quad (1)$$

In order to generate estimates of the landmark locations the following two processes are necessary. (1) A method of recasting the observation vector such that the information about the landmarks that is independent of the robot location can be

extracted. (2) A new landmark initialization and update method that does not require the robot location. The following sections provide details of these two processes.

2.1 Recasting the Measurement Vector

Suppose robot observes $m \geq 2$ landmarks f_1, \dots, f_m at a particular time where f_1, f_2 are landmarks that have been previously seen. The raw measurement and the associate Gaussian measurement noise covariance matrix are given by

$$z_{old} = [r_1, \theta_1, \dots, r_m, \theta_m]^T, \quad R_{old} = \text{diag}[R_1, R_2, \dots, R_m]. \quad (2)$$

This measurements can be recast to contain two parts as follows:

$$\begin{bmatrix} z_{rob} \\ z_{map} \end{bmatrix} = \begin{bmatrix} \alpha_{r12} \\ d_{1r} \\ \alpha_{\phi12} \\ \cdots \\ d_{12} \\ \alpha_{312} \\ d_{13} \\ \vdots \\ \alpha_{m12} \\ d_{1m} \end{bmatrix} = \begin{bmatrix} \frac{\text{atan2}\left(\frac{-\tilde{y}_1}{-\tilde{x}_1}\right) - \text{atan2}\left(\frac{\tilde{y}_2 - \tilde{y}_1}{\tilde{x}_2 - \tilde{x}_1}\right)}{\sqrt{(-\tilde{x}_1)^2 + (-\tilde{y}_1)^2}} \\ -\text{atan2}\left(\frac{\tilde{y}_2 - \tilde{y}_1}{\tilde{x}_2 - \tilde{x}_1}\right) \\ \cdots \\ \sqrt{(\tilde{x}_2 - \tilde{x}_1)^2 + (\tilde{y}_2 - \tilde{y}_1)^2} \\ \frac{\text{atan2}\left(\frac{\tilde{y}_3 - \tilde{y}_1}{\tilde{x}_3 - \tilde{x}_1}\right) - \text{atan2}\left(\frac{\tilde{y}_2 - \tilde{y}_1}{\tilde{x}_2 - \tilde{x}_1}\right)}{\sqrt{(\tilde{x}_3 - \tilde{x}_1)^2 + (\tilde{y}_3 - \tilde{y}_1)^2}} \\ \vdots \\ \frac{\text{atan2}\left(\frac{\tilde{y}_m - \tilde{y}_1}{\tilde{x}_m - \tilde{x}_1}\right) - \text{atan2}\left(\frac{\tilde{y}_2 - \tilde{y}_1}{\tilde{x}_2 - \tilde{x}_1}\right)}{\sqrt{(\tilde{x}_m - \tilde{x}_1)^2 + (\tilde{y}_m - \tilde{y}_1)^2}} \end{bmatrix} \quad (3)$$

where

$$\tilde{x}_i = r_i \cos \theta_i, \quad \tilde{y}_i = r_i \sin \theta_i, \quad i = 1, \dots, m. \quad (4)$$

The physical meaning of z_{rob} is the relative angles and distances from the robot to landmarks f_1, f_2 . The physical meaning of z_{map} is the distance between f_1 and f_2 , d_{12} , together with the relative angles and distances from the landmarks f_3, \dots, f_m to landmarks f_1, f_2 .

It is clear that z_{map} contains information about distances and angles among landmarks that are independent of the robot location and the coordinate system. The two measurement vectors z_{rob} and z_{map} are correlated and the associated measurement noise covariance matrices, R_{rob} and R_{map} respectively, are not diagonal although these matrices can be easily computed.

2.2 Mapping Using Information Relating Landmark Locations

The idea for mapping in D-SLAM is the following. (i) When robot is stationary at the starting point, the raw measurement and the robot starting location are used to initialize and update the landmarks location estimates. (ii) Once the robot moves, two previously seen landmarks and the recast observation z_{map} will be used to initialize and update landmarks.

After the robot moves, the measurement model is (assume f_1, f_2 are previously seen landmarks, recall that the state vector X is given in (1))

$$z_{map} = [d_{12}, \alpha_{312}, d_{13}, \dots, \alpha_{m12}, d_{1m}]^T = H_{map}(X) + w_{map} \quad (5)$$

where $H_{map}(X)$ is given by the last $2m - 3$ formulas in equation (3) by substituting \tilde{x}_i, \tilde{y}_i with the absolute locations of the landmarks x_i, y_i ($i = 1, \dots, m$). w_{map} is the new measurement noise whose covariance matrix R_{map} can be computed by (2), (3), and (4).

The mapping problem can now be solved as a linearized minimum mean square error problem. Let $i(k)$ represent information vector and $I(k)$ be the associated information matrix. The state vector and the information vector are related through

$$i(k) = I(k)\hat{X}(k). \quad (6)$$

The procedure for using the measurements z_{map} to update the information vector and the information matrix is as follows:

$$\begin{aligned} I(k+1) &= I(k) + \nabla H_{map}^T R_{map}^{-1} \nabla H_{map} \\ i(k+1) &= i(k) + \nabla H_{map}^T R_{map}^{-1} [z_{map}(k+1) - H_{map}(\hat{X}(k)) + \nabla H_{map}\hat{X}(k)] \end{aligned} \quad (7)$$

where ∇H_{map} is the Jacobian of the function H_{map} with respect to all the states evaluated on the current state estimation $\hat{X}(k)$.

2.3 Construction of Admissible Measurements

To be admissible in the mapping algorithm outlined in the previous section, observation vectors need to satisfy the following condition.

Definition. An observation made at a particular point is called admissible if it contains at least two previously seen landmarks.

Figure 1 shows an example where robot observes two old landmarks f_1, f_2 and two new landmarks f_3, f_4 at point P_1 , but it only observes one landmark f_5 at point P_2 and one other landmark f_6 at point P_3 . Later on at point P_4 , it observes landmarks f_5, f_6, f_7 . Thus the observations at P_2 and P_3 are not admissible. It is, however, possible to combine the measurements made from different points to generate new admissible measurements as follows. Once it is detected that the observation at point P_2 is not admissible, the update to the map using the observation information from P_1 will be removed. Then a virtual observation from P_2 to f_1, f_2, f_3, f_4 will be constructed using the observation from P_1 to f_1, f_2, f_3, f_4 and an estimate of the relative motion of the robot from P_1 to P_2 (Figure 1). The uncertainty associated with this composite observation can be computed using the relevant observation equations and the process and observation uncertainties. The mapping process will proceed as if landmarks f_1, f_2, f_3, f_4, f_5 are observed from P_2 and no observation is made at P_1 . This process is repeated wherever an inadmissible observation is encountered, for example at P_3 . This strategy allows D-SLAM to function where a cluster of landmarks are separated from another cluster of landmarks by a region of "featureless" terrain.

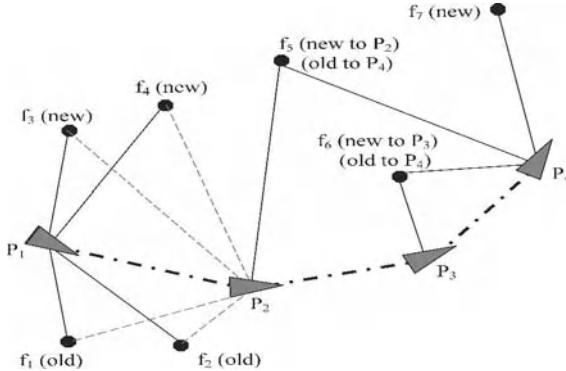


Fig. 1. Construct admissible measurement from the raw measurements

3 Localization in D-SLAM

In the localization process of D-SLAM, estimates from two approaches are combined to obtain an estimate for the robot location (and local landmark locations). One approach is to use a local traditional SLAM. The other is to use the current observation and the map generated in the previous step to solve the kidnapped robot problem. Figure 2 shows a flow-chart illustrating the localization process.

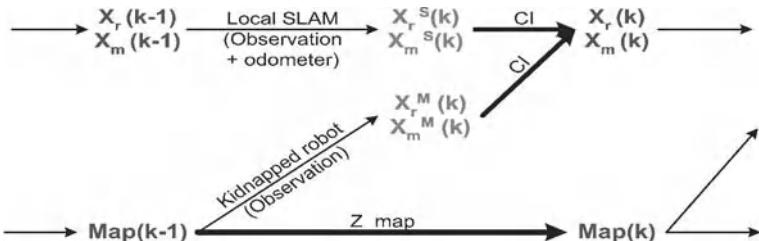


Fig. 2. Flow chart of localization and mapping process in D-SLAM

Suppose that robot observes landmarks f_1, \dots, f_m at time k , among which $f_1, \dots, f_{m_1}, m_1 \leq m$ are landmarks that have been previously seen. The state vector in D-SLAM localization contains the location of the robot and these previously seen landmarks f_1, \dots, f_{m_1} .

$$X_{loc}(k) = (X_r(k), X_1, \dots, X_{m_1})^T. \quad (8)$$

An estimate of X_1, \dots, X_{m_1} and the associated covariance matrix are available from the map obtained at time $k - 1$. These together with the part of the measurement vector z_{old} that involves landmarks f_1, \dots, f_{m_1} ,

$$z_{loc} = (r_1, \theta_1, \dots, r_{m_1}, \theta_{m_1})^T = H_{loc}(X_r(k), X_1, \dots, X_{m_1}) + w_{loc}, \quad (9)$$

can be used to estimate $X_{loc}(k)$. Here H_{loc} contains $2m_1$ typical range and bearing observation functions. The estimate of $X_{loc}(k)$ is a low dimensional linearized minimum mean square error estimation problem. This approach does not make use of the process model and therefore is clearly sub-optimal.

The alternative is to use a local traditional SLAM process to estimate $X_{loc}(k)$, where only the landmarks in the vicinity of the robot are retained in the state vector. Landmarks are removed from the state vector once they are not visible from the robot. When a previously deleted landmark is re-observed, the landmark is reinitialised and is treated as a new landmark. This is effectively a SLAM-aided dead reckoning process which provides a much better robot location estimate than that obtained using dead-reckoning alone.

Which of the two estimates is more accurate depends on the prevailing circumstances. Local SLAM estimate is optimal, until the robot closes a loop by revisiting a previously traversed region of the map. The kidnapped robot solution will be superior when loop closures are present. Fusing the outcomes of the two localization processes will result in a better estimate. However, these two estimates for the robot location are correlated. Therefore, it is necessary to combine these estimates using a strategy, for example covariance intersection (CI) [14], that facilitates combining two correlated pieces of information, when the extent of correlation itself is unknown (see Figure 2).

The robot location computed is sub-optimal and is correlated to the map. These correlations do not affect the mapping process as the observation used for mapping, z_{map} , is independent of the robot location. However, as information about the robot location is not exploited in the mapping process, estimate of the map will also be suboptimal.

4 Computational Complexity

A key feature of D-SLAM is that the information matrix in the mapping process is exactly sparse, and this reduces the computation cost significantly.

Since the measurement z_{map} only involves a small fraction of the total number of landmarks, the matrix $\nabla H_{map}^T R_{map}^{-1} \nabla H_{map}$ in (7) is sparse with the elements relating to the landmarks that are not present in the measurement vector being exactly zero. This can be easily seen by the fact $\nabla H_{map} = \left[\frac{\partial H_{map}}{\partial X_1}, \dots, \frac{\partial H_{map}}{\partial X_m}, 0, \dots, 0 \right]^T$.

In a typical sensor where the sensor range is finite, the observations only relate landmarks that are close to each other. Therefore, if landmark i and landmark j are not visible simultaneously from the robot, the measurement z_{map} will never contain both f_i and f_j . As the information matrix update involves a simple addition, the elements relating to i and j in the information matrix will remain exactly zero. Thus, in a large scale map, a significant portion of the information matrix will be exactly zero, resulting in an exactly sparse information matrix.

Let N be the size of the map. The storage requirement is $O(N)$ because the information matrix is sparse with non-zero elements $O(N)$. Localization step in D-SLAM requires updating a state vector containing only constant number of elements, thus computational cost is $O(1)$. Mapping in D-SLAM is formulated in the information form where the update step is a $O(1)$ time process and the prediction step, the computationally demanding stage of an information filter, does not exist. For data association, locations as well as the uncertainty of the landmarks in the vicinity of the robot are required. The vicinity here is defined in terms of the range of the sensor used and contains only $O(1)$ landmarks.

The major computational cost of D-SLAM is due to the need for recovering the state vector containing the landmark locations and the related parts of the covariance matrix. The state vector can be recovered by solving a sparse linear equation (6). The desired columns of the covariance matrix can also be obtained by solving a constant number of sparse linear equations. Since good initial guesses are available for the linear equations (the previous estimation is a good initial guess for state vector, zero vectors are good initial guesses for the columns of covariance matrix), few iterations are enough for iterative method (for example, Preconditional Conjugate Gradient method) to converge to the solutions. Thus the computation cost for the recovery is $O(N)$. The multigrid algorithm proposed in [13] may also be an efficient method for the recovery. Overall cost of D-SLAM is, therefore, $O(N)$.

5 Evaluation of D-SLAM

5.1 Experimental Evaluation with a Pioneer Robot in an Office Environment

The Pioneer 2 DX robot was used for the experimental implementation. This robot is equipped with a laser range finder with a field of view of 180 degrees and an angular resolution of 0.5 degree. Twelve laser reflectors were placed in a $8 \times 8m^2$ area and the Player software was used to control the robot and collect sensor data.

Matlab implementation of D-SLAM was used to process the data and compute the robot and landmark locations. Nearest neighbour algorithm was used for data association and for comparison, robot and landmark locations were also obtained using the traditional full SLAM algorithm. The results are presented in Figure 3. Although the robot localization estimates are conservative compared to traditional SLAM, the new algorithm provided a much superior estimate to that presented in [16].

5.2 Evaluation of D-SLAM in Simulation with a Large Number of Landmarks

A more complex simulation experiment with larger number of landmarks was conducted to further evaluate D-SLAM and demonstrate its properties. The environment used is a 40 meter square with 196 landmarks arranged in uniformly spaced rows and columns. The robot starts from the left bottom corner of the square and follows a random trajectory, revisiting many landmarks and closing many loops as seen in

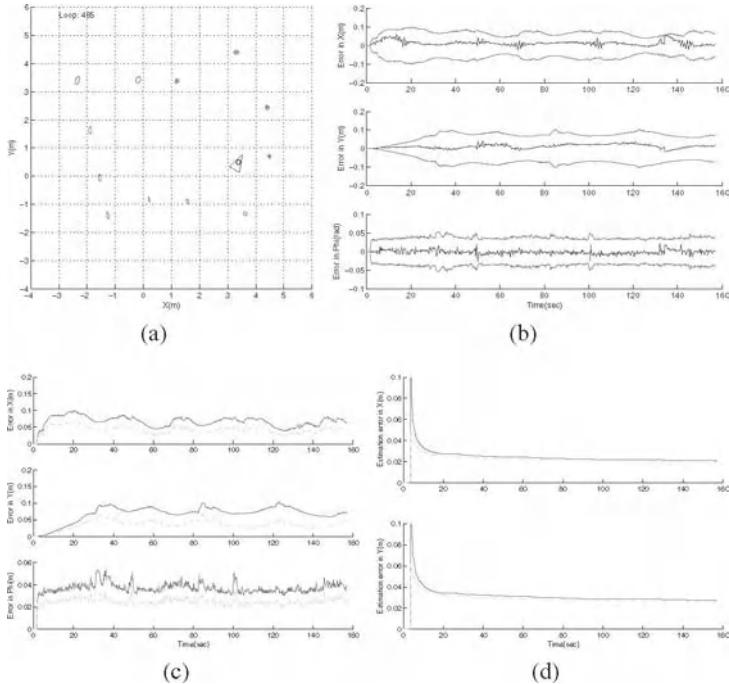


Fig. 3. D-SLAM implementation: (a) Map obtained by D-SLAM; (b) Robot location estimation error; (c) 2σ bounds of robot location estimation (solid line is from D-SLAM; dashed line is from traditional SLAM); (d) 2σ bounds of landmark 9 estimation (solid line is from D-SLAM; dashed line is from traditional SLAM).

Figure 4(a). 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 landmarks.

Figure 4(b) shows the estimation error and the associated 95% confidence levels for one landmark far away from the robot initial location. It is clear that the estimates are consistent. Figure 4(c) shows all the non-zero elements of the information matrix in black after reordering. It is clear that this matrix is sparse as there are 7312 non-zero elements and 68864 exactly zero elements. The blocks diagonal terms are due to landmarks in close vicinity observed together and the off diagonal terms are due to loop closures where a previously seen landmark is re-observed some time later. Reordering the information matrix, so that indices of the nearby landmarks are adjacent, results in the banded matrix. This matrix demonstrates the fact that only the nearby landmarks are linked in the information matrix.

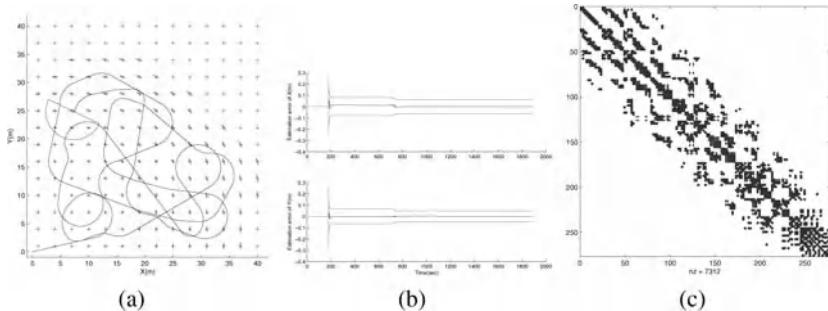


Fig. 4. D-SLAM simulations: (a) Map obtained by D-SLAM; (b) Estimation error of a landmark far away from robot starting location, and its 95% confidence limit; (c) Sparse information matrix obtained by D-SLAM after reordering (7312 non-zero elements and 68864 exactly zero elements).

6 Discussion and Conclusions

In this paper, a new decoupled SLAM algorithm: D-SLAM, is described. While the localization and mapping are performed simultaneously, mapping and localization are separated processes. The significant advantages gained are that there is no prediction step for the mapping, the information matrix associated with mapping is exactly sparse and only the landmarks that are in the close vicinity are linked through the information matrix. This results in an $O(N)$ SLAM algorithm where N is the number of landmarks.

Although the robot location is not incorporated in the state vector used in mapping, correlations between the landmarks are still preserved. Thus the location estimates of all the landmarks are improved using information from one local observation.

In D-SLAM, however, the knowledge about the robot location is not exploited in the mapping process and this results in some information loss. An analysis based on a linear one-dimensional model as well as 2-D simulations demonstrated that the information loss depends on the ratio between the sensor noise and the process noise. The smaller the ratio, the less amount of information lost. Further analytical work to quantify the extent of information loss is currently underway.

Additional work is necessary to further reduce the computation effort by exploring the possibilities of using D-SLAM in conjunction with the submap idea (e.g. [5]). Investigations in these directions together with a large scale experiment using Victoria Park data set [17] are currently in progress. Further work is required to compare D-SLAM with the recent developments in view-based SLAM [10]. In view-based SLAM the state vector consists of robot poses whereas the map is obtained through registration of successive observation sets. In D-SLAM, the map is represented in the state vector and one localization estimate is generated by registering the robot in the map. Both approaches result in significant computational advantages at the expense of some information loss. Examination of the relationship between D-SLAM with

the FastSLAM algorithm where particles are used to represent the possible robot trajectories will also be interesting.

Acknowledgment

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

References

1. Dissanayake G, Newman P, Clark S, Durrant-Whyte H, Csorba M (2001) "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. on Robotics and Automation*, vol. 17, pp. 229–241
2. 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*, vol. 17, pp. 242–257
3. 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
4. Castellanos JA, Neira J, Tardos JD (2001) "Multisensor fusion for simultaneous localization and map building," *IEEE Trans. on Robotics and Automation*, vol. 17, 908–914
5. Bosse M, Newman P, Leonard JJ, and Teller S (2004) "SLAM in Large-scale Cyclic Environments using the Atlas Framework", *International Journal on Robotics Research*, vol. 23 (12), pp. 1113–1139
6. Folkesson J, Christensen HI (2004) "Graphical SLAM - A Self-Correcting Map," *In Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, LA, New Orleans, pp. 383–390
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*, vol. 23, pp. 693–716
8. Frese U (2005) "A Proof for the Approximate Sparsity of SLAM Information Matrices," *In Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, pp. 331–337
9. Eustice RM, Walter M, Leonard JJ (2005) "Sparse Extended Information Filters: Insights into Sparsification," *In Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Alberta, Canada, August, pp. 641–648
10. Eustice RM, Singh H, Leonard JJ (2005) "Exactly sparse delayed-state filters," *In Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, pp. 2428–2435
11. Csorba M, Uhlmann JK, Durrant-Whyte H (1997) "A suboptimal algorithm for automatic map building," *In Proceedings of 1997 American Control Conference*, USA, pp. 537–541
12. Martinelli A, Tomatis N, Siegwart R (2004) "Open challenges in SLAM: An optimal solution based on shift and rotation invariants," *In Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, LA, New Orleans, pp. 1327–1332
13. Frese U, Larsson P, Duckett T (2005) "A Multigrid Algorithm for Simultaneous Localization and Mapping," *IEEE Transactions on Robotics*, vol. 21 (2), pp. 1–12

14. Chen L, Arambel PO, Mehra RK (2002) "Estimation under unknown correlation: covariance intersection revisited," *IEEE Transactions on Automatic Control*, 47(11), pp. 1879–1882
15. Wang Z, Huang S, Dissanayake G (2005) "Decoupling Localization and Mapping in SLAM Using Compact Relative Maps," *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Alberta, Canada, pp. 1041–1046
16. Wang Z, Huang S, Dissanayake G (2005) "Implementation Issues and Experimental Evaluation of D-SLAM," *In Proceedings of the 5th International Conference on Field and Service Robotics (FSR)*, Port Douglas, Australia. pp. 153–164
17. Nebot EM, UTE Experimental Data from Victoria Park, available online http://www.acfr.usyd.edu.au/homepages/academic/enebot/experimental_data_ute.htm

A Provably Consistent Method for Imposing Sparsity in Feature-Based SLAM Information Filters

Matthew Walter¹, Ryan Eustice², and John Leonard¹

¹ Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge MA, USA {mwalter,jleonard}@mit.edu

² Department of Applied Ocean Physics and Engineering
Woods Hole Oceanographic Institution
Woods Hole MA, USA ryan@whoi.edu *

Summary. An open problem in Simultaneous Localization and Mapping (SLAM) is the development of algorithms which scale with the size of the environment. A few promising methods exploit the key insight that representing the posterior in the canonical form parameterized by a sparse information matrix provides significant advantages regarding computational efficiency and storage requirements. Because the information matrix is naturally dense in the case of feature-based SLAM, additional steps are necessary to achieve sparsity. The delicate issue then becomes one of performing this sparsification in a manner which is consistent with the original distribution.

In this paper, we present a SLAM algorithm based in the information form in which sparseness is preserved while maintaining consistency. We describe an intuitive approach to controlling the population of the information matrix by essentially ignoring a small fraction of proprioceptive measurements whereby we track a modified version of the posterior. In this manner, the Exactly Sparse Extended Information Filter (ESEIF) performs exact inference, employing a model which is conservative relative to the standard distribution. We demonstrate our algorithm both in simulation as well as on two nonlinear datasets, comparing it against the standard EKF as well as the Sparse Extended Information Filter (SEIF) by Thrun *et al.* The results convincingly show that our method yields conservative estimates for the robot pose and map which are nearly identical to those of the EKF in comparison to the SEIF formulation which results in overconfident error bounds.

* R. Eustice is now with the Department of Mechanical Engineering at The Johns Hopkins University, Baltimore, MD USA, E-mail: rme@jhu.edu.

1 Introduction

A skill which plays an integral role in achieving robot autonomy is the ability to operate in *a priori* unknown environments. Viewed as a coupled problem of simultaneously performing localization and mapping, SLAM is further complicated by the stochastic nature of vehicle motion and observations. Most effective SLAM algorithms address these issues by posing the problem in a probabilistic framework with the goal then being the estimation of the joint distribution over the map and vehicle pose.

Beginning with the seminal work of Smith *et al.* [14], the Extended Kalman Filter (EKF) SLAM formulation has proven to be particularly popular. In large part, this is due to its relative simplicity, requiring that one only maintain the first two moments of the distribution to account for the coupling between the robot and map. From knowledge of the correlation, the EKF is able to exploit feature observation data to update the pose and map estimates. At the same time, this capability comes at the cost of complexity which is quadratic in the number of state elements. As a result, SLAM algorithms relying upon an EKF have traditionally been limited to relatively small environments.

Representing the joint Gaussian distribution in the dual canonical form, recent work has given rise to algorithms capable of scaling with the environment. Pivotal insights by Thrun *et al.* [15] and Frese *et al.* [8] have revealed that, in the context of SLAM, many of the off-diagonal elements in the inverse covariance (information) matrix are inherently near zero. Considering the graphical model represented by the information matrix [12], the implication is that a majority of the links in the Markov network are relatively weak. By essentially breaking these weak links, Frese [7] and Paskin [12] are able to approximate the graphical model by a sparse tree structure which provides for scalable SLAM algorithms. Alternatively, the Sparse Extended Information Filter (SEIF) by Thrun *et al.* [15] relies upon a version of the Extended Information Filter, the dual to the EKF. In the case where the information matrix is sparse, the authors demonstrate that state estimation can be performed in near-constant time. While a majority of the links in the information matrix are weak, though, they are nonetheless nonzero. SEIFs then employ a strategy by which the posterior is approximated with an information matrix having the desired sparse structure. The algorithm has efficiently been applied to large, real-world datasets with *a priori* unknown data association [10].

Together with the intuitive characteristics of the canonical representation noted in [15], the success of SEIFs has brought a lot of attention to the information filter formulation of the SLAM problem. The one issue which has, up to now, largely gone unnoticed is the implication of approximating the posterior to achieve the necessary sparseness. A close look at the sparsification strategy reveals that the resulting posterior is prone to overconfidence. In [5], the authors show that, while the state estimates are only slightly overconfident when expressed in a local reference frame, they suffer from an exaggerated global inconsistency. The paper presents a modified sparsification rule which

yields a posterior which is both locally and globally consistent relative to the full Kalman solution but is no longer computationally tractable.

Our objective in this paper is to present an information-based formulation to the SLAM problem which achieves exact sparseness while being computationally efficient. Rather than relying upon an approximation to remove links from the information matrix, the algorithm adopts a new strategy which actively controls the population of the matrix by relocating the robot within the map. The filter then maintains an estimate of the state which is both globally *and* locally conservative relative to the full Kalman solution. We demonstrate the algorithm alongside the EKF and SEIF on a linear Gaussian simulation as well as two real-world experiments, including a benchmark nonlinear dataset. The results reveal that while the SEIF is globally inconsistent, our algorithm yields estimates nearly identical to those of the EKF which are globally and locally conservative.

2 Information Filter

2.1 Canonical Form

Let ξ_t be a random vector having a Gaussian probability density, $\xi_t \sim \mathcal{N}(\xi_t; \mu_t, \Sigma_t)$ described completely by its mean, μ_t , and covariance matrix, Σ_t . An expansion of the exponential term defining the multivariate normal distribution, $p(\xi_t) \propto \exp\left\{-\frac{1}{2}(\xi_t - \mu_t)^\top \Sigma_t^{-1} (\xi_t - \mu_t)\right\}$, yields an equivalent representation for the probability density function, $\mathcal{N}^{-1}(\xi_t; \eta_t, \Lambda_t)$, parameterized by the information vector and information matrix, η_t and Λ_t , respectively.

$$\Lambda_t = \Sigma_t^{-1} \quad \eta_t = \Sigma_t^{-1} \mu_t \quad (1)$$

The canonical representation for the multivariate Gaussian is the dual of the standard form in the sense of the fundamental processes of marginalization and conditioning, as exemplified in Table 1. While marginalization is *hard* in the information form, requiring a matrix inversion, it is *easy* in the covariance form. The opposite is true in regards to the conditioning operation. Further details regarding this duality in the context of filtering can be found in [11].

One quality of the canonical form is its relationship with Gaussian Markov random fields in which nodes in the graph represent individual state variables and edge structure describes their conditional independence relationships. The information matrix effectively serves as an adjacency matrix for the graph [12], with the strength of constraints between pairs of variables proportional to the corresponding elements of the matrix. Off-diagonal components which are zero then denote the absence of links in the Markov network. Thus, the information matrix has the particular advantage of explicitly representing the conditional independence of state variables.

Table 1. Summary of Marginalization and Conditioning Operations on a Gaussian Distribution Expressed in Covariance and Information Form
$$p(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_\alpha \\ \boldsymbol{\mu}_\beta \end{bmatrix}, \begin{bmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\alpha\beta} \\ \Sigma_{\beta\alpha} & \Sigma_{\beta\beta} \end{bmatrix}\right) = \mathcal{N}^{-1}\left(\begin{bmatrix} \boldsymbol{\eta}_\alpha \\ \boldsymbol{\eta}_\beta \end{bmatrix}, \begin{bmatrix} \Lambda_{\alpha\alpha} & \Lambda_{\alpha\beta} \\ \Lambda_{\beta\alpha} & \Lambda_{\beta\beta} \end{bmatrix}\right)$$

	MARGINALIZATION	CONDITIONING
	$p(\boldsymbol{\alpha}) = \int p(\boldsymbol{\alpha}, \boldsymbol{\beta}) d\boldsymbol{\beta}$	$p(\boldsymbol{\alpha} \boldsymbol{\beta}) = p(\boldsymbol{\alpha}, \boldsymbol{\beta}) / p(\boldsymbol{\beta})$
COVARIANCE FORM	$\boldsymbol{\mu} = \boldsymbol{\mu}_\alpha$ $\Sigma = \Sigma_{\alpha\alpha}$	$\boldsymbol{\mu}' = \boldsymbol{\mu}_\alpha + \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)$ $\Sigma' = \Sigma_{\alpha\alpha} - \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}\Sigma_{\beta\alpha}$
INFORMATION FORM	$\boldsymbol{\eta} = \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\boldsymbol{\eta}_\beta$ $\Lambda = \Lambda_{\alpha\alpha} - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\beta\alpha}$	$\boldsymbol{\eta}' = \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\boldsymbol{\beta}$ $\Lambda' = \Lambda_{\alpha\alpha}$

2.2 Feature-Based SLAM

The goal of any SLAM algorithm is to concurrently perform navigation and map-building in the presence of uncertainty in vehicle motion and environmental observations. With feature-based SLAM formulations, the map is described as a collection of stationary primitives, e.g. lines, points, etc. The robot pose, \mathbf{x}_t , together with the set of map elements, $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n\}$, are represented together by the state vector, $\boldsymbol{\xi}_t = [\mathbf{x}_t^\top \mathbf{M}^\top]^\top$. The coupling between the pose and map is addressed by considering the joint probability distribution for the state. Adopting a Bayesian framework, a model of the joint posterior is tracked as it evolves as a result of the uncertainty in vehicle motion and measurement data. Typical SLAM implementations make the assumption that this uncertainty is a result of independent white Gaussian noise which corrupts the motion and measurement models. One can then show that the posterior obeys a Gaussian distribution.

$$p(\boldsymbol{\xi}_t | \mathbf{z}^t, \mathbf{u}^t) = \mathcal{N}(\boldsymbol{\xi}_t; \boldsymbol{\mu}_t, \Sigma_t) = \mathcal{N}^{-1}(\boldsymbol{\xi}_t; \boldsymbol{\eta}_t, \Lambda_t) \quad (2)$$

The belief function is traditionally represented in the standard form which can be tracked relatively easily with the EKF. Modifying the posterior to reflect the effect of vehicle motion is a constant-time process as it involves a combined process of state augmentation and marginalization, both of which are easily performed in the covariance form. On the other hand, it is well known that incorporating new measurement data requires a conditioning step which is quadratic in the size of the state. Furthermore, maintaining the correlation among state estimates leads to a dense covariance matrix which must be stored. For small scale environments, these problems are surmountable, but as the map size becomes increasingly large, implementing a full EKF quickly becomes intractable.

Alternatively, employing the canonical representation of the posterior, the filtering process reflects the duality between the two forms. Performing measurement updates (conditioning) is constant-time, while the marginalization component of the projection step, in general, is quadratic in the state dimension, at best. Furthermore, recovering the estimate of the mean requires the $\mathcal{O}(n^3)$ inversion of the information matrix per (1). As a result of these limitations, the information filter has had relatively limited use in SLAM.

Recently, Thrun *et al.* [15] and Frese *et al.* [8] have made the pivotal observation that, when normalized, the information matrix tends to be nearly sparse. The matrix is dominated by a small percentage of terms which are significantly larger than the remaining elements. In general, the links between the robot and the map are stronger for nearby, recently observed features while the constraints are weak for distant features. The same is true for inter-landmark terms which tend to decay exponentially with the distance traversed by the robot between observations [6]. Referring to the graphical interpretation of the information matrix, these weak links then imply that, given relatively few features, the robot is nearly conditionally independent of much of the map.

Though many of the terms in the normalized information matrix are very small, the SLAM process naturally leads to the full population of the matrix. To get a better understanding of why this is, consider a simple example in which the map consists of five features. Suppose that the off-diagonal terms in the information matrix corresponding to the robot, \mathbf{x}_t , are non-zero for four of the features and that the remaining landmark, \mathbf{m}_4 , has shared information with another feature. These links between the robot and the map are created when features are observed. The graphical model along with the information matrix are illustrated in the left-hand side of Figure 1(a). The time projection step can be viewed as an initial augmentation of the state with the new robot pose, \mathbf{x}_{t+1} , which, evolving by a Markov process, is linked only to the previous pose as indicated in the middle figure. At this point, the information matrix remains sparse. Subsequently marginalizing out \mathbf{x}_t , though, creates links between all states which share constraints with the previous pose. The result is a fully connected subset of nodes and, correspondingly, a population of the information matrix. The only remaining zero entries correspond to the lone feature, \mathbf{m}_4 , which will become linked to the robot upon the next observation. The time projection step will then lead to a fully connected graph and, correspondingly, a dense information matrix.

Hence, with *online* SLAM implementations in which only the current pose of the robot is estimated, the marginalization of the previous pose in the projection step naturally results in a dense information matrix. Alternatively, by retaining an entire trajectory history, exact sparsity can be maintained [3] at the cost of storage requirements which become significant for large datasets.

Returning to the example pictorialized in Figure 1(a), note that while the time projection step populates the information matrix, the strength of the off-diagonal links decays with time. This behavior is the reason why a majority of the elements in the normalized matrix are very small. The authors show

in [15, 8, 12] that if the posterior can be represented by an exactly sparse approximation, it is possible to achieve significant gains when it comes to both storage and time requirements. In particular, a bound on the number of links between the robot and the map allows for near constant-time performance of the time projection step and also controls the fill-in of the information matrix resulting from marginalization.

3 Exactly Sparse Extended Information Filters

Map elements having shared information with the robot are said to be *active*. In feature-based SLAM implementations, a feature becomes active when it is first observed. With time, the strength of the link with the robot decays and is strengthened only upon being re-observed. Thus, while the off-diagonal terms may become arbitrarily small, they will never become zero. In order for a landmark to become *passive* (i.e. no shared information), the link with the robot must explicitly be broken.

3.1 Problem Formulation

In describing the desired sparsity of the information matrix, we adopt the two measures utilized by Thrun *et al.* [15]. Denote the maximum allowable number of active features as Γ_a and the number of inter-landmark links in the matrix by Γ_p . Let us then partition the map into two sets of features, $\mathbf{M} = \{\mathbf{m}^+, \mathbf{m}^-\}$, where \mathbf{m}^+ represents the active features for which the off-diagonal terms for the robot pose are non-zero, and \mathbf{m}^- denotes the passive landmarks, having no direct constraint to the vehicle.

Controlling the sparsity of the information matrix is, in large part, a direct consequence of maintaining the Γ_a bound. By regulating the number of active features, it is possible to limit the population of the matrix. Consider, for example, the situation depicted in the left-hand side of Figure 1(b) in which four of the five features are active. At this point, if \mathbf{x}_t were marginalized out, the four active features in \mathbf{m}^+ would become fully connected, potentially violating the Γ_p bound. Instead, if one of landmarks, \mathbf{m}_1 , were first made passive, the number of non-zero elements created as a result of marginalization could be controlled. Thus, enforcing the desired sparsity pattern corresponds to maintaining a bound on the number of active features. Since features do not naturally become passive, a sparsification routine which deliberately breaks the links is necessary.

3.2 SEIF Sparsification

The SEIF breaks a link between the robot and a feature by approximating the posterior with a distribution in which the robot is conditionally independent of

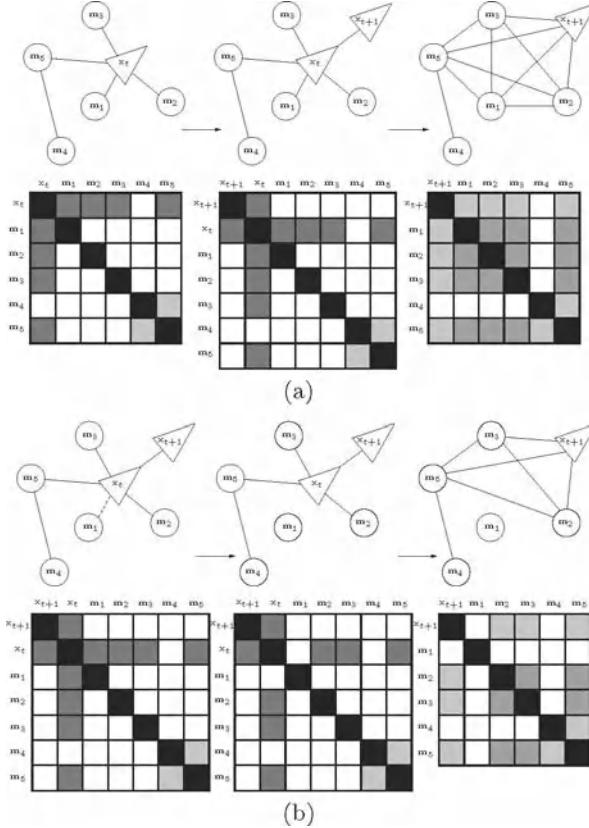


Fig. 1. A graphical explanation of SEIF’s methodology for controlling sparsity in the information matrix. (a) A sequence of illustrations depicting the evolution of the Markov network and corresponding information matrix resulting from time projection when viewed as a two-step process of state augmentation followed by marginalization. Darker shades imply larger magnitudes with white indicating zero values. From left to right we have: (1) the robot \mathbf{x}_t connected to four active features, $\mathbf{m}_{1:3}$ and \mathbf{m}_5 ; (2) state augmentation of the time-propagated robot pose \mathbf{x}_{t+1} ; (3) marginalized distribution where the old pose, \mathbf{x}_t , has been eliminated. (b) A sequence of illustrations highlighting the concept behind sparsification. If feature \mathbf{m}_1 can first be made passive by eliminating its link to the old pose, \mathbf{x}_t , then marginalization over \mathbf{x}_t will not link it to the other active features. This implies that we can control fill-in of the information matrix by bounding the number of currently active features.

the landmark. The map is broken into three disjoint sets, $\mathbf{M} = \{\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\}$, where \mathbf{m}^- refers to the passive landmarks which will remain passive and, in a slight abuse of notation, \mathbf{m}^+ is the set of active features which will remain active, and \mathbf{m}^0 are the active features which will be made passive. The sparsification routine proceeds from a decomposition of the posterior

$$\begin{aligned} p(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) &= p(\mathbf{x}_t | \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) p(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) \\ &= p(\mathbf{x}_t | \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^- = S_{\mathbf{m}^-}^\top \boldsymbol{\mu}_t) p(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) \end{aligned} \quad (3)$$

where setting the passive elements to their mean, $S_{\mathbf{m}^-}^\top \boldsymbol{\mu}_t$, in the last line is valid due to their conditional independence with respect to the robot. SEIFs then deactivate the landmarks by replacing (3) with an approximation to the posterior which drops the dependence upon \mathbf{m}^0 :

$$\tilde{p}_{\text{SEIF}}(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) = p(\mathbf{x}_t | \mathbf{m}^+, \mathbf{m}^- = S_{\mathbf{m}^-}^\top \boldsymbol{\mu}_t) p(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) \quad (4)$$

While the decomposition in (3) is theoretically sound, it is no longer valid to condition on a particular value for the passive features while simultaneously ignoring the dependence upon \mathbf{m}^0 . Given only a subset of the active features, the robot pose is no longer conditionally independent of the passive map.

By enforcing the conditional independence between the robot and the deactivated features, SEIFs rely upon approximate inference on an approximate posterior and, as a result, are prone to inconsistency [5]. In particular, the authors show that sparsifying in this manner leads to a *global* map which is significantly overconfident while the *local* relationships are preserved.

3.3 ESEIF Sparsification

Rather than deliberately breaking constraints with the robot to maintain a bound on the number of active features, ESEIFs take the approach of essentially controlling the initial formation of links. As soon as a feature is first observed, it is linked to the current robot pose. As noted earlier, the strength of this constraint will decay with time but never truly disappear, leading to a growing number of links between the robot and the map.

Noting the nature of this link formation, ESEIFs control the number of active features by deliberately marginalizing out the robot pose. The vehicle is relocated within the map using observations of a few known landmarks. The new pose is then conditionally independent of the rest of the map, and the robot is linked only to the features used for relocalization.

For a more detailed description of the ESEIF sparsification strategy, we consider a situation which would give rise to the representation in Figure 1 which consists of both active and passive features. Suppose that the robot makes four observations, $\mathcal{Z}_t = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_5\}$, three being of active features and one of a passive feature:

$$\begin{array}{ll} \mathbf{z}_2 = \mathbf{h}(\mathbf{x}_v, \mathbf{m}_2), \mathbf{m}_2 \in \mathbf{m}^+ & \mathbf{z}_5 = \mathbf{h}(\mathbf{x}_v, \mathbf{m}_5), \mathbf{m}_5 \in \mathbf{m}^+ \\ \mathbf{z}_1 = \mathbf{h}(\mathbf{x}_v, \mathbf{m}_1), \mathbf{m}_1 \in \mathbf{m}^0 & \mathbf{z}_3 = \mathbf{h}(\mathbf{x}_v, \mathbf{m}_3), \mathbf{m}_3 \in \mathbf{m}^- \end{array}$$

Updating the posterior based upon all four measurements would result in the strengthening of the off-diagonal entries in the matrix pairing the robot with the three observed active features. Additionally, a link would be created

with the currently passive map element, \mathbf{m}_3 , leading to the graph structure depicted in the left-hand side of Figure 1(a). In the case where this would lead to a violation of the Γ_a bound, one strategy would be to disregard the observation of the passive feature entirely. With ESEIFs, though, it is possible to incorporate all measurement data while maintaining the desired sparsity pattern.

In the ESEIF sparsification step, the measurement data is partitioned into two sets, \mathbf{z}_α and \mathbf{z}_β , where the first set is used for updating the filter and the second is reserved for performing relocalization. Of the four measurements available in our example, group that of the passive feature together with one active measurement as $\mathbf{z}_\alpha = \{\mathbf{z}_1, \mathbf{z}_3\}$, leaving $\mathbf{z}_\beta = \{\mathbf{z}_2, \mathbf{z}_5\}$. To sparsify, we first apply the update step followed by the combined process of marginalization and relocalization.

Posterior Update

A Bayesian update is performed on the joint posterior, $p(\boldsymbol{\xi}_t \mid \mathbf{z}^{t-1}, \mathbf{u}^t) = \mathcal{N}^{-1}(\boldsymbol{\xi}_t; \boldsymbol{\eta}_t, \Lambda_t)$ based upon the \mathbf{z}_α measurements:

$$p(\boldsymbol{\xi}_t \mid \mathbf{z}^{t-1}, \mathbf{u}^t) \xrightarrow{\mathbf{z}_\alpha = \{\mathbf{z}_1, \mathbf{z}_3\}} p_1(\boldsymbol{\xi}_t \mid \{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\}, \mathbf{u}^t)$$

where $p_1(\boldsymbol{\xi}_t \mid \{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\}, \mathbf{u}^t) = \mathcal{N}^{-1}(\boldsymbol{\xi}_t; \bar{\boldsymbol{\eta}}, \bar{\Lambda})$ follows from the standard update process for the information filter. Note that we can perform this step in constant-time with, in the nonlinear case, access to the mean estimate for the robot as well as \mathbf{m}_1 and \mathbf{m}_3 . The information matrix, $\bar{\Lambda}_t$, is modified as depicted in Figure 2 with the strengthening of the constraints between the vehicle and the active feature, \mathbf{m}_1 and importantly, the creation of shared information with the previously passive feature, \mathbf{m}_3 .

Marginalization and Relocalization

The addition of a new constraint between the robot and a map element results in a violation of, Γ_a , the bound on the number of active features. The ESEIF sparsification routine then proceeds by first marginalizing out the vehicle pose

$$\begin{aligned} p_2(\mathbf{M}_t \mid \{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\}, \mathbf{u}^t) &= \int_{\mathbf{x}_t} p_1(\boldsymbol{\xi}_t \mid \{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\}, \mathbf{u}^t) d\mathbf{x}_t \\ &= \mathcal{N}^{-1}(\mathbf{M}_t; \check{\boldsymbol{\eta}}_t, \check{\Lambda}_t) \end{aligned}$$

Following the representation of the marginalization process presented in Table 1, the canonical parameterization of the marginal is calculated as

$$p_2(\mathbf{M}_t \mid \{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\}, \mathbf{u}^t) = \mathcal{N}^{-1}(\mathbf{M}_t; \check{\boldsymbol{\eta}}_t, \check{\Lambda}_t)$$

$$\begin{aligned} \check{\Lambda}_t &= S_{\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-}^\top \bar{\Lambda}_t S_{\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-} \\ &\quad - S_{\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-}^\top \bar{\Lambda}_t S_{\mathbf{x}_t} (S_{\mathbf{x}_t}^\top \bar{\Lambda}_t S_{\mathbf{x}_t})^{-1} S_{\mathbf{x}_t}^\top \bar{\Lambda}_t S_{\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-} \end{aligned} \quad (5a)$$

$$\check{\boldsymbol{\eta}}_t = S_{\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-}^\top \check{\boldsymbol{\eta}}_t - S_{\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-}^\top \bar{\Lambda}_t S_{\mathbf{x}_t} (S_{\mathbf{x}_t}^\top \bar{\Lambda}_t S_{\mathbf{x}_t})^{-1} S_{\mathbf{x}_t}^\top \check{\boldsymbol{\eta}}_t \quad (5b)$$

where $S_{\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-}$ and $S_{\mathbf{x}_t}$ are projection matrices mapping the state space to the $\{\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\}$ and \mathbf{x}_t subspaces, respectively.

The inverse term involves the block diagonal of the information matrix corresponding to the vehicle pose, $S_{\mathbf{x}_t}^\top \bar{\Lambda}_t S_{\mathbf{x}_t}$, which is of fixed size. Meanwhile, the $S_{\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-}^\top \bar{\Lambda}_t S_{\mathbf{x}_t}$ matrix corresponds to the shared information between the map and the vehicle pose and, taken as an outer product over the vehicle sub-block, yields a matrix having nonzero values only for the active feature indices. It is a result of this term that marginalization establishes the connectivity among the active features shown in the right-hand side of Figure 2. The computational complexity of this matrix outer product is limited by the T_a bound and the order of the matrix inversion is fixed. Thus, the marginalization can be performed in constant-time.

We complete sparsification in ESEIFs by relocating the vehicle within the map using the remaining \mathbf{z}_β measurements. The new pose estimate is, in general, given by a nonlinear function of measurement data and corresponding feature estimates of the form in (6a) where $\mathbf{w}_t \sim \mathcal{N}(\mathbf{w}_t; \mathbf{0}, \mathbf{R})$ is white Gaussian noise. Equation (6b) corresponds to the linearization about the mean of the marginal distribution, $\mathcal{N}^{-1}(\mathbf{M}_t; \check{\boldsymbol{\eta}}_t, \Lambda)$ in (5). The Jacobian, \mathbf{G} , is sparse as the only non-zero columns are those corresponding to the map elements used for relocation. Subsequently, only the mean estimates for these features are necessary for the linearization.

$$\mathbf{x}_t = \mathbf{g}(\mathbf{m}_\beta, \mathbf{z}_\beta) + \mathbf{w}_t \quad (6a)$$

$$\approx \mathbf{g}(\check{\boldsymbol{\mu}}_{m_\beta}, \mathbf{z}_\beta) + \mathbf{G}(\mathbf{m} - \check{\boldsymbol{\mu}}_t) + \mathbf{w}_t \quad (6b)$$

Augmenting the map distribution (5) with the new pose estimate yields a state which can be shown to have the following canonical parameterization:

$$p_{\text{ESEIF}}(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t) = \mathcal{N}^{-1}(\boldsymbol{\xi}_t; \check{\boldsymbol{\eta}}_t, \check{\Lambda}_t)$$

$$\check{\boldsymbol{\eta}}_t = \begin{bmatrix} \mathbf{R}^{-1}(\mathbf{g}(\check{\boldsymbol{\mu}}_{m_\beta}, \mathbf{z}_\beta) - \mathbf{G}\check{\boldsymbol{\mu}}_t) \\ \check{\boldsymbol{\eta}}_t - \mathbf{G}^\top \mathbf{R}^{-1}(\mathbf{g}(\check{\boldsymbol{\mu}}_{m_\beta}, \mathbf{z}_\beta) - \mathbf{G}\check{\boldsymbol{\mu}}_t) \end{bmatrix} \quad (7a)$$

$$\check{\Lambda}_t = \begin{bmatrix} \mathbf{R}^{-1} & -\mathbf{R}^{-1}\mathbf{G} \\ -\mathbf{G}^\top \mathbf{R}^{-1} (\check{\Lambda}_t + \mathbf{G}^\top \mathbf{R}^{-1}\mathbf{G}) & \end{bmatrix} \quad (7b)$$

Due to the sparsity of \mathbf{G} , most terms in $-\mathbf{R}^{-1}\mathbf{G}$ of the information matrix in (7b) that link the robot to the map are zero, except for those corresponding to the landmarks used for relocation. The new instantiation for the

robot pose is then conditionally independent of the rest of the map. As a result, ESEIF sparsification leads to the joint posterior having the desired factorization:

$$p_{\text{ESEIF}}(\xi_t \mid \mathbf{z}^t, \mathbf{u}^t) = p(\mathbf{x}_t \mid \mathbf{m}_\beta, \mathbf{z}_\beta) p_2(\mathbf{M}_t \mid \{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\}, \mathbf{u}^t)$$

As reflected by the resulting information matrix depicted in Figure 2, the active features are then limited to those used for relocalization.

In this manner, ESEIFs control the size of the active map and, in turn, the sparseness of the information matrix. Like the full EKF, the ESEIF performs exact inference on an approximate model, albeit on a different posterior. When we first marginalize out (kidnap) and subsequently relocalize the robot, we are performing the dual of kidnapping and relocation for the standard EKF. Essentially, we are ignoring the odometry data which links the current and previous poses. Hence, whereas the full EKF tracks the Gaussian approximation to the posterior, $p(\xi_t \mid \mathbf{Z}^t)$, ESEIFs and the relocated EKF maintain the Gaussian model of an alternate distribution, $p(\xi_t \mid \mathbf{Z}^{t^*})$. In this way, the ESEIF employs exact inference on an approximate model for which the information matrix is exactly sparse.

3.4 Recovering the Mean

A drawback of representing the posterior in the canonical form is that we no longer have access to the mean vector or covariance matrix. When the system equations are nonlinear, a subset of the mean is required to perform linearizations. Naively, we could recover the entire mean vector as $\boldsymbol{\mu}_t = \Lambda_t^{-1} \boldsymbol{\eta}_t$, though this operation is cubic in the dimension of the state and quickly becomes intractable. Instead, we can pose the problem in terms of solving a set of linear equations

$$\Lambda_t \boldsymbol{\mu}_t = \boldsymbol{\eta}_t \quad (8)$$

and take advantage of the sparseness of the information matrix. There are a number of techniques which iteratively solve such sparse, symmetric positive definite systems including conjugate gradient descent [13] and, more recently, the multilevel method proposed by [9]. Aside from loop closures, the mean vector evolves rather slowly in SLAM and, thus, the optimization can be performed over the course of multiple time steps. This then allows us to bound the number of iterations required per time step [2].

3.5 Data Association

Traditionally, the problem of data association is addressed by evaluating the likelihood of a measurement for different correspondence hypothesis. The distribution follows from marginalizing out all state elements except for the variables we are interested in, \mathbf{x}_i and \mathbf{x}_j . From the duality indicated in Table 1,

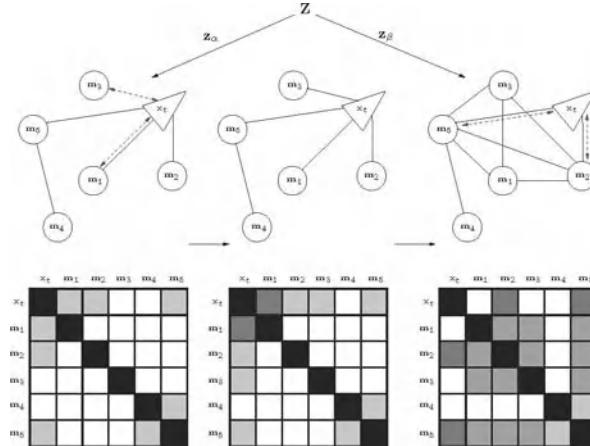


Fig. 2. Sparsification as performed by ESEIFs during the measurement update step. At time t , three of the mapped features are active, $\mathbf{m}^+ = \{m_1, m_2, m_5\}$ and two are passive, $\mathbf{m}^- = \{m_3, m_4\}$ as indicated by shaded off-diagonal elements of the information matrix. The robot makes three observations of active features, \mathbf{z}_1 , \mathbf{z}_2 , and \mathbf{z}_5 , and one of a passive feature, \mathbf{z}_3 . The first step of the ESEIF sparsification algorithm, as shown in the left-most diagram, is to update the posterior based upon a subset of the measurements, $\mathbf{z}_\alpha = \{\mathbf{z}_1, \mathbf{z}_3\}$, resulting in a stronger constraint with m_1 as well as the formation of a link with m_3 , as depicted in the middle figure. Sparsification then proceeds with the marginalization of the vehicle pose and subsequent relocation of the robot based upon the remaining measurements, \mathbf{z}_β . The implication on the information matrix is the connectivity of the initial set of active features and a desired restriction on the number of constraints with the vehicle pose.

this operation is easy in the standard form but difficult with the canonical parameterization where a large matrix inversion is necessary. Instead, Thrun *et al.* [15] first compute the conditional distribution for the Markov blanket for \mathbf{x}_i and \mathbf{x}_j , $p(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k | \mathbf{x}_l)$, which involves simply extracting a sub-block of the information matrix. They then invert this matrix and take the covariance sub-block corresponding to $p(\mathbf{x}_i, \mathbf{x}_j | \mathbf{x}_l)$ which they use for data association. While the authors have had success using this conditional covariance, it can be shown to yield overconfident estimates for the likelihood [4].

Alternatively, Eustice *et al.* [4] propose a method which solves for conservative estimates for the marginal covariance. The technique stems from posing the relationship, $\Lambda_t \Sigma_t = \mathbf{I}$, as a sparse system of linear equations, $\Lambda_t \Sigma_{*i} = \mathbf{e}_i$, where Σ_{*i} and \mathbf{e}_i denote the i^{th} columns of the covariance and identity matrices, respectively. To determine the robot pose covariance, the iterative algorithms previously presented for mean recovery can be used to solve the set of equations formed from the robot pose columns. Combining the estimate for robot pose covariance with a conservative estimate for the covariance of any map element gives rise to a joint covariance which is it-

self conservative. The joint covariance is then used to represent the marginal distribution for data association.

4 Results

To better understand the effectiveness of the two different sparse information filters, we compare the performance of ESEIFs and SEIFs to the standard EKF when applied to different forms of the SLAM problem. In the first case, we take a look at a controlled linear Gaussian simulation for which the KF, the optimal Bayesian estimator, is the “gold standard”. We then follow with experiments using real-world nonlinear datasets including a benchmark outdoor data set widely popular in the SLAM community.

4.1 Linear Gaussian Simulation

To systematically analyze the two information-based filters, we first apply the three estimators in a controlled simulation. The environment consists of a set of point features, uniformly distributed to achieve a desired density of 0.10 features per unit area. The vehicle moves translationally according to a linear, constant velocity motion model and, at any time step, is able to observe the relative position to a limited number of neighboring features. Both the vehicle motion as well as the measurements are corrupted by additive white Gaussian noise.

As a basis for comparison, we apply the Kalman Filter, the optimal estimator for linear Gaussian problems. The ESEIF and SEIF are implemented with a limit of $\Gamma_a = 10$ active features. When sparsifying the ESEIF, we reserve as many of the observations for relocating the robot as possible, to the extent that we do not violate the Γ_a bound.

In the LG case, sparse information filters have already been shown to be computationally efficient [15]. Instead, we are interested in evaluating the effect that the different sparsification strategies have on the estimation accuracy. To that end, we perform a series of Monte Carlo simulations, using the normalized estimation error squared (NEES) [1] to measure filter consistency using a pair of metrics. As one measure, we use the Euclidean distance between the state estimates and the ground truth which corresponds to the *global* error. To get a *local/relative* measure of error, we first reference the robot and map positions relative to the first observed feature, \mathbf{x}_m using the standard compounding operation, $\mathbf{x}_{mi} = \ominus \mathbf{x}_m \oplus \mathbf{x}_i$. We then compute the second error metric as the distance to the root-shifted representation of the ground truth. We plot the global normalized errors for the estimated vehicle position as well as for one of the map elements in Figures 3(a) and 3(b), respectively. Comparing these errors with the 97.5% chi-square upper bound indicated by the horizontal line, we see that the ESEIF yields consistent position estimates with errors similar to those of the KF. The normalized errors attributed to

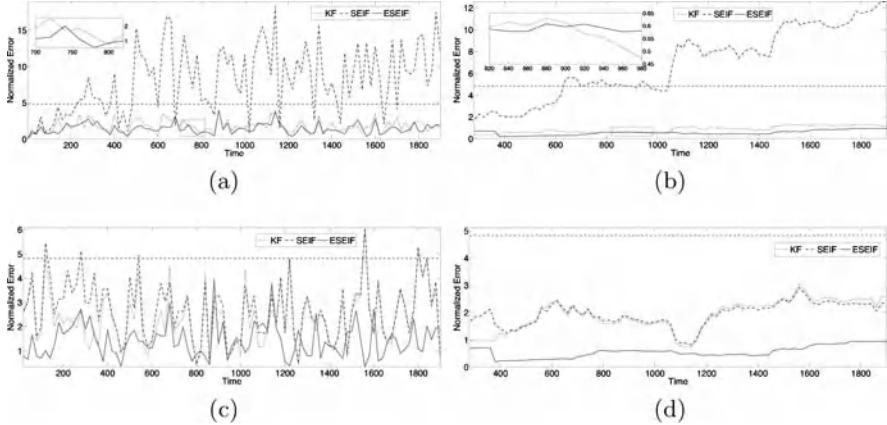


Fig. 3. The time history of the (a), (b) *global* and (c), (d) *local* normalized errors for the LG results, estimated from a series of Monte Carlo simulations. Plotted in (a) and (c) are the two errors for the vehicle. In (b) and (d) we show the errors for one of the features which is representative of the other elements in the map. The horizontal threshold denotes the 97.5% chi-square confidence bound. The local ESEIF and SEIF estimation errors are similar in magnitude to that of the Kalman Filter. The global error attributed to the SEIF, meanwhile, is noticeably larger, exceeding the chi-square bound. This indicates that the SEIF preserves local relationships but leads to estimates which are globally overconfident while the ESEIF maintains both global and local consistency.

the SEIF, on the other hand, are noticeably larger, frequently exceeding the chi-square bound. The local errors shown in Figures 3(c) and 3(d) are similar for all three filters, generally smaller than the confidence threshold. This behavior indicates that, in the linear Gaussian case, ESEIF's maintain a state estimate which is both globally and locally consistent while the SEIF leads to errors which are consistent locally but inconsistent in the absolute sense.

As a related consequence of the ESEIF sparsification strategy, the filter maintains conservative uncertainty estimates. In Figure 4(a) we compare the global map uncertainties for the two information filters to those of the Kalman Filter. In particular, from the inverse of the information matrices, we compute, for each feature, the log of the ratio of the covariance sub-block determinant to the determinant of the sub-block for the KF. Since the KF solution represents the true distribution, values larger than zero correspond to conservative estimates for a feature's position while values less than zero are a sign of overconfidence. As the histogram demonstrates, the ESEIF is conservative in its estimate for the absolute position of each feature while each of the marginals represented by the SEIF are overconfident. When we transform the maps relative to the first observed feature, we see in Figure 4(b) that the overconfidence of the SEIF is less severe while the ESEIF remains conservative. As a conse-

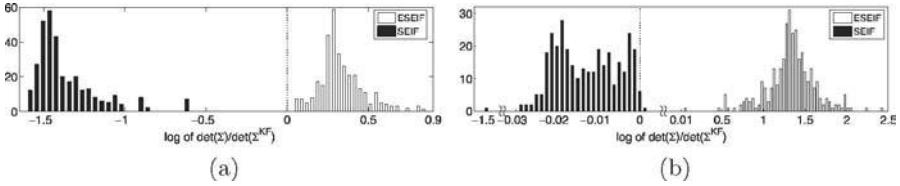


Fig. 4. The LG simulation estimates of map uncertainty maintained by the ESEIF and SEIF compared with that of the KF. For each feature, we consider the log of the ratio of the covariance sub-block determinant for the information filters to the determinant for the KF. Values equal to zero indicate an exact estimate for the uncertainty. Log ratios greater than zero imply conservative estimates while values less than zero correspond to overconfidence. In (a) we show a histogram describing the global measure of uncertainty determined directly from the inverse of the information matrices. The SEIF yields map estimates which are largely overconfident while the ESEIF leads to estimates which are conservative. Depicted in (b), the overconfidence of the SEIF is less severe when we consider the relative map uncertainty which follows from root-shifting the state to the first feature added to the map. The one outlier corresponds to the original world origin as represented in the new reference frame. Meanwhile, the histogram shows that the ESEIF maintains conservative estimates for the relative map covariance matrix.

quence of the overconfidence of its global map, the one exception in the case of the SEIF is the representation of the original world origin in the root-shifted reference frame.

4.2 Experimental Validation

The linear Gaussian simulations allow us systematically analyze the accuracy of the sparsified filters when we are able to perform inference on an exact model. Unfortunately, for most real-world applications, both the vehicle motion and observation models are nonlinear and are corrupted by noise which is not Gaussian. To demonstrate the application of ESEIFs to typical SLAM problems, we implement the algorithm along with the SEIF and the EKF on two nonlinear datasets.

For the first real-world application of SLAM, we consider the benchmark Victoria Park dataset, widely used as a testbed for SLAM algorithms. A truck equipped with dead-reckoning sensors and a laser scanner drives in a series of loops within Victoria Park, Sydney. Using a simple perceptual grouping implementation, we are able to detect tree trunks located throughout the park among the laser data which is cluttered with spurious returns. We solve the data association problem offline to ensure that the correspondences are the same for each filter.

We implement the ESEIF and SEIF estimators together with the EKF which has been successfully applied to this dataset in the past. We limit

the number of active features to a maximum number of $\Gamma_a = 10$ for the information filters. When we perform sparsification in the ESEIF, our priority is again on relocation in that we reserve as many tree observations as possible (i.e. no more than $\Gamma_a = 10$) for the purpose of adding the vehicle back into the map. Remaining measurements, if any, are used to update the ESEIF prior to marginalization. This helps to minimize the influence of spurious data on the relocated vehicle pose.

We plot the ESEIF and SEIF estimates of the map together with the three sigma uncertainty bounds in Figures 5(a) and 5(b), respectively. The estimates of the 3 km trajectory for the car are superimposed on the plot. As a basis for comparison, the plots include the feature locations resulting from the EKF which are nearly identical to those published elsewhere. Both sparsified filters yield similar maps though the deviation from the EKF estimates is noticeably larger for the SEIF than it is for the ESEIF. Furthermore, the global confidence bounds for the ESEIF are conservative, yet comparable to the feature uncertainties maintained by the EKF while they are significantly overconfident for the SEIF. While not ground truth, the EKF represents the baseline which the information filters strive to match and, yet, many of the EKF estimates lie outside the three sigma uncertainty bounds for the SEIF. This is especially evident in the periphery as we indicate in the inset plot. As we saw in the LG simulation, all three algorithms seem to equivalently represent the local map relationships given by the transformation of the map into the vehicle's reference frame at its final pose. Both the ESEIF relative map shown in Figure 5(c) and the SEIF relative map in Figure 5(d) are almost identical to the corresponding EKF results. In this case, the relative ESEIF and SEIF uncertainty bounds now capture the EKF estimate for the feature locations. The SEIF algorithm allows us to achieve results which are similar to the standard EKF in the local but not global sense while ESEIFs provide a conservative map estimate which is nearly identical to the EKF both globally *and* locally.

We have seen from the plots of the two SLAM maps that SEIFs are much more confident in their state estimates. In Figure 6(a) we compare the global uncertainty of each feature for the ESEIF and SEIF to the EKF, again using the log of the ratio of the determinant of the feature covariances. As with the linear Gaussian simulations, the ESEIF log ratios are all greater than zero, indicating that ESEIFs maintain conservative estimates for the global uncertainty of each state element. On the other hand, those of the SEIF are largely overconfident. Expressing the state in the vehicle reference frame, the histogram in Figure 6(b) reveals that the SEIF remains overconfident, although to a lesser extent. The one exception is again the representation of the global origin in the vehicle frame and is a direct consequence of the global inconsistency of SEIFs. The ESEIF, meanwhile, remains conservative in the relative frame.

In the second experiment, a wheeled robot drives around a gymnasium in which 64 track hurdles are positioned at known locations along the baselines of

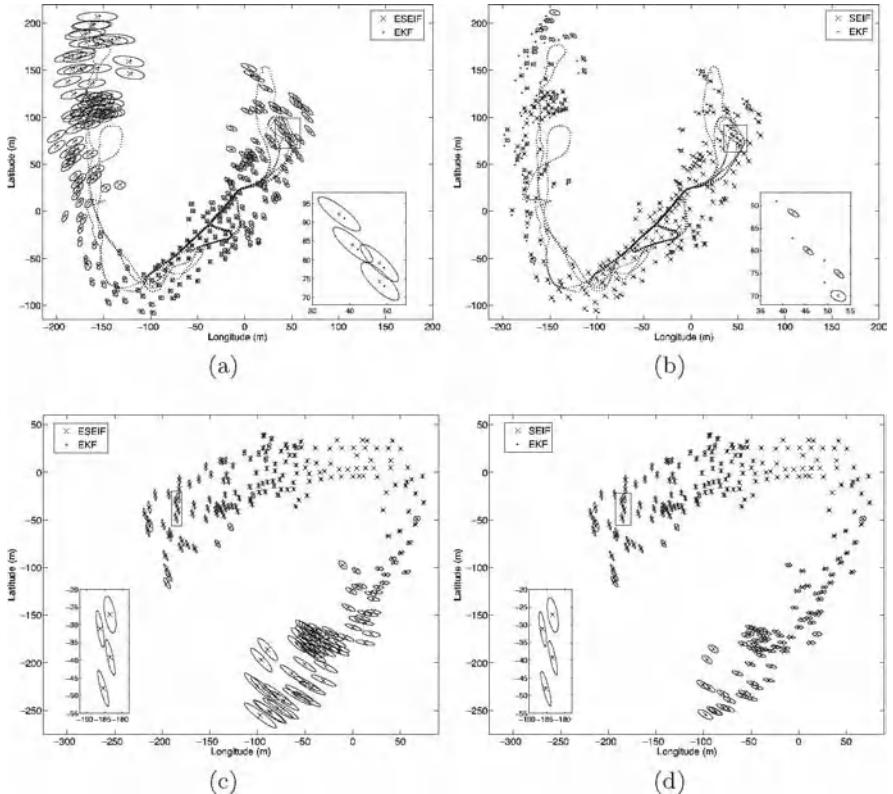


Fig. 5. Map and vehicle trajectory estimates for the Victoria Park dataset. In each, we include the final EKF map which agrees with previous results published in the literature. The top two plots represent the global state estimate while the two at the bottom are the result of root-shifting the map into the vehicle frame via compounding: $\mathbf{x}_{vi} = \ominus \mathbf{x}_v \oplus \mathbf{x}_i$. The plot in (a) presents the results of the ESEIF, including the three sigma confidence bounds for each of the features. The ESEIF produces feature estimates which are nearly identical to those of the EKF and, while it is omitted to make the plot readable, the uncertainty ellipses are very similar for the two filters. In (b), we see that while the SEIF and EKF maps are alike, the difference between the two estimates is noticeably larger for the SEIF algorithm. Additionally, the inset reveals that the SEIF yields global error estimates which are significantly overconfident. Looking at the maps expressed in the vehicle frame, though, we see that both (c) the ESEIF and (d) SEIF preserve the relative map structure.

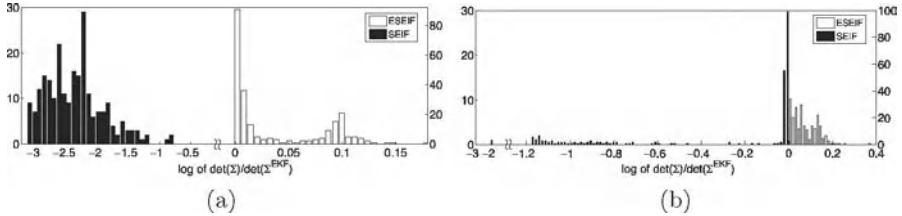


Fig. 6. Histograms of the ESEIF and SEIF uncertainty estimates as compared to the EKF results for the Victoria Park dataset. We again use the log of the ratio of the covariance sub-block determinants for each feature. The histogram in (a) corresponds to the direct filter estimates and is representative of the global uncertainty. The ESEIF maintains conservative estimates for the uncertainties while the SEIF estimates are overconfident when compared to the EKF. Expressing the map in the vehicle’s reference frame, (b) demonstrates that SEIFs remain overconfident but are better able to capture the relative uncertainty. Due to the global overconfidence, there is an outlier corresponding to the representation of the global origin in the robot’s frame. Meanwhile, the ESEIF local estimates remain conservative relative to the EKF.

four adjacent tennis courts. Wheel encoders provide the input to the kinematic motion model while observations of the environment are made using a SICK laser scanner. Data association is again performed offline and is the same for each filter.

We perform SLAM on the data again using both the ESEIF and SEIF alongside a standard EKF implementation. When necessary, we employ the two sparsification strategies to maintain a bound of $\Gamma_a = 10$ active features. During ESEIF sparsification, we relocate the robot using a single feature observation which provides a measurement of the relative transformation (translation and rotation) between the vehicle and the hurdle.

In Figure 7(a), we show the final map estimated by the ESEIF, overlayed onto a depiction of the ground truth. The ellipses drawn around each feature correspond to the three sigma bound on the position of one of the hurdle legs. The same plot is shown in Figure 7(b) for the map estimated using the SEIF algorithm. Notice that the uncertainty bounds maintained by the SEIF are significantly overconfident and, for many hurdles, do not include the true feature position. While we are able to maintain an estimate of the state which is both globally *and* locally conservative compared with that of the EKF using ESEIFs, enforcing sparsity in the SEIF results in an estimate which suffers from global inconsistency.

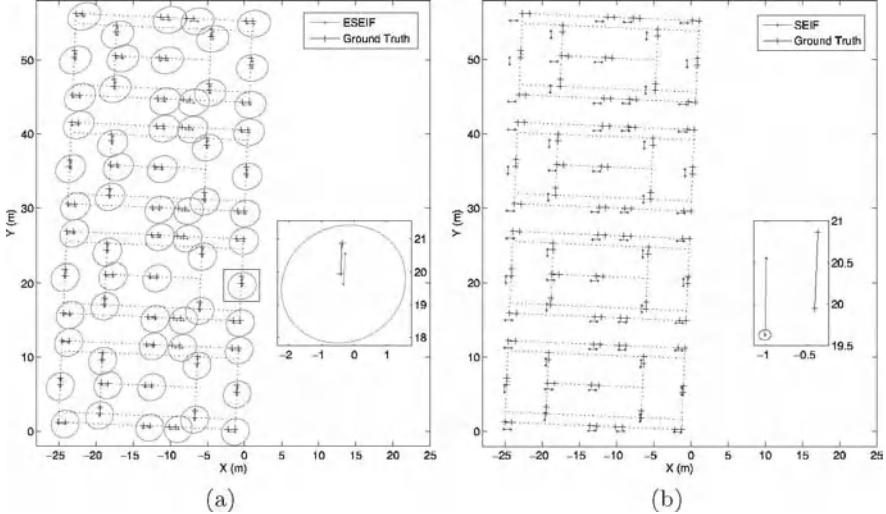


Fig. 7. The final maps generated by the (a) ESEIF and (b) SEIF algorithms. Included is an outline of the tennis courts overlayed with the ground truth hurdle poses indicated by the black cross hairs. The ellipses centered at the base leg of each feature represent the three sigma uncertainty bounds for position. Note the significant difference in magnitude between the confidence estimates maintained by the two filters. While the true feature locations are captured by the ESEIF uncertainty regions, a majority of the hurdles fall outside the SEIF ellipses. This overconfidence is a result of the approximation employed by SEIFs to enforce sparseness and is indicative of global inconsistency.

5 Discussion

We have shown both in simulation as well as with a pair of nonlinear datasets that the ESEIF maintains error measures which are both globally and locally conservative relative to the full Kalman estimates. In the linear Gaussian case, the implication is that the ESEIF sparsification strategy preserves consistency according to both metrics. On the other hand, as the ESEIF is formulated upon the dual of the EKF, it is subject to the same convergence issues that are attributed to the EKF for nonlinear applications [1]. As such, though the ESEIF error estimates are relatively conservative, this does not guarantee consistency in such cases. Nonetheless, the ESEIF algorithm is able to capitalize upon the computational benefits of the sparse information form without the cost of additional overconfidence. In this manner it provides an efficient means of achieving estimates nearly identical to those of the EKF which has been successfully applied in a number of real-world situations.

6 Conclusion

Of late, many researchers in the robotics community have been interested in developing solutions to the SLAM problem which scale with environments of arbitrary size. One approach that is particularly promising follows from the key insight that the information matrix is relatively sparse for feature-based SLAM. In the case where the matrix is exactly sparse, state estimation can be performed in near-constant time, irrespective of the number of landmarks in the environment.

While a majority of the elements in the information matrix are relatively weak, the matrix is naturally dense due to the effect of marginalizing out old robot poses. To achieve the efficiency benefits, the SEIF algorithm enforces sparsity by deliberately breaking weak links between the robot and the map. As a consequence of this pruning strategy, the SEIF state estimate suffers from global inconsistency.

In this paper, we have introduced an algorithm for feature-based SLAM which achieves an exactly sparse information matrix while maintaining global *and* local consistency, relative to the standard EKF. We have shown that, by periodically marginalizing out the robot and then relocalizing it within the map, we control the number of active landmarks and, in turn, the population of the information matrix. The ESEIF is then able to benefit from the efficiency of the sparse information form while yielding conservative estimates for the robot pose and map.

We have demonstrated the performance of ESEIFs, both in a systematic linear Gaussian simulation as well as on two different nonlinear datasets. In all three, we have shown that ESEIFs maintain estimates nearly identical to those of the EKF which, in comparison, are both globally *and* locally conservative.

Acknowledgements

This work was funded in part by the CenSSIS ERC of the NSF under grant EEC-9986821, by ONR under grants N00014-02-C-0210, N00014-05-1-0244, N00014-97-1-0202, and N00014-03-1-0879, and by the MIT Sea Grant College Program under grant NA86RG0074 (project RCM-3).

References

1. Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, 2001.
2. T. Duckett, S. Marshland, and J. Shapiro. Learning globally consistent maps by relaxation. In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3841–3846, San Francisco, USA, 2000.

3. R. Eustice, H. Singh, and J. Leonard. Exactly sparse delayed state filters. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2428–2435, Barcelona, Spain, April 2005.
4. R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard. Visually navigating the RMS Titanic with SLAM information filters. In *Proceedings of Robotics: Science and Systems (RSS)*, Cambridge, MA, USA, June 2005.
5. R. Eustice, M. Walter, and J. Leonard. Sparse extended information filters: Insights into sparsification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Alberta, Canada, August 2005.
6. U. Frese. A proof for the approximate sparsity of SLAM information matrices. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 331–337, Barcelona, Spain, April 2005.
7. U. Frese. Treemap: An $\mathcal{O}(\log n)$ algorithm for simultaneous localization and mapping. In C. Freksa, editor, *Spatial Cognition IV*, pages 455–476. Springer Verlag, 2005.
8. U. Frese and G. Hirzinger. Simultaneous localization and mapping - a discussion. In *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, pages 17–26, 2001.
9. U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localization and mapping. *IEEE Transactions on Robotics*, 21(2):196–207, April 2005.
10. Y. Liu and S. Thrun. Results for outdoor-SLAM using sparse extended information filters. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1227–1233, Taipei, Taiwan, 2003.
11. Peter S. Maybeck. *Stochastic Models, Estimation, and Control, Volume 1*. Academic Press, New York, NY, 1979.
12. M. Paskin. Thin junction tree filters for simultaneous localization and mapping. Technical Report CSD-02-1198, University of California, Berkeley, September 2002.
13. Jonathan Shewchuck. An introduction to the conjugate gradient method without the agonizing pain. Technical Report CMU-CS-94-125, Carnegie Mellon University, August 1994.
14. R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer Verlag, 1990.
15. S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7-8):693–716, July-August 2004.

Field Robots

Session Overview

Field Robotics

Alonzo Kelly and Chuck Thorpe

Carnegie Mellon University {alonzo,cet}@ri.cmu.edu

Field robots do not operate in factories or other controlled settings, but rather operate outdoors, underwater, underground, or even on other planets. They are characterized by a focus on real applications, and on operation in complex terrain. Field robots are often large vehicles, and often have forceful interactions with their workspace. Given their complex setting and complex (and often dangerous) tasks, most field robots are not fully autonomous: a great deal of effort goes into the user interface, providing mixed modes of human and robot interaction.

Field Robotics is a branch of robotics characterized by its domain: the applications of robotics in the unstructured world to perform useful tasks. The papers in this session illustrate well the breadth of concerns addressed in building field robots. Some of earliest field robots were configured only for mobility and data gathering to perform such missions as exploration and mapping. Today, systems have been fielded which interact forcefully with the environment in such applications as excavation, mining and sampling. Many field robots are characterized by large scales (big machines, long distances covered); forceful interaction (either with large loads or with difficult terrain); complex machines (robots with many degrees of freedom); and complex environments (moving objects, soft and uneven terrain); and difficult operating environments (limited bandwidth, large distances between operator and machine).

Many of the applications that field robotics aspires to automate take place outdoors, in fairly unstructured environments, because we would like to give our robots the worst jobs; those that are difficult, dirty, and dangerous. Outdoor environments are complex due to their lack of predictable structure, uncontrollable weather conditions, and the pervasiveness of hazards. Often, the surface over which the robot moves is soft, or uneven, or difficult to sense. Such complexity often leads us to either choose more benign environments or to reduce the level of autonomy and involve humans: mixed-mode control, with varying degrees of control shared between a human and a robot, is an active area of field robotics research. Nonetheless, there are situations where

higher levels of autonomy can still be argued to be prudent. In particular, being productive outdoors is a dangerous business, for robots as for humans. This session includes three papers which exploit autonomy to address extremes, respectively, in environmental complexity, remoteness, and danger to humans.

Some rich environments are characterized by complicated topology and many spatially-distributed degrees of mobility hazard. In these environments, perception must often make up for characteristically inadequate prior information. Yet, perception is only half the problem. Once something unexpected is perceived, a new mobility plan must be generated, and it must be generated in real-time if the vehicle is to move continuously during the process. The first paper *Field D*: An Interpolation Based Path Planner and Replanner* describes a version of the D* real-time replanning algorithm which is designed for such environments. *Field Dstar* uses interpolation to remove the discretized heading constraint under which most discrete motion planners operate. This leads to smoother plans which can be superior to those generated by optimal discrete planners.

Due to the well-known difficulties of teleoperation, many situations persist where autonomy is the only effective option. Extraterrestrial environments are so extremely remote that even the speed of light is a limitation. The Mars Exploration Rovers named Spirit and Opportunity have recently achieved a landmark in field robotics history. Kilometers of terrain on another planet have now been successfully traversed under autonomous control. The second paper in this session *Tradeoffs Between Directed and Autonomous Driving on The MER Rovers* describes the issues associated with controlling the rovers and the graduated autonomy levels that arose to address them. The logistics of communicating only twice daily with the rovers, combined with the need to move quickly to the next science target, leads to the judicious use of autonomy in order to optimize productivity while managing risk to the rover.

Many terrestrial applications also present a plain tradeoff between risk and productivity and robotics can, of course, be used to redefine that trade-off. Among commercial applications, mining is well-known to challenge our capacity to remove risk while simultaneously addressing the need to get a job done quickly and well. The third paper in this session is *Surface Mining: Challenges and Main Research Issues for Autonomous Operations*. It surveys the reasons for our present successes in mining automation as well as the remaining challenges to be addressed in order to increase the impact of field robots on mining in the future.

Field D*: An Interpolation-Based Path Planner and Replanner

Dave Ferguson and Anthony Stentz

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania
`{dif,tony}@cmu.edu`

Summary. We present an interpolation-based planning and replanning algorithm for generating direct, low-cost paths through nonuniform cost grids. Most grid-based path planners use discrete state transitions that artificially constrain an agent's motion to a small set of possible headings (e.g. $0, \frac{\pi}{4}, \frac{\pi}{2}$, etc). As a result, even 'optimal' grid-based planners produce unnatural, suboptimal paths. Our approach uses linear interpolation during planning to calculate accurate path cost estimates for arbitrary positions within each grid cell and to produce paths with a range of continuous headings. Consequently, it is particularly well suited to planning low-cost trajectories for mobile robots. In this paper, we introduce the algorithm and present a number of example applications and results.

1 Introduction

In mobile robot navigation, we are often provided with a grid-based representation of our environment and tasked with planning a path from some initial robot location to a desired goal location. Depending on the environment, the representation may be binary (each grid cell contains either an obstacle or free space) or may associate with each cell a cost reflecting the difficulty of traversing the respective area of the environment.

In robotics, it is common to improve efficiency by approximating this grid with a graph, where nodes are placed at the center of each grid cell and edges connect nodes within adjacent grid cells. Many algorithms exist for planning paths over such graphs. Dijkstra's algorithm computes paths from every node to a specified goal node [3]. A* uses a heuristic to focus the search from a particular start location towards the goal and thus produces a path from a single location to the goal very efficiently [5, 18]. D*, Incremental A*, and D* Lite are extensions of A* that incrementally repair solution paths when changes occur in the underlying graph [26, 7, 8, 9]. These incremental algorithms have been used extensively in robotics for mobile robot navigation in unknown or dynamic environments.



Fig. 1. Some robots that currently use Field D* for global path planning. These range from indoor planar robots (the Pioneers) to outdoor robots able to operate in harsh terrain (the XUV).

However, almost all of these approaches are limited by the small, discrete set of possible transitions they allow from each node in the graph. For instance, given a graph extracted from a uniform resolution 2D grid, a path planned in the manner described above restricts the agent’s heading to increments of $\frac{\pi}{4}$. This results in paths that are suboptimal in length and difficult to traverse in practice. Further, even when these paths are used in conjunction with a local arc-based planner (e.g. as in the RANGER system [6, 25]), they can still cause the vehicle to execute expensive trajectories involving unnecessary turning.

In this paper we present Field D*, an interpolation-based planning and replanning algorithm that alleviates this problem. This algorithm extends D* and D* Lite to use linear interpolation to efficiently produce low-cost paths that eliminate unnecessary turning. The paths are optimal given a linear interpolation assumption and very effective in practice. This algorithm is currently being used by a wide range of fielded robotic systems (see Figure 1).

We begin by discussing the limitations of paths produced using classical grid-based planners and mention recent approaches that attempt to overcome some of these limitations. We then present an interpolation-based method for obtaining more accurate path cost approximations and show how this method can be incorporated into existing planning and replanning algorithms. We provide a number of example illustrations and applications of our approach and conclude with discussion and extensions.

2 Limitations of Classical 2D Path Planning

Consider a robotic ground vehicle navigating an outdoor environment. We can represent this environment as a uniform resolution 2D traversability grid, in which cells are given a cost per unit of traverse (traversal cost) reflecting the difficulty of navigating the respective area of the environment. If this traversability grid encodes the configuration space costs (i.e. the traversal costs have been expanded to reflect the physical dimensions of the vehicle), then planning a path for the robot translates to generating a trajectory through this grid for a single point. A common approach used in robotics for performing this planning is to combine an approximate *global* planner with an accurate

local planner [6, 25, 1, 23]. The global planner computes paths through the grid that ignore the kinematic and dynamic constraints of the vehicle. Then, the local planner takes into account the constraints of the vehicle and generates a set of feasible local trajectories that can be taken from its current position. To decide which of these trajectories to execute, the robot evaluates both the cost of each local trajectory and the cost of a global path from the end of each trajectory to the robot's desired goal location.

To formalize the global planning task, we need to define more precisely some concepts already introduced. First, each cell in the grid has assigned to it some real-valued traversal cost that is greater than zero. The cost of a line segment between two points within a cell is the Euclidean distance between the points multiplied by the traversal cost of the cell. The cost of any path within the grid is the sum of the costs of its line segments through each cell. Then, the global planning task (involving a uniform resolution grid) can be specified as follows.

The Global Planning Task: *Given a region in the plane partitioned into a uniform grid of square cells \mathcal{T} , an assignment of traversal costs $c : \mathcal{T} \rightarrow (0, +\infty]$ to each cell, and two points s_{start} and s_{goal} within the grid, find the path within the grid from s_{start} to s_{goal} with minimum cost.*

This task can be seen as a specific instance of the Weighted Region Problem [17], where the regions are uniform square tiles. A number of algorithms exist to solve this problem in the computational geometry literature (see [16] for a good survey). In particular, [17] and [21] present approaches based on Snell's law of refraction that compute optimal paths by simulating a series of light rays that propagate out from the start position and refract according to the different traversal costs of the regions encountered. These approaches are efficient for planning through environments containing a small number of homogenous-cost regions, but are computationally expensive when the number of such regions is very large, as in the case of a uniform grid with varying cell costs.

Because of the computational expense associated with planning optimal paths through grids, researchers in robotics have focussed on basic approximation algorithms that are extremely fast. The most popular such approach is to approximate the traversability grid as a discrete graph, then generate paths over the graph. A common way to do this is to assign a node to each cell center, with edges connecting the node to each adjacent cell center (node). The cost of each edge is a combination of the traversal costs of the two cells it transitions through and the length of the edge. Figure 2(a) shows this node and edge extraction process for one cell in a uniform resolution 2D grid.

We can then plan over this graph to generate paths from the robot's initial location to a desired goal location. As mentioned previously, a number of efficient algorithms exist for performing this planning, such as A* for initial planning and D* and its variants for replanning [5, 18, 26, 8]. Unfortunately, paths produced using this graph are restricted to headings of $\frac{\pi}{4}$ increments.

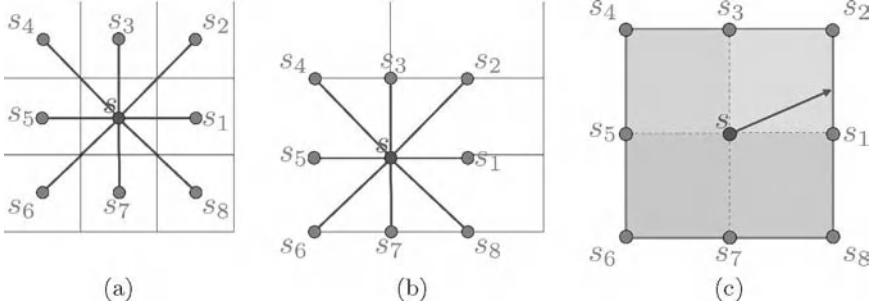


Fig. 2. (a) A standard 2D grid used for global path planning in which nodes reside at the centers of the grid cells. The arcs emanating from the center node represent all the possible actions that can be taken from this node. (b) A modified representation used by Field D*, in which nodes reside at the corners of grid cells. (c) The optimal path from node s must intersect one of the edges $\{\overrightarrow{s_1s_2}, \overrightarrow{s_2s_3}, \overrightarrow{s_3s_4}, \overrightarrow{s_4s_5}, \overrightarrow{s_5s_6}, \overrightarrow{s_6s_7}, \overrightarrow{s_7s_8}, \overrightarrow{s_8s_1}\}$.

This means that the final solution path may be suboptimal in path cost, involve unnecessary turning, or both.

For instance, consider a robot facing its goal position in a completely obstacle-free environment (see Figure 3). Obviously, the optimal path is a straight line between the robot and the goal. However, if the robot's initial heading is not a multiple of $\frac{\pi}{4}$, traditional grid-based planners would return a path that has the robot first turn to attain the nearest grid heading, move some distance along this heading, and then turn $\frac{\pi}{4}$ in the opposite direction of its initial turn and continue to the goal. Not only does this path have clearly suboptimal length, it contains possibly expensive or difficult turns that are purely artifacts of the limited representation. Such global paths, when coupled with the results of a local planner, cause the robot to behave suboptimally. Further, this limitation of traditional grid-based planners is not alleviated by increasing the resolution of the grid.

Sometimes it is possible to reduce the severity of this problem by post-processing the path. Usually, given a robot location s , one finds the furthest point p along the solution path for which a straight line path from s to p is collision-free, then replaces the original path to p with this straight line path. However, this does not always work, as illustrated by Figure 4. Indeed, for nonuniform cost environments such post-processing can often *increase* the cost of the path.

A more comprehensive post-processing approach is to take the result of the global planner and use it to seed a higher dimensional planner that incorporates the kinematic or dynamic constraints of the robot. Stachniss and Burgard [24] present an approach that takes the solution generated by the global planner and uses it to extract a local waypoint to use as the goal for a 5D trajectory planner. The search space of the 5D planner is limited to a small

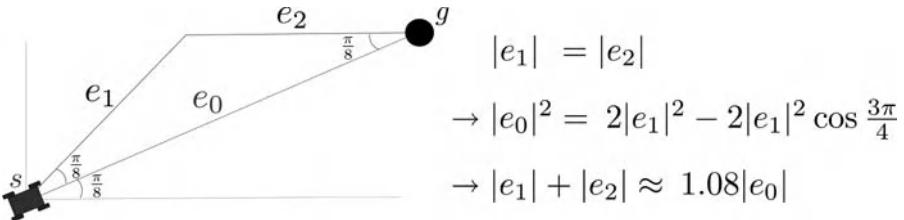


Fig. 3. A uniform resolution 2D grid-based path (e_1 plus e_2) between two grid nodes can be up to 8% longer than an optimal straight-line path (e_0). Here, the desired straight-line heading is $\frac{\pi}{8}$ and lies perfectly between the two nearest grid-based headings of 0 and $\frac{\pi}{4}$. This result is independent of the resolution of the grid.

area surrounding the global solution path. Likhachev et al. [15, 14] present an approach that uses the cost-to-goal value function of the global planner to focus an anytime global 4D trajectory planner. Their approach improves the quality of the global trajectory while deliberation time allows. However, these higher dimensional approaches can be much more computationally expensive than standard grid-based planners and are still influenced by the results of the initial grid-based solution.

Recently, robotics researchers have looked at more sophisticated methods of obtaining better paths through grids without sacrificing too much of the efficiency of the classic grid-based approach described above. Konolige [10] presents an interpolated planner that first uses classic grid-based planning to construct a cost-to-goal value function over the grid and then interpolates this result to produce a shorter path from the initial position to the goal. This method results in shorter, less-costly paths for agents to traverse but does not incorporate the reduced path cost into the planning process. Consequently, the resulting path is not necessarily as good as the path the algorithm would produce if interpolated costs were calculated during planning. Further, if we are computing paths from several locations (which is common when combining the global planner with a local planner) then this post-processing interpolation step can be expensive. Also, this approach provides no replanning functionality to update the solution when new information concerning the environment is received.

Philipsen and Siegwart [20] present an algorithm based on Fast Marching Methods [22] that computes a value function over the grid by growing a surface out from the goal to every region in the environment. The surface expands according to surface flow equations, and the value of each grid point is computed by combining the values of two neighboring grid points. This approach incorporates the interpolation step into the planning process, producing low-cost, interpolated paths. This technique has been shown to generate nice paths in indoor environments [19, 20]. However, the search is not focussed towards the robot location (such as in A*) and assumes that the transition cost from a particular grid node to each of its neighbors is constant. Consequently, it is

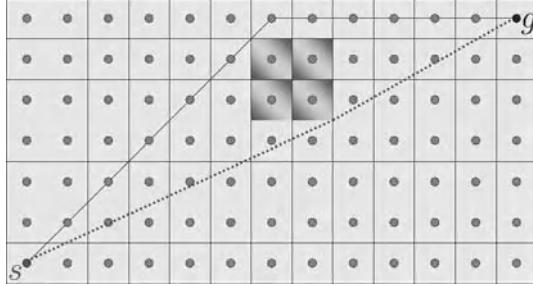


Fig. 4. 2D grid-based paths cannot always be shortened in a post-processing phase. Here, the grid-based path from s to g (top, in black) cannot be shortened because there are four obstacle cells (shaded). The optimal path is shown dashed.

not as applicable to navigation in outdoor environments, which are often best represented by large grids with widely-varying cell traversal costs.

The idea of using interpolation to produce better value functions for discrete samples over a continuous state space is not new. This approach has been used in dynamic programming for some time to compute the value of successors that are not in the set of samples [11, 12, 13]. However, as LaValle points out [13], this becomes difficult when the action space is also continuous, as solving for the value of a state now requires minimizing over an infinite set of successor states.

The approach we present here is an extension of the widely-used D* family of algorithms that uses linear interpolation to produce near-optimal paths that eliminate unnecessary turning. It relies upon an efficient, closed-form solution to the above minimization problem for 2D grids, which we introduce in the next section. This method produces much straighter, less-costly paths than classical grid-based planners without sacrificing real-time performance. As with D* and D* Lite, our approach focusses its search towards the most relevant areas of the state space during both initial planning and replanning. Further, it takes into account local variations in cell traversal costs and produces paths that are optimal given a linear interpolation assumption. As the resolution of the grid increases, the solutions returned by the algorithm improve, approaching true optimal paths.

3 Improving Cost Estimation Through Interpolation

The key to our algorithm is a novel method for computing the path cost of each grid node s given the path costs of its neighboring nodes. By the path cost of a node we mean the cost of the cheapest path from the node to the goal. In classical grid-based planning this value is computed as

$$g(s) = \min_{s' \in nbrs(s)} (c(s, s') + g(s')), \quad (1)$$

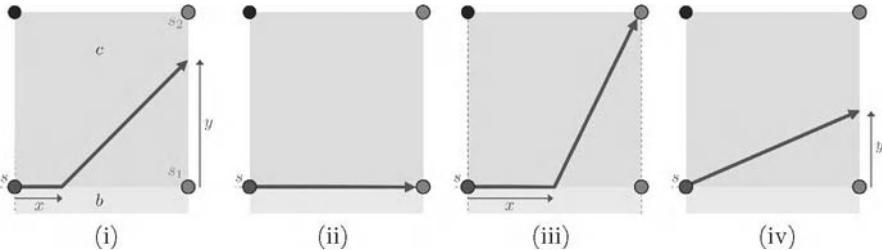


Fig. 5. Computing the path cost of node s using the path cost of two of its neighbors, s_1 and s_2 , and the traversal costs c of the center cell and b of the bottom cell. Illustrations (ii) through (iv) show the possible optimal paths from s to edge $\overrightarrow{s_1s_2}$.

where $nbrs(s)$ is the set of all neighboring nodes of s (see Figure 2), $c(s, s')$ is the cost of traversing the edge between s and s' , and $g(s')$ is the path cost of node s' .

This calculation assumes that the only transitions possible from node s are straight-line trajectories to one of its neighboring nodes. This assumption results in the limitations of grid-based plans discussed earlier. However, consider relaxing this assumption and allowing a straight-line trajectory from node s to any point on the boundary of its grid cell. If we knew the value of every point s_b along this boundary, then we could compute the optimal value of node s simply by minimizing $c(s, s_b) + g(s_b)$, where $c(s, s_b)$ is computed as the distance between s and s_b multiplied by the traversal cost of the cell in which s resides. Unfortunately, there are an infinite number of such points s_b and so computing $g(s_b)$ for each of them is not possible.

It is possible, however, to provide an approximation to $g(s_b)$ for each boundary point s_b by using linear interpolation. To do this, we first modify the graph extraction process discussed earlier. Instead of assigning nodes to the centers of grid cells, we assign nodes to the *corners* of each grid cell, with edges connecting nodes that reside at corners of the same grid cell (see Figure 2(b)).

Given this modification, the traversal costs of any two equal-length segments of an edge will be the same. This differs from the original graph extraction process in which the first half of an edge was in one cell and the second half was in another cell, with the two cells possibly having different traversal costs. In the modified approach the cost of an edge that resides on the boundary of two grid cells is defined as the minimum of the traversal costs of each of the two cells.

We then treat the nodes in our graph as sample points of a continuous cost field. The optimal path from a node s must pass through an edge connecting two consecutive neighbors of s , for example $\overrightarrow{s_1s_2}$ (see Figure 2(c)). The path cost of s is thus set to the minimum cost of a path through any of these edges, which are considered one at a time. To compute the path cost of node s using

ComputeCost(s, s_a, s_b)

```

01. if ( $s_a$  is a diagonal neighbor of  $s$ )
02.    $s_1 = s_b; s_2 = s_a;$ 
03. else
04.    $s_1 = s_a; s_2 = s_b;$ 
05.    $c$  is traversal cost of cell with corners  $s, s_1, s_2$ ;
06.    $b$  is traversal cost of cell with corners  $s, s_1$  but not  $s_2$ ;
07.   if ( $\min(c, b) = \infty$ )
08.      $v_s = \infty;$ 
09.   else if ( $g(s_1) \leq g(s_2)$ )
10.      $v_s = \min(c, b) + g(s_1);$ 
11.   else
12.      $f = g(s_1) - g(s_2);$ 
13.     if ( $f \leq b$ )
14.       if ( $c \leq f$ )
15.          $v_s = c\sqrt{2} + g(s_2);$ 
16.       else
17.          $y = \min(\frac{f}{\sqrt{c^2-f^2}}, 1);$ 
18.          $v_s = c\sqrt{1+y^2} + f(1-y) + g(s_2);$ 
19.     else
20.       if ( $c \leq b$ )
21.          $v_s = c\sqrt{2} + g(s_2);$ 
22.       else
23.          $x = 1 - \min(\frac{b}{\sqrt{c^2-b^2}}, 1);$ 
24.          $v_s = c\sqrt{1+(1-x)^2} + bx + g(s_2);$ 
25. return  $v_s;$ 
```

Fig. 6. The Interpolation-based Path Cost Calculation

edge $\overrightarrow{s_1 s_2}$, we use the path costs of nodes s_1 and s_2 and the traversal costs c of the center cell and b of the bottom cell (see Figure 5).

To compute this cost efficiently, we assume the path cost of any point s_y residing on the edge between s_1 and s_2 is a linear combination of $g(s_1)$ and $g(s_2)$:

$$g(s_y) = yg(s_2) + (1 - y)g(s_1), \quad (2)$$

where y is the distance from s_1 to s_y (assuming unit cells). This assumption is not perfect: the path cost of s_y may not be a *linear* combination of $g(s_1)$ and $g(s_2)$, nor even a function of these path costs. However, this linear approximation works well in practice, and allows us to construct a closed form solution for the path cost of node s .

Given this approximation, the path cost of s given s_1, s_2 , and cell costs c and b can be computed as

$$\min_{x,y} [bx + c\sqrt{(1-x)^2 + y^2} + yg(s_2) + (1 - y)g(s_1)], \quad (3)$$

where $x \in [0, 1]$ is the distance traveled along the bottom edge from s before cutting across the center cell to reach the right edge a distance of $y \in [0, 1]$ from s_1 (see Figure 5(i)). Note that if both x and y are zero in the above equation the path taken is along the bottom edge but its cost is computed from the traversal cost of the center cell.

Let (x^*, y^*) be a pair of values for x and y that solve the above minimization. Because of our use of linear interpolation, at least one of these values

will be either zero or one. We formally prove this in an extended technical report version of this paper [4]. Intuitively, if it is less expensive to partially cut through the center cell than to traverse around the boundary, then it is least expensive to completely cut through the cell. Thus, if there is any component to the cheapest solution path from s that cuts through the center cell, it will be as large as possible, forcing $x^* = 0$ or $y^* = 1$. If there is no component of the path that cuts through the center cell, then $y^* = 0$.

Thus, the path will either travel along the entire bottom edge to s_1 (Figure 5(ii)), or will travel a distance x along the bottom edge then take a straight-line path directly to s_2 (Figure 5(iii)), or will take a straight-line path from s to some point s_y on the right edge (Figure 5(iv)). Which of these paths is cheapest depends on the relative sizes of c , b , and the difference f in path cost between s_1 and s_2 : $f = g(s_1) - g(s_2)$. Specifically, if $f < 0$ then the optimal path from s travels straight to s_1 and will have a cost of $(\min(c, b) + g(s_1))$ (Figure 5(ii)). If $f = b$ then the cost of a path using some portion of the bottom edge (Figure 5(iii)) will be equivalent to the cost of a path using none of the bottom edge (Figure 5(iv)). We can solve for the value of y that minimizes the cost of the latter path as follows.

First, let $k = f = b$. The cost of a path from s through edge $\overrightarrow{s_1 s_2}$ is

$$c\sqrt{1+y^2} + k(1-y) + g(s_2). \quad (4)$$

Taking the derivative of this cost with respect to y and setting it equal to zero yields

$$y^* = \sqrt{\frac{k^2}{c^2 - k^2}}. \quad (5)$$

Whether the bottom edge or the right edge is used, we end up with the same calculations and path cost computations. So all that matters is which edge is cheaper. If $f < b$ then we use the right edge and compute the path cost as above (with $k = f$), and if $b < f$ we use the bottom edge and substitute $k = b$ and $y^* = 1 - x^*$ into the above equation. The resulting algorithm for computing the minimum-cost path from s through an edge between *any* two consecutive neighbors s_a and s_b is provided in Figure 6. Given the minimum-cost paths from s through each of its 8 neighboring edges, we can compute the path cost for s to be the cost of the cheapest of these paths. The corresponding path is optimal given our linear interpolation assumption.

4 Field D*

Once equipped with this interpolation-based path cost calculation for a given node in our graph, we can plug it into any of a number of current planning and replanning algorithms to produce low-cost paths. Figure 7 presents our simplest formulation of *Field D**, an incremental replanning algorithm that

```

key( $s$ )
 01. return [ $\min(g(s), \text{rhs}(s)) + h(s_{\text{start}}, s); \min(g(s), \text{rhs}(s))$ ];

UpdateState( $s$ )
 02. if  $s$  was not visited before,  $g(s) = \infty$ ;
 03. if ( $s \neq s_{\text{goal}}$ )
 04.    $\text{rhs}(s) = \min_{(s', s'') \in \text{connbrs}(s)} \text{ComputeCost}(s, s', s'')$ ;
 05. if ( $s \in \text{OPEN}$ ) remove  $s$  from  $\text{OPEN}$ ;
 06. if ( $g(s) \neq \text{rhs}(s)$ ) insert  $s$  into  $\text{OPEN}$  with  $\text{key}(s)$ ;

ComputeShortestPath()
 07. while ( $\min_{s \in \text{OPEN}}(\text{key}(s)) \prec \text{key}(s_{\text{start}})$  OR  $\text{rhs}(s_{\text{start}}) \neq g(s_{\text{start}})$ )
 08.   remove state  $s$  with the minimum key from  $\text{OPEN}$ ;
 09.   if ( $g(s) > \text{rhs}(s)$ )
 10.      $g(s) = \text{rhs}(s)$ ;
 11.     for all  $s' \in \text{nbrs}(s)$  UpdateState( $s'$ );
 12.   else
 13.      $g(s) = \infty$ ;
 14.     for all  $s' \in \text{nbrs}(s) \cup \{s\}$  UpdateState( $s'$ );

Main()
 15.  $g(s_{\text{start}}) = \text{rhs}(s_{\text{start}}) = \infty; g(s_{\text{goal}}) = \infty$ ;
 16.  $\text{rhs}(s_{\text{goal}}) = 0; \text{OPEN} = \emptyset$ ;
 17. insert  $s_{\text{goal}}$  into  $\text{OPEN}$  with  $\text{key}(s_{\text{goal}})$ ;
 18. forever
 19.   ComputeShortestPath();
 20.   Wait for changes in cell traversal costs;
 21.   for all cells  $x$  with new traversal costs
 22.     for each state  $s$  on a corner of  $x$ 
 23.       UpdateState( $s$ );

```

Fig. 7. The Field D* Algorithm (basic D* Lite version).

incorporates these interpolated path costs. This version of Field D* is based on D* Lite¹.

In this figure, $\text{connbrs}(s)$ contains the set of consecutive neighbor pairs of node s : $\text{connbrs}(s) = \{(s_1, s_2), (s_2, s_3), (s_3, s_4), (s_4, s_5), (s_5, s_6), (s_6, s_7), (s_7, s_8), (s_8, s_1)\}$, where s_i is positioned as shown in Figure 2(c). Apart from this construction, notation follows the D* Lite algorithm: $g(s)$ is the current path cost of node s (its g -value), $\text{rhs}(s)$ is the one-step lookahead path cost for s (its rhs -value), OPEN is a priority queue containing inconsistent nodes (i.e., nodes s for which $g(s) \neq \text{rhs}(s)$) in increasing order of key values (line 1), s_{start} is the initial agent node, and s_{goal} is the goal node. $h(s_{\text{start}}, s)$ is a heuristic estimate of the cost of a path from s_{start} to s . Because the key value of each node contains two quantities a lexicographic ordering is used: $\text{key}(s) \prec \text{key}(s')$ iff the first element of $\text{key}(s)$ is less than the first element of $\text{key}(s')$ or the first element of $\text{key}(s)$ equals the first element of $\text{key}(s')$ and

¹ Differences between Field D* and D* Lite appear on lines 4 and 20 through 23.

As opposed to the original, graph-based version of D* Lite, lines 20 - 22 tailor Field D* to grids. Also, because paths intersect edges and not just nodes, the heuristic value $h(s_{\text{start}}, s)$ must be small enough that when added to the cost of any edge incident on s it is still not greater than a minimum cost path from s_{start} to s .

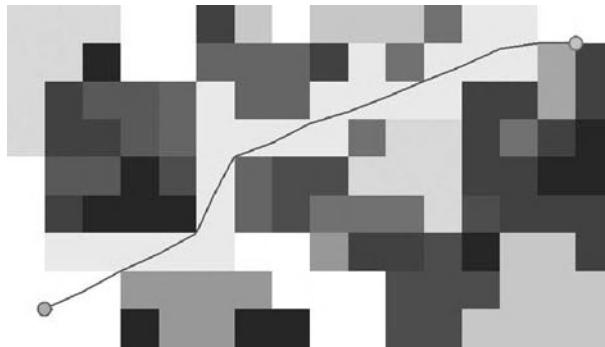


Fig. 8. A close-up of a path planned using Field D* showing individual grid cells. Notice that the path is not limited to entering and exiting cells at corner points.

the second element of $\text{key}(s)$ is less than the second element of $\text{key}(s')$. For more details on the D* Lite algorithm and this terminology, see [8, 7]. Also, the termination and correctness of the Field D* algorithm follow directly from D* Lite and the analysis of the cost calculation provided in Section 3.

This is an unoptimized version of Field D*. In our extended technical report [4] we discuss a number of optimizations that significantly improve the overall efficiency of planning and replanning with this algorithm.

Once the cost of a path from the initial state to the goal has been calculated, the path is extracted by starting at the initial position and iteratively computing the cell boundary point to move to next. Because of our interpolation technique, it is possible to compute the path cost of *any* point inside a grid cell, not just the corners, which is useful for both extracting the path and getting back on track if execution is not perfect (which is usually the case for real robots).

Figures 8 and 9 illustrate paths produced by Field D* through three nonuniform cost environments. In each of these figures, darker areas represent regions that are more costly to traverse. Notice that, unlike paths produced using classical grid-based planners, the paths produced using Field D* are not restricted to a small set of headings. As a result, Field D* provides lower-cost paths through both uniform and nonuniform cost environments.

5 Results

The true test of an algorithm is its practical effectiveness. We have found Field D* to be extremely useful for a wide range of robotic systems navigating through terrain of varying degrees of difficulty (see Figure 1).

To provide a quantitative comparison of the performance of Field D* relative to D* Lite, we ran a number of replanning simulations in which we measured both the relative solution path costs and runtimes of the opti-

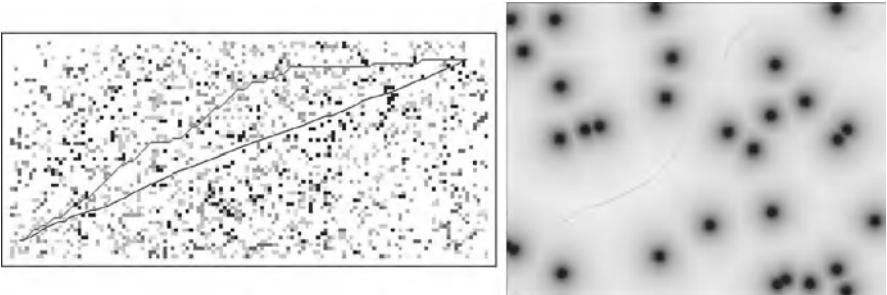


Fig. 9. (left) Paths produced by D* Lite (top) and Field D* (bottom) in a 150×60 nonuniform cost environment. **(right)** Field D* planning through a potential field of obstacles.

mized versions of the two approaches. We generated 100 different 1000×1000 nonuniform cost grid environments in which each grid cell was assigned an integer traversal cost between 1 (free space) and 16 (obstacle). With probability 0.5 this cost was set to 1, otherwise it was randomly selected. For each environment, the initial task was to plan a path from the lower left corner to a randomly selected goal on the right edge. After this initial path was planned, we randomly altered the traversal costs of cells close to the agent (10% of the cells in the environment were changed) and had each approach repair its solution path. This represents a significant change in the information held by the agent and results in a large amount of replanning.

During initial planning, Field D* generated solutions that were on average 96% as costly as those generated by D* Lite, and took 1.7 times as long to generate these solutions. During replanning, the results were similar: Field D* provided solutions on average 96% as costly and took 1.8 times as long. The average initial planning runtime for Field D* on a 1.5 GHz Powerbook G4 was 1.5s, and the average replanning runtime was 0.07s. In practice, the algorithm is able to provide real-time performance for fielded systems.

6 Discussion

Although the results presented above show that Field D* generally produces less costly paths than regular grid-based planning, this is not guaranteed. It is possible to construct pathological scenarios where the linear interpolation assumption is grossly incorrect (for instance, if there is an obstacle in the cell to the right of the center cell in Figure 5(i) and the optimal path for node s_2 travels above the obstacle and the optimal path for node s_1 travels below the obstacle). In such cases, the interpolated path cost of a point on an edge between two nodes may be either too low or too high. This in turn can affect the quality of the extracted solution path. However, such occurrences are very rare, and in none of our random test cases (nor any cases we have ever

encountered in practice) was the path returned by Field D* more expensive than the grid-based path returned by D* Lite. In general, even in carefully-constructed pathological scenarios the path generated by Field D* is very close in cost to the optimal solution path.

Moreover, it is the ability of Field D* to plan paths with a continuous range of headings, rather than simply its lower-cost solutions, that is its true advantage over regular grid-based planners. In both uniform and nonuniform cost environments, Field D* provides direct, sensible paths for our agents to traverse.

7 Conclusion

In this paper we presented Field D*, an extension of classical grid-based planners that uses linear interpolation to efficiently produce less costly, more natural paths through grids. We have found Field D* to be extremely useful for mobile robot path planning in both uniform and nonuniform cost environments.

We and others are currently extending the Field D* algorithm in a number of ways. Firstly, a 3D version of the Field D* algorithm has been developed for vehicles operating in the air or underwater [2]. We are also developing a version that interpolates over headings, not just path costs, to produce smoother paths when turning is expensive. Finally, we are also working on a version of the algorithm able to plan over nonuniform grids, for vehicles navigating through very large environments.

Acknowledgments

This work was sponsored by the U.S. Army Research Laboratory, under contract “Robotics Collaborative Technology Alliance” (contract number DAAD19-01-2-0012). The views contained in this document are those of the authors and should not be interpreted as representing the official policies or endorsements of the U.S. Government. Dave Ferguson is supported in part by a National Science Foundation Graduate Research Fellowship.

References

1. O. Brock and O. Khatib. High-speed navigation using the global dynamic window approach. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.
2. J. Carsten. 3D Field D*. Master’s thesis, Carnegie Mellon University, Pittsburgh, PA, 2005.
3. E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

4. D. Ferguson and A. Stentz. The Field D* Algorithm for Improved Path Planning and Replanning in Uniform and Non-uniform Cost Environments. Technical Report CMU-RI-TR-05-19, Carnegie Mellon School of Computer Science, 2005.
5. P. Hart, N. Nilsson, and B. Rafael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4:100–107, 1968.
6. A. Kelly. *An Intelligent Predictive Control Approach to the High Speed Cross Country Autonomous Navigation Problem*. PhD thesis, Carnegie Mellon University, 1995.
7. S. Koenig and M. Likhachev. D* Lite. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2002.
8. S. Koenig and M. Likhachev. Improved fast replanning for robot navigation in unknown terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002.
9. S. Koenig and M. Likhachev. Incremental A*. In *Advances in Neural Information Processing Systems*. MIT Press, 2002.
10. K. Konolige. A gradient method for realtime robot control. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2000.
11. R. Larson. A survey of dynamic programming computational procedures. *IEEE Transactions on Automatic Control*, pages 767–774, 1967.
12. R. Larson and J. Casti. *Principles of Dynamic Programming, Part 2*. Marcel Dekker, New York, 1982.
13. S. LaValle. *Planning Algorithms*. Cambridge University Press (also available at <http://msl.cs.uiuc.edu/planning/>), 2006. To be published in 2006.
14. M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun. Anytime Dynamic A*: An Anytime, Replanning Algorithm. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2005.
15. M. Likhachev, G. Gordon, and S. Thrun. ARA*: Anytime A* with provable bounds on sub-optimality. In *Advances in Neural Information Processing Systems*. MIT Press, 2003.
16. J. Mitchell. *Handbook of Computational Geometry*, chapter Geometric Shortest Paths and Network Optimization, pages 633–701. Elsevier Science, 2000.
17. J. Mitchell and C. Papadimitriou. The weighted region problem: finding shortest paths through a weighted planar subdivision. *Journal of the ACM*, 38:18–73, 1991.
18. N. Nilsson. *Principles of Artificial Intelligence*. Tioga Publishing Company, 1980.
19. R. Philippson. *Motion Planning and Obstacle Avoidance for Mobile Robots in Highly Cluttered Dynamic Environments*. PhD thesis, EPFL, Lausanne, Switzerland, 2004.
20. R. Philippson and R. Siegwart. An Interpolated Dynamic Navigation Function. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.
21. N. Rowe and R. Richbourg. An efficient Snell’s-law method for optimal-path planning across two-dimensional irregular homogeneous-cost regions. *International Journal of Robotics Research*, 9(6):48–66, 1990.
22. J. Sethian. A fast marching level set method for monotonically advancing fronts. *Applied Mathematics, Proceedings of the National Academy of Science*, 93:1591–1595, 1996.

23. S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja, and K. Schwehr. Recent progress in local and global traversability for planetary rovers. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2000.
24. C. Stachniss and W. Burgard. An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2002.
25. A. Stentz and M. Hebert. A complete navigation system for goal acquisition in unknown environments. *Autonomous Robots*, 2(2):127–145, 1995.
26. Anthony Stentz. The Focussed D* Algorithm for Real-Time Replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

Tradeoffs Between Directed and Autonomous Driving on the Mars Exploration Rovers

Jeffrey J. Biesiadecki, Chris Leger, and Mark W. Maimone

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA USA
first.last@jpl.nasa.gov

Summary. NASA's Mars Exploration Rovers (MER) have collected a great diversity of geological science results, thanks in large part to their surface mobility capabilities. The six wheel rocker/bogie suspension provides driving capabilities in many distinct terrain types, the onboard IMU measures actual rover attitude changes (roll, pitch and yaw, but not position) quickly and accurately, and stereo camera pairs provide accurate position knowledge and/or terrain assessment. Solar panels generally provide enough power to drive the vehicle for at most four hours each day, but drive time is often restricted by other planned activities. Driving along slopes in nonhomogeneous terrain injects unpredictable amounts of slip into each drive. These restrictions led us to create driving strategies that maximize drive speed and distance, at the cost of increased complexity in the sequences of commands built by human Rover Planners each day.

The MER rovers have driven more than a combined 10 kilometers over Martian terrain during their first 21 months of operation using these basic modes. In this paper we describe the strategies adopted for selecting between human-planned directed drives versus rover-adaptive Autonomous Navigation and Visual Odometry drives.

Keywords: Mars Rover, MER, Space Robotics, Autonomy, Mission Planning

1 Background

NASA successfully landed two mobile robot geologists on the surface of Mars in January 2004: the Spirit and Opportunity Mars Exploration Rovers (MER). Their primary goal was to find evidence of past water at Gusev Crater and Meridiani Planum, two geologically distinct sites on opposite sides of the planet. Each rover was instrumented with a suite of tools for remote sensing (multi-filter and stereo camera pairs and a thermal emission spectrometer) and *in situ* measurement (5 DOF arm for deploying a grinding Rock Abrasion Tool, Microscopic Imager, Alpha Particle X-ray Spectrometer, and Mössbauer

Spectrometer). Although the achievement of their successful landings stands out as a technological tour de force, it was their ability to traverse while on the surface of Mars that enabled both rovers to succeed in their primary goals.

The MER rovers are typically commanded once per Martian solar day (or *sol*). A sequence of commands sent in the morning specifies the sol's activities: what images and data to collect, how to position the robotic arm, and where to drive. At the end of each sol, the rovers send back the images and data human operators will use to plan the next sol's activities. The next sol's mobility commands are selected based on what is known – and what is unknown – about the terrain ahead.

1.1 Rover Mobility Commands

The rovers are driven using three primary modes: low-level commands that specify exactly how much to turn each wheel and steering actuator, directed driving primitives for driving along circular arcs (of which straight line driving and turn-in-place are special cases), and autonomous path selection.

Several types of potential vehicles hazards are checked *reactively*, most of them during Real Time Interrupts (RTIs) which occur 8 times per second. Available checks include Tilt/Pitch/Roll, Northerly Tilt, Rocker/Bogie Suspension Angles, Motor Stalls, Limit Cycle (no forward progress), and Resource Contention.

The rovers maintain an estimate of their local position and orientation updated at 8 Hz while driving. Position is first estimated based on wheel odometry, and orientation is estimated using an Inertial Measurement Unit that has 3-axis accelerometers and 3-axis angular rate sensors. In between driving primitives, the rover can use camera-based Visual Odometry (VisOdom) to correct the errors in the initial wheel odometry-based estimate. VisOdom tracks terrain features in NavCam stereo images and uses the tracking information to estimate true vehicle motion during small steps; the rover can only move roughly 60cm, or turn 15 degrees, before successive NavCam images lack enough overlap to reliably estimate motion [2].

Both directed and path selection modes of driving can make use of onboard stereo vision processing and terrain analysis software to determine whether the rover would encounter geometric hazards along its chosen path.

The computing resources required by these different commands vary greatly. Directed driving commands execute the most quickly (achieving speeds up to 124 m/hour), but also have greater risk since the rover can only count wheel rotations to estimate position, and never looks ahead to evaluate the terrain before driving onto it. AutoNav commands detect and avoid geometric hazards, but only achieve driving speeds from 10 m/hour in obstacle-laden terrain up to 36 m/hour in safe terrain, and also rely on the accuracy of the wheel odometry to track obstacles once they leave the field of view of the cameras. VisOdom commands provide accurate position estimates

(but not obstacle detection), and require close spacing between images which limits the top speed to 10 m/hour.

Autonomous Terrain Analysis

When information about nearby terrain is unavailable or uncertain, the rover can be commanded to evaluate terrain safety by performing stereo vision and terrain assessment autonomously. This allows the rover to *predictively* locate traverse hazards and avoid them. The procedure is not summarized here; see [4, 1] for details and [9] for the approach that inspired it.

The rock-strewn terrain encountered by Spirit at Gusev Crater corresponded well to the exponential rock distribution models predicted using data from Viking I, II and Pathfinder missions [5]. The body-mounted 120-degree Field of View (FOV) HazCams were designed with this terrain model in mind, and Spirit has performed all of its autonomous terrain assessment using these cameras. However, the terrain encountered by Opportunity at Meridiani Planum is vastly different. Instead of a wide variety of rocks at many scales, much of the terrain consists of very fine-grained materials; so fine, in fact, that no large scale features can be found in the wide FOV HazCam images at 256x256 resolution. Fortunately, the lack of large scale features implies a lack of large “step” obstacles. So, Opportunity was reconfigured to perform terrain assessment with more narrow FOV NavCam images. Rock and fissure obstacles can still be detected, but the limited FOV means less of the terrain around the obstacle will be understood, which reduces its ability to steer around them autonomously.

All MER surface software runs on a 20 MHz RAD6000 computer under the VxWorks operating system. The slow processor speed, and the sharing of a single address space and cache by dozens of tasks, mean Autonomous Navigation (AutoNav) and VisOdom software run slowly.

1.2 Ground-Based Terrain Analysis

Ground-based terrain assessment is generally performed using stereo image pairs taken by any of the three types of stereo camera pairs found on MER vehicles. There are two pairs of wide field-of-view (120 degree, 10cm baseline) Hazard Cameras (HazCams) rigidly mounted 53cm above the ground plane on the front and back sides, one pair of medium field-of-view (45 degree, 20cm baseline) Navigation Cameras (NavCams) mounted 152cm above the ground plane on a pan/tilt head, and one pair of narrow field-of-view (18 degree, 28cm baseline) Panoramic Cameras (PanCams) also mounted 152cm above the ground plane on the pan/tilt head. [8] These cameras take up to 1024x1024 12-bit images that provide information about terrain texture throughout their images, and stereo range-derived terrain shape at different scales: around 0.5m - 5m in the HazCams, 2m - 20m in the NavCams, and 4m - 70m in the PanCams.

The amount of directed driving that can be commanded depends on both the terrain itself and on how much information about the terrain is available. Orbital imagery, while crucial for long-range planning, cannot resolve vehicle hazards like 20cm rocks. So after each long drive, images from each appropriate camera pair are requested.

Downlinked stereo image pairs are processed by an automated pipeline that generates derived products including 3D range maps, texture-mapped terrain meshes, and color overlays indicating terrain properties such as slope and elevation [6]. Rover operators use image-based querying tools to measure ranges to terrain features and estimate distances and rock sizes [3]. For example, a “ruler” tool allows the operator to measure the distance between the 3D points corresponding to two pixels in an image or image mosaic, useful for identifying discrete obstacles such as rocks or steps. Terrain meshes give the operator a geometric understanding of the terrain and of spatial relationships between terrain features and the planned path, and allow simulation of drive sequences to predict drive safety and performance [10]. The raw images are also extremely useful in assessing traversability: operators can readily identify very sandy or very rocky areas that present hazards, though new terrain types always carry an element of uncertainty regarding vehicle performance. In some cases, no image cues enable rover operators to predict the performance of a drive; patches of terrain only a few meters apart, with similar surface texture and geometry, can lead to drastically different traction or sinkage. For example, while driving uphill toward a topographic high point named “Larry’s Lookout” on Sol 399, Spirit reached 100% slip (i.e. no forward progress) on a 16 degree slope, but only a few meters further had only 20% slip on a 19 degree slope with no discernible difference in appearance.

Humans are very good at terrain analysis for motion planning. In addition to geometric hazards such as rocks or drop-offs, humans can readily identify and classify new terrain types (e.g., sandy versus rocky slopes) on the basis of appearance alone. In contrast, the MER software does not have any appearance-based terrain analysis capabilities, it only detects geometric obstacles. Nevertheless, the most serious and frequent hazards (rocks, steps, and high-center hazards) can be detected by geometric analysis—assuming sufficient range data is available. At longer ranges (over 15m in NavCam images, and over 50m in PanCam images), range data becomes sparse, making it impossible to rely solely on geometric analysis. The rover is better able to assess nearby hazards, but its lack of a global planner (which the human stands in for during manual drives) can cause the rover to get stuck in cul de sacs.

2 Drive Techniques and Templates

Most drive sequences can be classified as either traverses (covering maximum distance) or approaches (driving to a specific position for subsequent *in situ* arm operations). The techniques used for each drive type are determined based

on the time allocated for driving, the amount of terrain visible in imagery, known hazards, and level of uncertainty in rover position given the terrain type. Generally, driving on level ground requires a mix of blind and AutoNav driving, and driving on slopes requires using VisOdom to allow the rover to compensate for unpredictable slip.

2.1 Traversing the Plains

We learned during our initial drives in each terrain that driving on level ground typically leads to accurate and predictable mobility performance; e.g., Spirit only accumulated 3% position error over 2 kilometers of driving [7]. Because of the rover's limited processing power, drives using autonomous hazard avoidance are several times slower than "blind" (manually-directed) drives. These two facts favor long initial blind drives to achieve the longest drives in the least amount of rover execution time. Human operators can easily identify rocks that are large enough to be hazardous to the rover, and can plan complex paths that avoid them. The firm surfaces found on the plains of Gusev crater often allowed for blind drives of up to 70m.

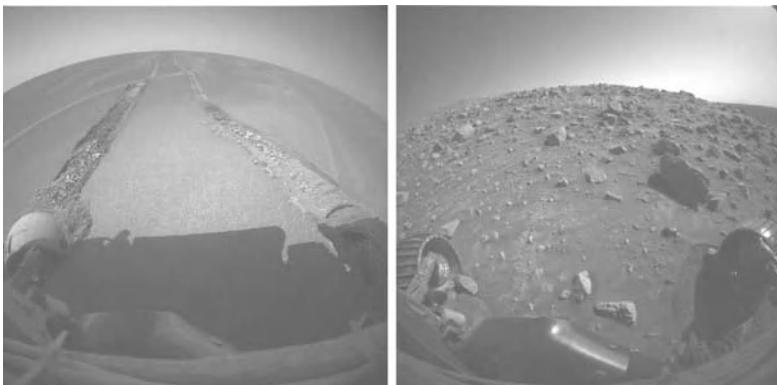


Fig. 1. Left: On Sol 446, Opportunity found its wheels more than half buried in sand. Although not a geometric hazard, the ripple of sand on which it stopped kept the human planners busy for weeks. Right: On Sol 454, Spirit terminated its drive early after detecting 90% slip. This image shows rocks that had collected next to the left front wheel.

On the plains of Meridiani, the terrain hazards were quite different and initially allowed for blind drives over 100m. Unlike the Gusev plains, there was a near-total absence of rocks at Meridiani, and until Sol 446 (see Figure 1) none of the innumerable sandy ripples posed a threat to the rover. Craters, visible in orbital imagery, and small linear depressions were the most significant hazards for Opportunity. While driving over flat terrain, the rover's suspension does not articulate significantly, which suggested that a measured

suspension articulation change could be used to halt driving if the rover were to encounter a depression. In April 2004, the rover's software was upgraded to allow the rover's suspension angles to be checked against preset limits at 8Hz, thus enabling the rover to stop at negative terrain features (i.e., holes) that were not visible *a priori*. Because the reason for halting a drive (e.g., time-out, suspension check, slip amount, or tilt check) is accessible to the rover sequencing language, a recovery maneuver could be performed whenever the suspension check tripped. The recovery consists of backing up several meters and continuing the drive with AutoNav, since AutoNav is able to detect and avoid negative hazards.

Both rovers use a common strategy at the end of long traverses to acquire necessary images for manipulator operations and turn to a preset heading that minimizes the multi-path interference caused by the rover's mast during communication with Earth or an orbiter. However, this presents a problem for the next sol's IDD operations: since no camera can see the part of the IDD deployment volume under the rover, a front HazCam image pair of the final terrain must be safely acquired 0.5-3m before driving to the rover's final location in order to determine if the IDD can be safely deployed.

The obvious solution is to turn to the desired heading, acquire the image pair, then drive a short distance to the final location. The "guarded arc" drive primitive solves this problem by only executing the post-turn drive segment if the onboard terrain analysis shows that it is safe to do so.

2.2 Driving on Slopes: Mountains and Craters

While most of the distance covered by the rovers was on level ground, most of the sols and most of the approach drives occurred on slopes. The rovers invariably slip when driving on slopes, making VisOdom essential for safe and accurate driving. But using AutoNav along with VisOdom takes roughly twice as much time as VisOdom alone, making the combination impractical for normal use.

This presents a challenge: the rover has the ability to know where it is, but in that mode cannot detect obstacles. Additionally, in steep terrain the rover cannot identify all obstacle classes, since the rover has no means of detecting sandy, high-slip areas in advance. Even otherwise safe areas of moderate slope may represent hazards if there are steeper slopes or rocks downhill, since slippage in moderate slopes could take the rover into dangerous areas. In these cases, the rover operators specify "keep out zones" which will cause the rover to halt driving before a hazard is encountered (e.g., see Figure 3). The rover keeps track of its position using VisOdom (and can preclude driving if VisOdom fails) and can close the loop to correct for slippage. On Sol 454, Spirit promptly halted driving after detecting slippage over 90%, and post-drive HazCam images showed several rocks on the verge of falling into the wheels, since the wheels had dug into the terrain by nearly one wheel radius (see Figure 1). The recurrence of high slopes, sandy terrain with intermixed

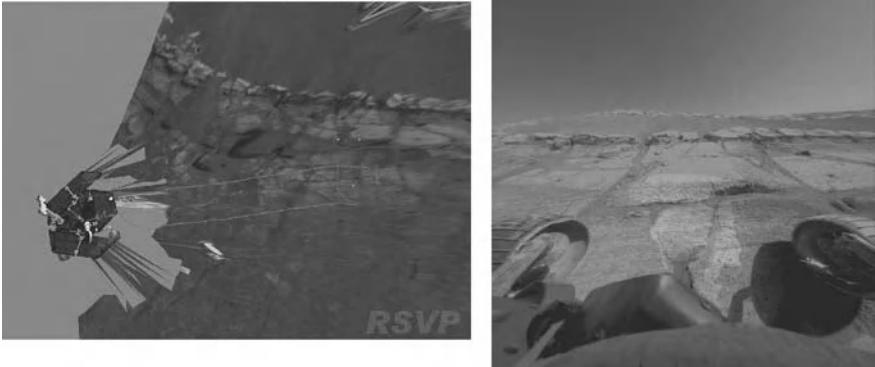


Fig. 2. Opportunity’s planned 8.7 meter drive along a 20-24 degree slope on Burns Cliff on Sol 304, and the front HazCam view confirming a successful single sol approach. The shaded area shows those parts of the surface reachable by the instrument arm, which includes the light bedrock that was the target of the drive. A combination of VisOdom and conditional sequencing was used to accomplish this drive.

small rocks, and frequent obstacle-sized rocks caused us to retreat and find a new route to the summit of Husband Hill.

2.3 Target Approach

Whereas traverse sequences focus on covering maximum distance over terrain, target approach sequences aim to place the rover at a specific target position and orientation for *in situ* examination of rocks and soil with the rover’s manipulator, or less frequently, high-resolution imagery of a distributed or inaccessible target region. The accuracy requirements for positioning the rover for *in situ* work are relatively tight, often within 10cm.

On level ground, directed drive primitives are usually sufficient for accurate target approaches from 2-10m away. On sloped terrain, VisOdom is required to close the loop on the rover’s position. After each motion, VisOdom updates the rover’s position knowledge, allowing it to correct for slip-induced errors. Conditional sequencing that confirms the current distance to multiple targets is often used in conjunction with visual odometry to accurately approach targets 5-10m away while driving on slopes in the 10 to 20 degree range (e.g., see Figure 2), with the caveat that on surfaces with sufficiently low bearing strength, the rover is mechanically incapable of making direct uphill progress.

3 Relative Merits of Directed/Autonomous Driving

There are significant differences in resource usage between manual and autonomous driving, with execution time and generated data volume being the

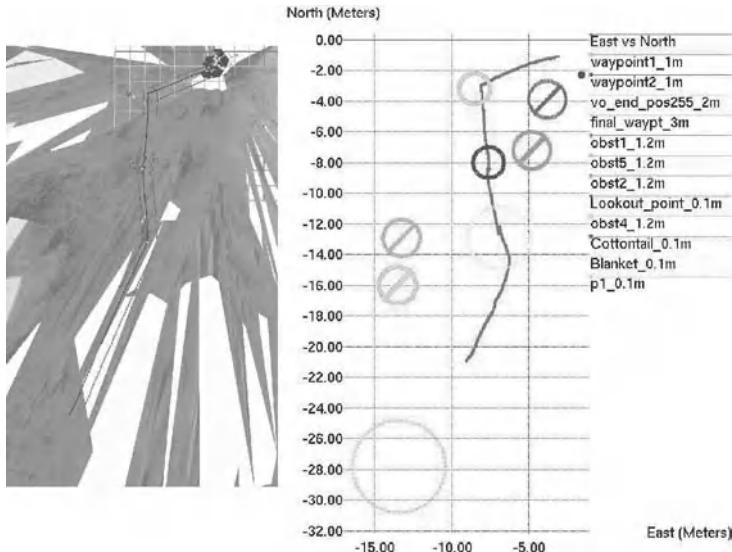


Fig. 3. Spirit's Sol 436 drive used a variety of driving modes. A simulation of the planned drive over a 3D terrain mesh is shown on the left, the actual course taken on the right. Circles indicate the waypoints, slashed circles the obstacles and keep-out zones developed by human Rover Drivers by inspecting stereo images and simulating drives over the 3D mesh. Spirit drove south (downward) over 26 actual meters before reaching its time limit. Jagged lines in the course plot above the -12 meter line indicate the discrete jumps resulting from VisOdom updates, those at the -12 meter line show AutoNav backing up to avoid a small ridge blocking its path southwest.

most obvious. Power is also impacted by execution time, for although the power used by the mobility system is the same whether a trajectory was generated manually or autonomously, the rover's CPU, IMU, and other electronics draw power for the duration of the drive and thus an autonomous drive will require more power than a manual drive of the same distance.

Less obvious differences in resource requirements between manual and autonomous driving also exist. The most significant is planning time: it takes a rover operator more time to identify obstacles and choose appropriate waypoints when sequencing a blind drive than when sequencing a drive using AutoNav (e.g., see Figure 3). During the first few months of the mission, it often took up to 10 hours to build a drive sequence to travel 20-40m across the plains of Gusev. This decreased dramatically later in the mission, often requiring only 2-4 hours to sequence drives over 100m in length on either rover. Still, a directed drive places full responsibility for vehicle safety on the rover operator rather than allowing the rover to safeguard itself, thus requiring more time for manual terrain analysis and waypoint selection. This suggests

an obvious trade-off between sequencing time and execution time for directed and autonomous drives.

There is an additional long-term resource trade-off: humans can rapidly adapt their sequences to deal with new terrain types or drive requirements, but changing the onboard software involves a lengthy software development, testing, and uplink process. Instead of a day-to-week turnaround in sequence development, software updates to cope with new terrain and drive techniques occur on a months-to-year cycle.

3.1 Driving into the Unknown

There is one notable circumstance in which the human has no ability to safely select paths: when driving into terrain that has not been imaged. On Sol 109, Spirit was commanded to drive over the local horizon 50m distant as it descended from the rim of Missoula Crater. In this case, AutoNav was the only option available to drive further and use the available time and power, and post-drive images showed AutoNav correctly avoiding large rocks while traversing slopes up to 9 degrees (see Figure 4). Obviously, a high degree of confidence in the hazard avoidance software is needed in situations such as this; *AutoNav has kept both vehicles safe through over 2500 meters of traverse as of August 2005*. Less severe, but more frequent, instances in which humans cannot guarantee rover safety occur when the rover drives beyond the distance at which obstacles can be resolved, or through smaller occluded regions. In practice, even when using AutoNav the rover operator typically chooses waypoints that avoid the most hazardous areas, thus taking advantage of the perceptual strengths of both human and rover.



Fig. 4. On Sol 109, Spirit avoided obstacles in previously-unseen terrain.

3.2 Execution

Directed drives have a limited ability to deal with errors or uncertainty in execution. Whereas AutoNav can close the loop on vehicle safety by imaging the terrain that the rover is about to drive through, a directed drive must make the assumption that the rover does not deviate far enough from the planned path to encounter any hazards. For longer drives or in high-slip areas, the rover must be able to deal with accumulated position error, either through safeguarding itself or by using VisOdom to update its position knowledge. When using VisOdom, the rover operator is responsible for specifying the criteria for halting the drive, since manually sequencing reliable obstacle avoidance is too difficult. Typically, the halting criteria include proximity to known obstacles, the number of times VisOdom has failed to provide a position update, and a threshold on slippage.

Figure 5 summarizes the distance covered and the type of driving modes used for each rover during their first 19 months of operation.

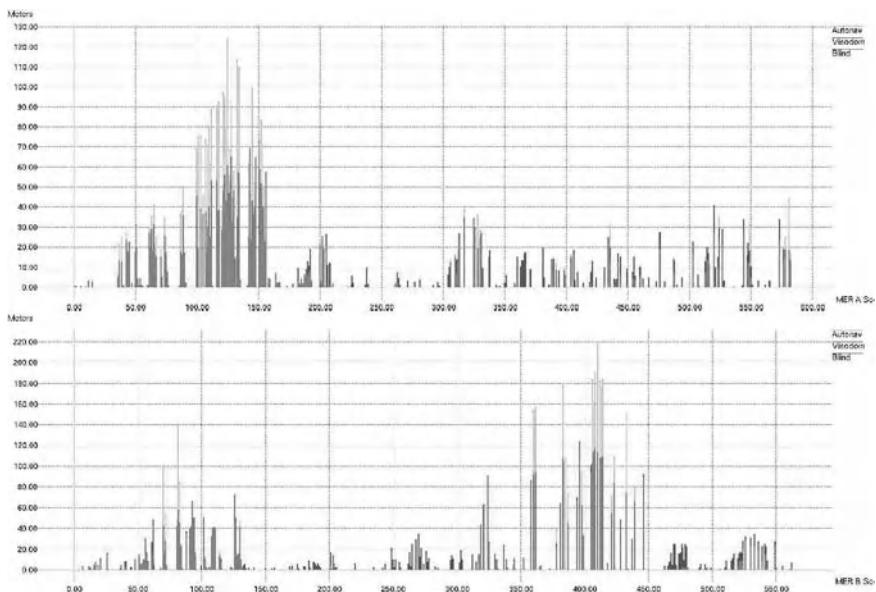


Fig. 5. Summary of distances driven by each rover (Spirit above Opportunity) per Sol. AutoNav drives (in green) include any mode in which terrain assessment was done onboard (i.e., both AutoNav and Guarded motion), VisOdom drives (in blue) include both directed and adaptive driving modes but not AutoNav, and Blind drives (in red) include both directed arcs and rover-adapted drives that compensated for yaw changes measured during the drive. The changing quality of the drive types suggests how human and rover driving strategies alike had to adapt to new terrains many times over the course of each mission.

3.3 Adaptation

Mobility performance is uncertain in any novel terrain type and can vary substantially in known terrain types, but humans can quickly learn to steer the rover clear of newly identified hazard types. For example, after Spirit drove through a loose mixture of fine sand and rocks on Sol 339, a potato-sized rock jammed in one of the wheels, finally coming out a week later. When the rover encountered similar terrain over 100 sols later, rover operators knew to direct Spirit to check for slippage while driving and stop if the rover became bogged down. Post-drive images after the rover detected over 90% slip showed a similar mixture of sand and rocks, with two rocks having the potential to jam in the wheels, and we subsequently retreated to look for another route (see Figure 1). This sort of perception and adaptation with a single training example is a key strength of manual terrain analysis.

4 Future Work

While Spirit and Opportunity continue to perform well beyond our original expectations, our experience operating the rovers suggests some areas for improvement. Perhaps the most obvious area for improvement is computational efficiency: driving with either VisOdom or AutoNav can slow the rovers' progress by up to an order of magnitude compared to directed drives. Some speedup can likely be obtained by accepting decreased accuracy: one use of VisOdom is to simply detect when the rover is slipping substantially, in which case a precise motion estimate is not required.

Another promising avenue for future work is terrain classification. Our current hazard avoidance software detects only geometric hazards, but areas with weak soil—particularly wind-driven drifts—have proven treacherous for both rovers. The ability to learn what high-slip terrain looks like so that it can be autonomously avoided (even dynamically updating the onboard interpretation of the terrain) would be a great benefit. One potentially useful observation is that slippage is almost always correlated with sinkage, and sinkage can be measured by observing either the wheels or their tracks.

In terms of mobility system development, one area that seems to be underemphasized is precision mobility in natural terrain. For the types of investigation undertaken by Spirit and Opportunity, mere mobility—the ability to traverse a certain-sized obstacle, travel at a certain rate, or climb a certain slope—is not sufficient. The ability to reliably navigate the rover to within centimeters of a desired location, on slopes, near obstacles, and with external constraints on final vehicle heading, has been of the utmost importance in uncovering the water history of Mars.

Flexibility in the rovers' command language and onboard software has been critical in allowing us to encode our ever-changing understanding of the terrain and vehicle performance. While not a traditional robotics problem, it

would be beneficial to introduce methods for easily formalizing and re-using new sequence idioms to reduce human errors and speed the sequence design, simulation and validation processes. Writing a sequence is writing a program, and perhaps techniques could be applied from extreme programming and other software development paradigms.

MER software development continues today. Several technologies are being evaluated for possible uplink in mid-2006. These include autonomous *in situ* instrument placement following a successful drive (aka Go and Touch), global path planning to enable intelligent backtracking, visual servoing, and autonomous detection of dust devils and clouds in onboard imagery.

Future vehicles will have faster processors, allowing more advanced terrain analysis and path selection to be performed. But path planning can only be as good as the underlying obstacle avoidance methodology, and if rovers are to become substantially autonomous then appearance-based adaptive terrain analysis will also be required.

5 Conclusion

Successful operation of the MER vehicles has depended on both manually-directed and autonomous driving. The two methods are complementary, and careful selection of the right techniques leads to better overall performance in the face of limited time, power, imagery, and onboard computation.

While most of the distance covered by both rovers has been on level ground with varying degrees of geometric hazards, most of the time has been spent in more challenging environments coupling steep slopes with loose materials and positive obstacles. Careful terrain analysis is required in these cases, and VisOdom has also been absolutely essential for safe and accurate driving.

Acknowledgements

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology under contract to the National Aeronautics and Space Administration. Thousands of people were involved in the mission at JPL and throughout the world, in government, academia, and industry. We gratefully acknowledge their outstanding work, which enabled us to explore Mars through Spirit and Opportunity. We would like to thank the entire Mars Exploration Rover team, and in particular the other rover planners: Khaled Ali, Eric Baumgartner, Paolo Bellutta, Bob Bonitz, Brian Cooper, Frank Hartman, Scott Maxwell, Ashley Stroupe, Ashitey Trebi-Ollennu, Edditt Tunstel, John Wright, and Jeng Yen.

References

1. Jeffrey Biesiadecki, Mark Maimone, and Jack Morrison. The Athena SDM rover: A testbed for Mars rover mobility. In *International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS)*, Montreal, Canada, June 2001. <http://robotics.jpl.nasa.gov/people/mwm/sdm-mobility/>.
2. Yang Cheng, Mark Maimone, and Larry Matthies. Visual odometry on the Mars Exploration Rovers. In *IEEE Conference on Systems, Man and Cybernetics*, The Big Island, Hawaii, USA, October 2005.
3. Paul Backes et al. Sequence planning for the Fido Mars rover prototype. In *IEEE Aerospace Conference*, Big Sky, Montana, USA, 2003.
4. Steven B. Goldberg, Mark W. Maimone, and Larry Matthies. Stereo vision and rover navigation software for planetary exploration. In *IEEE Aerospace Conference*, volume 5, pages 2025–2036, Big Sky, Montana, USA, March 2002. <http://robotics.jpl.nasa.gov/people/mwm/visnavsw/>.
5. M. Golombek and D. Rapp. Size-frequency distributions of rocks on Mars and Earth analog sites: Implications for future landed missions. *Journal of Geophysical Research - Planets*, 102(E2):4117–4129, February 1997.
6. C. Leger and R. Deen. Remote image analysis for Mars Exploration Rover mobility and manipulation operations. In *IEEE Conf. on Systems, Man and Cybernetics*, Big Island, Hawaii, USA, October 2005.
7. Rongxing Li, Brent A. Archinal, Raymond E. Arvidson, Jim Bell, Philip Christensen, Larry Crumpler, David J. Des Marais, Kaichang Di, Tom Duxbury, Matt Golombek, John Grant, Ronald Greeley, Joe Guinn, Andrew Johnson, Randolph L. Kirk, Mark Maimone, Larry H. Matthies, Mike Malin, Tim Parker, Mike Sims, Shane Thompson, Steven W. Squyres, and Larry A. Soderblom. Spirit rover localization and topographic mapping at the landing site of Gusev Crater, Mars. *JGR-Planets, Special Issue on Spirit Rover*, to appear, 2005.
8. J. N. Maki, J. F. Bell III, K. E. Herkenhoff, S. W. Squyres, A. Kiely, M. Klimesh, M. Schwochert, T. Litwin, R. Willson, A. Johnson, M. Maimone, E. Baumgartner, A. Collins, M. Wadsworth, S. T. Elliot, A. Dingizian, D. Brown, E. C. Hagerott, L. Scherr, R. Deen, D. Alexander, and J. Lorre. Mars exploration rover engineering cameras. *Journal of Geophysical Research*, 108(E12):12–1–12–24, December 2003. <http://www.agu.org/pubs/crossref/2003/2003JE002077.shtml>.
9. Reid Simmons, Lars Henriksen, Lonnie Chrisman, and Greg Whelan. Obstacle avoidance and safeguarding for a lunar rover. In *AIAA Forum on Advanced Developments in Space robotics*, Madison, WI, August 1996. <http://www.cs.cmu.edu/~reids/papers/AIAAobsAvoid.pdf>.
10. J. Yen, B. Cooper, F. Hartman, S. Maxwell, and J. Wright. Sequence rehearsal and validation on surface operations of the Mars Exploration Rovers. In *SpaceOps*, Montreal, Canada, 2004.

List of Abbreviations and Symbols

- AutoNav Autonomous Navigation for Surface Operations
 DOF Degrees of Freedom
 HazCams Hazard-detection Camera

MER Mars Exploration Rover

NavCams Navigation Camera

PanCams Panoramic Imaging Cameras

RTI Real-time Interrupt (8 per second)

Sol Martian Solar Day, about 24 hours and 40 minutes in duration

VisOdom Visual Odometry

Surface Mining: Main Research Issues for Autonomous Operations

Eduardo M. Nebot

Australian Centre for Field Robotics / CRC Mining University of Sydney
nebot@cas.edu.au

1 Abstract

This paper presents the author's view on the main challenges for autonomous operation in surface mining environment. A brief overview of the mine operation is presented showing the number of components that needs to interact in a safe, robust and efficient manner. Successful implementation of autonomous systems in field robotic applications are presented with a discussion of the fundamental problems that needs to be addressed to have this technology accepted in mining operations.

2 Introduction

Resource based economies will be facing enormous challenges to remain competitive in a global economy. Mine operations are in most cases located in isolated areas making the relocation of personnel very expensive. Furthermore, this situation is becoming more difficult these days since the deposits discovered are much smaller and the life of the mine may not justify the establishment of new towns. The development of field robotics automation is one of the key factors to address this problem. Automated and autonomous systems are beginning to make a significant appearance in different areas. At the simplest level, such systems act as adjuncts to manned vehicles in providing, for example, location information, collision warning, or driver enhancements. At a more complex level, a number of automated machines for hauling, excavation and loading are being introduced and have had some success mainly in underground mining. At the level of a complete mine, it is possible to envision the fusion of positional, geophysical information into a complete mine "picture" and the subsequent use of this to exert overall coordinated control of individuals, vehicles and systems in the mine. The overriding strategy in the development of the digital mine is the concept of "systems of systems". This concept is pervasive in military and other complex systems communities. It

recognizes that the ultimate system, a mine in this case, is composed of many different system units, and that these in turn are composed of yet smaller systems. The key to the successful management of the overall mining system is to understand how component systems need to work together and to devise technology and procedures to allow these elements to function as part of the overall system. Specifically in mining, the mine site consists of personnel, vehicles and machines that have to be allocated according to the best geophysical knowledge available, market conditions and financial constraints, Figure 1. Figure 2 shows a simplified model of the mine operation. Once the mine pit is prepared the trucks are loaded with different type of machines and the ore is moved through haul roads to the crusher. It has been estimated that the cost of haulage accounts for 40 – 50% of the surface mine operating expenses [1]. The real time availability of information such as ground conditions, orebody morphology, grade distribution will make high fidelity simulation of systems essential to rapidly adapt to the dynamics of the actual circumstances. The full integration and analysis of systems will also be of fundamental importance to develop the concept of distance mining. It is now conceivable to integrate machine monitoring and control, geophysical sensing and remote image analysis with different levels of resolution sensors into a global database. This information can then be accessed from different locations around the world to minimize the number of personnel at the site. This will require the development and deployment of different technologies into the mine environment. This paper presents discussion of where automation have worked in the past, a vision for the mine in the future and some steps that will need to be follows to enable full autonomous mines.



Fig. 1. Mining equipment involved in a mine operation

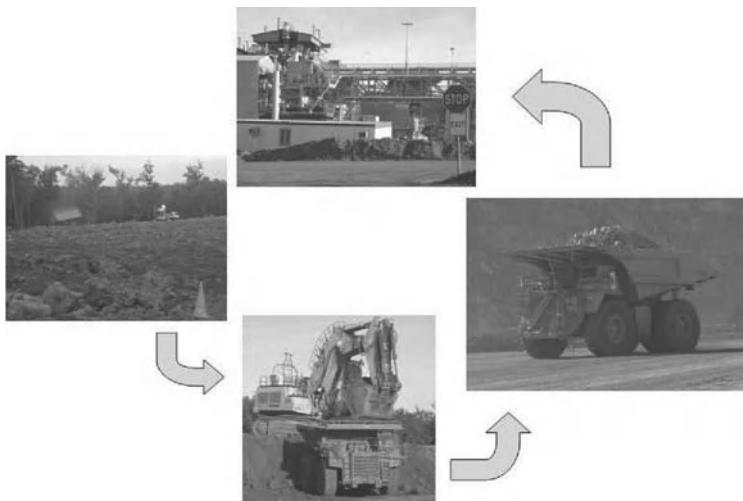


Fig. 2. Simplified view of a mine operation

3 Mining Automation

3.1 Future of Surface Mining

Due to the uncertainty on the state of resources actual surface mining operations needs to operate in a conservative manner to be able to satisfy customer requirements and shareholder profit. In the future Surface mining operation will be:

- **Product Driven:** customers will demand specialized product that meets their individual requirements.
- **Flexible:** they will employ operations methods focussed towards swings as opposed to base line productions.
- **Agile:** capable of operating in a market where orders change quickly or even daily basis.

Mine sites will use whole-of-mine plans and will view planning as a dynamic and reactive process. The management system are likely to take the form of very comprehensive spatio-temporal database representing the actual physical structure of the mine: geology, resources, pit layout, road structure etc., [3]. They will dynamically revise and evaluate operating decision based on comprehensive costing models and forecasting tools. Will use accurate real time monitoring technologies to track their performance against projections, feeding back information to the planning process. These changes will be facilitated by the emergence of technology that deliver focussed, high quality

timely information that enables performance against production targets to be accurately tracked. These new technology will also reduce the variability in the operation making the equipment more reliable and the production more predictable. This operation will require breakthroughs in resource characterization. It is expected that new tools will be available to enable the resources to be mapped (seam thickness, structural properties, compositions) to sub-meter accuracies. Excavation equipment will have on-board sensing used to provide real-time visualization ahead of the mine face. These sensors will be part of the mine network and their data will be used to update and validate the resource map in real-time. The mine plan and mine status will be contained in an environment that contains up to the minute whole-of-mine data including mine topography, resource maps, equipment deployment etc. The information will be maintained and updated automatically from a variety of sources including fly-over images, sensors mounted on mobile platforms, dynamic resource maps etc. This information will be visualizable anywhere in the site making the mine operation safe and efficient.

3.2 Where Has Mining Automation Worked in the Past

There are a number areas where automation has been successful in the mining area. These examples are in rail haulage, process control type applications, conveyor systems and to a certain extend in underground drilling. An example of a process type application is shown if Figure 3. In this case a laser is used to track the position of the wagon and control the bin door to load the coal. The system also recognizes the locomotive and establish the communication with the train driver to synchronize the whole operation.

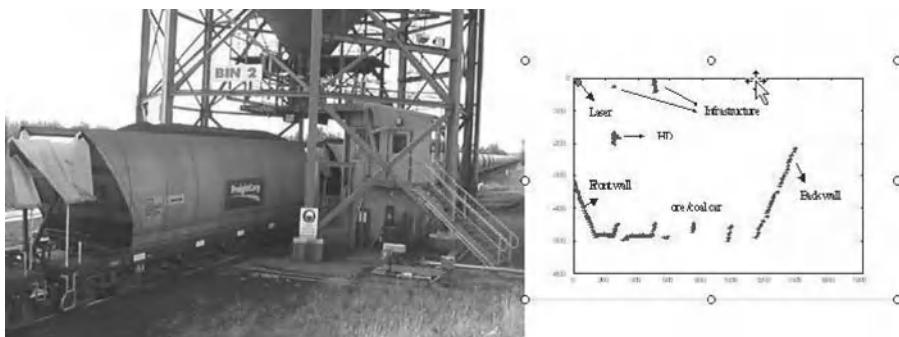


Fig. 3. Automatic coal loading application using a laser to detect the position of the wagon

There are other new automation application that are much more sophisticated. These include:

- Autonomous LHD's
- Autonomous Underground and Surface trucks
- Autonomous straddle carrier
- Automated underground face drills
- Longwall automation
- Dragline swing automation
- Autonomous drilling / rock recognition
- ...

Among them the most advanced implementation of autonomous system in underground environment is the LHD automation. Figure 4 shows an early version of an automated LHD. The system navigates in the mine by looking at the walls of the tunnel[2]. It has been demonstrated in 1999 and has been operating in various mines since then. Figure 5 shows a commercial implementation of a free range autonomous straddle carrier [6]. The system allows the straddles to move and stack containers from the quay, into the holding yards, onto vehicles and back to quay cranes with cm accuracy. Both systems work in an area where only autonomous systems operate.



Fig. 4. Automated LHD



Fig. 5. Autonomous Straddle carriers operating in Fishermans Islands, Australia

3.3 Hard Problems in Mining Automation

The unit operation that offers the greatest potential for reducing operating costs in surface mining is haulage. The enabling technologies (navigation, truck control, and collision detection) for autonomous haul trucks (AHS) exist in a semi-mature form and prototype AHS integrating these technologies are known to have been developed and tested by at least two of the major manufacturers (Komatsu and Caterpillar). The Komatsu system has the capability to navigate a haul route, dump automatically to hoppers or to the ground, and work with some type of loading equipment. The system leverages off several mature technologies, notably the global positioning system (GPS) and inertial navigation systems (INS) for navigation and millimetre-wave radar and laser systems for safety and collision detection. The system is designed to operate in an area with only autonomous trucks.

Nevertheless there are no actual deployment of these systems in any mine in the world. Although automation of large machines is already well advanced this technology is limited by the extent to which automated trucks interact with other equipment and by the system integrity that can be incorporated at a reasonable cost. The last statement requires further clarification. All the previous successful autonomous applications have the following characteristics:

- Structured environment
- Well defined automated task requirements
- There is a need for the solution
- Site willing to adopt the new technology

- Simple / Robust technology
- No interaction with manned machines

On the contrary the environment where surface mine haul truck need to operate are:

- Rugged environments(Dust, moisture, extreme weather conditions)
- Dynamic and often unpredictable
- Unstructured and often defined by geology
- Difficult to sense and costly to incorporate integrity
- Difficult to build simple, effective and robust models
- Significant interaction with manned machines

In general the challenges in mining automation become increasingly formidable as the level of autonomy increases. Automation technologies will only flourish after the evolution of the real-time, whole-of-mine, information systems takes place. Such systems are virtually mandated where autonomous equipment is to interact with other equipment (manned or autonomous) by providing the framework for managing the interaction. Without such a system, equipment interactions need to be eliminated or very closely managed to the point where the limitations outweigh the benefits of automation. Removing the driver means that the functions he or she performs beyond driving need to be performed automatically. These consist largely of monitoring the overall health of the truck including detecting, isolating, and reporting faults and monitoring the environment, e.g. the quality of the road surface. Most of these activities are not formal tasks, but rather occur as part of the driver's broader state of awareness. For this to happen we need new sensors and perception technology capable of determining the actual state of the world under all possible environmental conditions. For example, an autonomous truck should be able to determine the different states of the road as shown in Figure 6 and adjust the driving conditions accordingly to optimize the use of the truck.

One of the main issues in autonomy is integrity. That is the design of autonomous system with enough sensory redundancy in the frequency domain to detect any possible fault [5]. This problem can be seen in Figure 8 where a millimetre wave radar has been designed to be able to monitor the state of the face under the presence of smoke / dust. Perception and interpretation will also be an area of significant importance in mining applications. Figure 8 shows an example where the 3-D radar data is used to extract the position of nearby vehicles next to shovel based of known models of potential resources in proximity.

3.4 The Next Step

The most significant advances in the next few years are likely to come through tools such as operator assists and partial automation that develop and prove



Fig. 6. Typical road state under different environmental conditions

component technologies while building deeper understanding and awareness of the equipment operating and issues leading full automation. This is very important since mine operation are essentially high risk. Each year, hundreds of mining haulage accidents occur, resulting in a significant number of deaths and injuries as well as costs through replacement, repair and downtime. Many of these accidents are due to microsleep events that are a manifestation of operator fatigue. This issue has been researched extensively and it has been demonstrated that humans will have a significant number of microsleep occurrences per shift when they are required to work at night [4]. During these events the driver loses control of the truck for a short period of time. This is becoming a more important issue where larger trucks are introduced without a corresponding enlargement of haul road widths. Most mine managers are now more aware of these problems and are starting to take a more active role in supporting the development of new operator fatigue systems. Figure 9 presents a variety of accidents involving haul trucks. Some of these accidents are due to the driver falling asleep and veering off / or crossing the centre of the road. Others are due to poor visibility of smaller objects or vehicles in close proximity to the truck. This section presents two examples of aiding equipment that can prevent many of these accidents. The first problem is addressed



Fig. 7. Millimetre Wave radar installed in a rope shovel

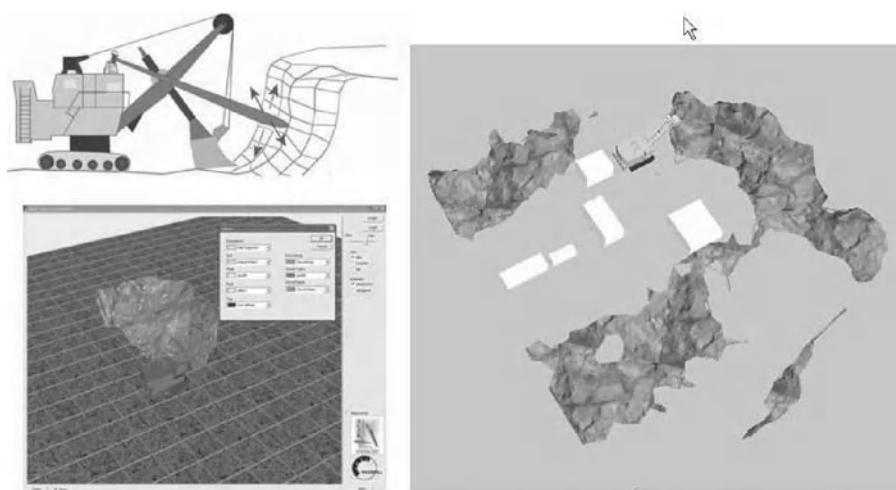


Fig. 8. Automatic object recognition

with an approach that uses a scanning lasers mounted on haul trucks. These sensors monitor the position of a truck with respect to a series of PVC poles embedded in the berms along the length of the haul road. This information is used to define a valid corridor for the vehicle, Figure 10. If the truck wanders outside this corridor an alarm is triggered. This system has been fitted to a fleet of 15 Komatsu 730E trucks, Figure 11, and has been in routine use at Alcoa's Huntly and Willowdale mines for more than a year. The system has proven to be reliable and is regarded by these mines as an invaluable tool to allow large trucks to be used safely on relatively narrow roads [7]. It has the obvious additional safety benefit of helping to prevent accidents caused by operator fatigue.



Fig. 9. Haul Truck accidents

The second problem can be addressed by using a wireless network and GPS sensors to detect the position of other vehicles in the area of operation, Figure 12. The area of operation will be a function of line of sight of the antennas of the agents in proximity. Once the agent becomes part of the network they will know the position and velocity of all the other agents in the proximity area. A protocol implemented in the Haul Truck system will generate a series of alarms according to the threats. The installation of this technology has also

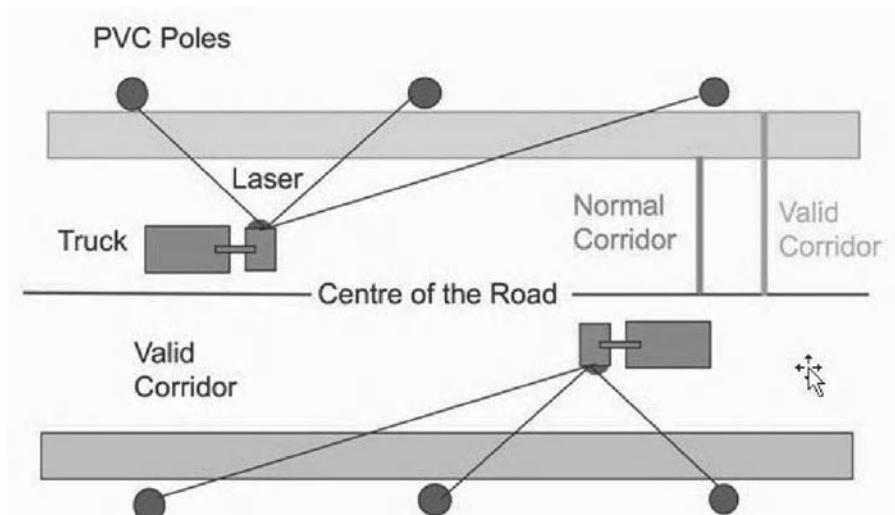


Fig. 10. Basic principle to determine the position of the truck in the road



Fig. 11. Truck fleet and a truck retrofitted with a laser based tracking system

the potential to be used to move information around the mine. It is easy to see that by using the haul trucks we can move information from the different parts of the mine and download it to base stations installed in frequently visited areas like the crusher. This information can then flow in the internal intranet of the mine and can be used for other monitor and optimization purposes.

4 Conclusions

This paper presented some successful autonomous application of field robotic in structured mining type environments. It also presented an overview of some

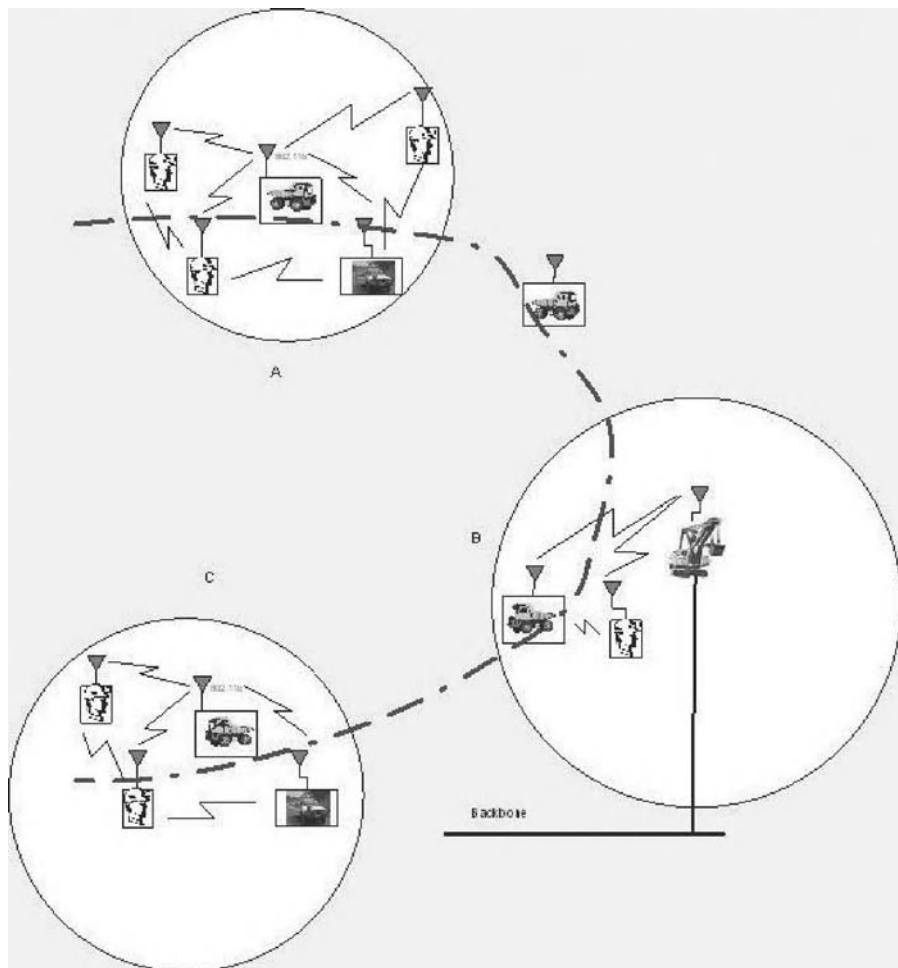


Fig. 12. Local ad-hoc network based proximity system

of the important challenge faced in mine automation and the areas where progress is needed to enable fully autonomous mining.

References

1. Mukhopadhyay A (1989) Selection, maintenance, and relations of various parameters for off-highway hauling tires. In: Off-Highway Haulage in Surface Mines, Ed. Golosinski, Sraje, Balkema, pp 153-159.

2. Roberts J, Duff E, Corke P, Sikka P, Winstanley G, Cunningham. Autonomous control of underground mining vehicles using reactive navigation. In: Proceedings of IEEE Int. Conf. on Robotics and Automation, pages 3790-3795, San Francisco, USA.
3. Lever P, McAree R ACARP (2003) Project C11054 Open-cut Automation Scoping Study.
4. Williams S, Asleep at the wheel March 2002, World Mining Equipment, incorporating World Mining and Mining Equipment International.
5. Scheding S, Nebot E, Durrant-Whyte H High integrity Navigation using frequency domain techniques IEEE Transaction of System Control Technology, July 2000, Vol 8, Iss. 4, pp 676-694.
6. Sukkarieh S, Nebot E, Durrant-Whyte A High Integrity IMU/GPS Navigation Loop for Autonomous Land Vehicle applications”, IEEE Transaction on Robotics and Automation, June 1999, Vol 15, No 3, p 85-95.
7. Nebot E, Guivant J, Worrall S, Haul Truck Alignment Monitoring and operator warning system To appear in Journal of Field Robotics (2006)

Robotic Vision

Session Overview

Robotic Vision

Yoshiaki Shirai¹ and Bob Bolles²

¹ Osaka University

² SRI International

At the DARPA Grand Challenge in October 2005, laser range finders, especially the ones manufactured by SICK, were the predominant range sensors. Does that mean that stereo sensors are dead? No. It means that laser scanners satisfied the requirements of the Grand Challenge outdoor vehicle navigation application better than stereo. Stereo sensors, on the other hand, are the sensor of choice for several other applications, such as people monitoring and human-computer interfaces, because they are passive, relatively inexpensive, have no moving parts, and provide registered range and color data.

In this session, the authors present camera-based algorithms for computing 3-D descriptions of scenes. Two of the papers focus on stereo techniques and one on monocular constraints.

Sibley, Matthies, and Sukhatme describe two types of biases associated with stereo analysis, and then describe approaches that dramatically reduce their effects. In the case of triangulation-based range estimation, their new approach reduces the bias by an order of magnitude. To accomplish this reduction, they re-express stereo triangulation as a second order series expansion, taking into account the distribution of errors associated with image-based stereo matching and the calculation of range. The second type of bias occurs when several stereo measurements are combined, often over time, to improve the precision of the measured range values. To reduce this bias, they developed an iterative non-linear Gauss-Newton technique that focuses on image space measurements instead of directly averaging/filtering 3-D range values.

Blake et al describe a technique for combining stereo analysis, color analysis, and occlusion reasoning to segment an image into foreground, background, and occluded regions. Their approach uses a probabilistic model formulated as a Conditional Random Field that fuses prior information about the expected structures in a scene with stereo and color results. They apply their technique to a video conferencing application and show its effectiveness at precisely extracting the foreground people, which is the key to several enhancements, including automatic camera control, eye-gaze correction, and the insertion of virtual objects into the scene.

Delage, Lee, and Ng approach the problem of extracting a 3-D description of a scene quite differently than the other two groups in this session. They use generic knowledge about buildings and cameras to derive 3-D models of indoor scenes from monocular images. In particular, their technique uses such facts as floors are horizontal planes and walls are vertical planes to extract and interpret linear edge patterns as floors and walls. They use a Markov Random Field to label every pixel in an image as part of a surface or edge, and then apply an iterative 3-D reconstruction algorithm. Their approach, which includes an analysis of the vanishing points, locates a floor footprint first, and then fills in the walls.

Bias Reduction and Filter Convergence for Long Range Stereo

Gabe Sibley^{1,2}, Larry Matthies², and Gaurav Sukhatme¹

¹ Robotic Embedded Systems Laboratory,
University of Southern California,
Los Angeles, CA 90089
`gsibley|gaurav@usc.edu`

² Jet Propulsion Laboratory,
California Institute of Technology,
Pasadena, CA 91109
`lhm@helios.jpl.nasa.gov`

Summary. We are concerned here with improving long range stereo by filtering image sequences. Traditionally, measurement errors from stereo camera systems have been approximated as 3-D Gaussians, where the mean is derived by triangulation and the covariance by linearized error propagation. However, there are two problems that arise when filtering such 3-D measurements. First, stereo triangulation suffers from a range dependent statistical bias; when filtering this leads to over-estimating the true range. Second, filtering 3-D measurements derived via linearized error propagation leads to apparent filter divergence; the estimator is biased to under-estimate range. To address the first issue, we examine the statistical behavior of stereo triangulation and show how to remove the bias by series expansion. The solution to the second problem is to filter with image coordinates as measurements instead of triangulated 3-D coordinates. Compared to the traditional approach, we show that bias is reduced by more than an order of magnitude, and that the variance of the estimator approaches the Cramer-Rao lower bound.

1 Introduction

This paper details our efforts to enhance long range depth estimation in stereo systems by filtering feature measurements from image sequences. We would like to accurately estimate the depth of distant objects from disparities on the order of 1 pixel. Improving depth estimation in stereo systems is an important pursuit. For instance, better stereo range resolution will enhance a robot's ability to perform tasks such as navigation, long range path planning, obstacle avoidance, mapping and localization and high-speed driving. Unbiased sensing is a prerequisite for these algorithms to perform well.

In the balance of this paper we will encounter two problems with traditional stereo error modeling and in turn describe their solutions. First, because

of the non-linearity in stereo triangulation, we will see that range estimates produced by standard triangulation methods are statistically biased. While bias in stereo is a known phenomenon, previous research focused on how range bias is induced from uncertain camera positions[3, 16, 20], or dismissed it as insignificant [12]. Second, filtering sequences of 3-D measurements from stereo leads to biased range estimates when the uncertainty of each 3-D measurement is modeled by standard linearized error propagation techniques; this stems from the fact that the uncertainty model is biased.

For the former issue, analyzing the statistical behavior of stereo triangulation leads us to new triangulation equations based on series expansion; this new bias-corrected formulation is shown to be an improvement over traditional stereo triangulation by more than order of magnitude. For the latter biased filter problem, we find that formulating the filter with image coordinates as measurements leads to efficient and unbiased estimation of range. Lastly, using the Fisher information inequality we show that the combination of bias-corrected stereo and a Gauss-Newton recursive filter yield estimates that closely approach the minimum variance Cramer-Rao lower bound.

2 Statistical Bias in Stereo

Consider some general stereo triangulation function $s : \mathbb{R}^4 \mapsto \mathbb{R}^3$

$$s(\mathbf{z}) = \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \quad (1)$$

where the current observation, \mathbf{z} is the vector of pixel coordinates $[u_1, v_1, u_2, v_2]^T$, and the pixels $[u_1, v_1]^T$ and $[u_2, v_2]^T$ are projections of \mathbf{x} into the two camera image planes. Let x_2 be the range component of \mathbf{x} - i.e. x_2 is aligned with the optical axis of the cameras.

We are interested in how a particular model of pixel measurement uncertainty will translate into range uncertainty. Before we address the issue of bias in more detail we first need to establish an appropriate observation probability density function.

2.1 Measurement Distribution

A common approximation is that many measurements of a stationary feature point, such corner features [6, 8], follow a normal distribution [10, 12, 13, 15]. To establish how features are actually distributed we have performed the following experiment: we took a sequence of images from a stationary camera of a stationary checker board and tracked the corners over time with sub-pixel accuracy [11]; we then re-centered each feature track about zero by subtracting its mean. For each pixel dimension a histogram of all the measurements is then

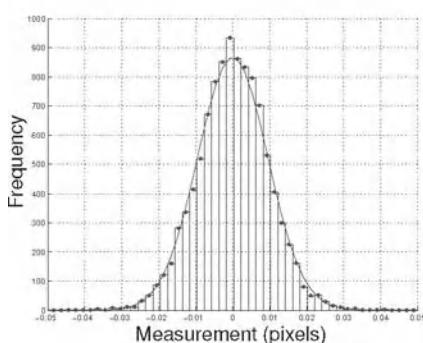


Fig. 1. Feature measurement histogram of 10,360 measurements.

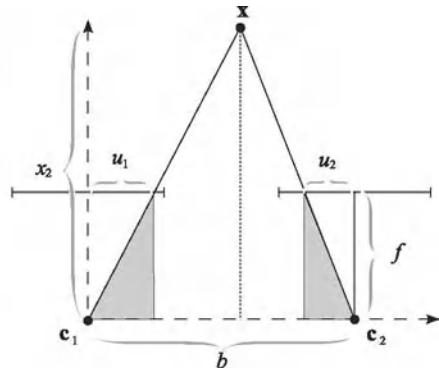


Fig. 2. Standard model of linear perspective projection for stereo triangulation with axis aligned cameras.

plotted. This histogram approximates the true distribution we should expect in measurements. Qualitatively, the histogram in Fig. 1 indicates that the distribution is close to Gaussian.

2.2 Derived Range Distribution

Recall the fronto-parallel configuration, whose geometry is shown in Fig. 2. We will derive the range p.d.f. using this simple geometry, though the methods and results used here apply to other camera models as well. Using Fig. 2, the stereo equations are

$$s(\mathbf{z}) = \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} u_1 b/d \\ v_1 b/d \\ (b f)/d \end{bmatrix} \quad (2)$$

where the last element of \mathbf{x} is the range component, and $d = (u_1 - u_2)$ is the disparity. Monte-Carlo simulation using these equations indicates that if image feature positions, and hence disparity, are normally distributed, then the expected range will be biased toward over estimating the true value³. The bias is empirically visible in Fig. 3. Analytically, the bias can be seen by deriving the range p.d.f., $f_{x_2}(x_2)$, from the disparity p.d.f., $f_d(d)$ [4, 9, 12]. From (2) we have $x_2 = s_2(d) = k/d$, where $k = bf$. Since s_2 and s_2^{-1} are continuously differentiable, then

$$f_{x_2}(x_2) = f_d(d) \left| \frac{\partial f_d(d)}{\partial x_2} \right| = f_d[s_2^{-1}(x_2)] \left| \frac{\partial s_2^{-1}(x_2)}{\partial x_2} \right|$$

³ Throughout this paper we use linear camera models with a resolution of 512x384 pixels, a horizontal FOV of 68.12 degrees and vertical FOV 51.3662 of degrees

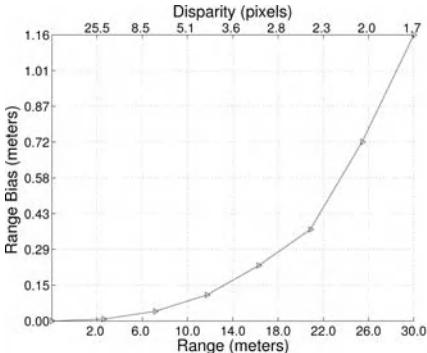


Fig. 3. Range vs. Bias at 8 different ranges averaged over 10,000 trials. Clearly, bias is a strong function of range. Pixel standard deviation is 0.3 pixels in each pixel dimension, with no covariance.

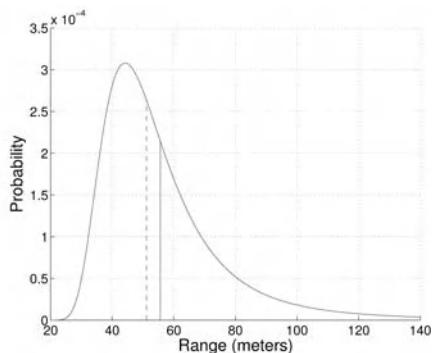


Fig. 4. Range p.d.f. for $\sigma_d = 0.3$ pixels. True range is 51.08m = 1 pixel disparity. Note that because of the tail, the mean is at 55.63m, which is a bias of almost 10%.

where $|\cdot|$ denotes absolute value of the Jacobian determinant and $s_2^{-1} = d = k/x_2$. Thus, since $f_d(d)$ is modeled as Gaussian

$$f_{x_2}(x_2) = \frac{k}{\sqrt{2\pi}\sigma_d x_2^2} \exp\left(\frac{-(k/x_2 - \mu_d)^2}{2\sigma_d^2}\right) \quad (3)$$

where μ_d and σ_d are the disparity mean and variance. The mean of (3) is

$$\mu_{x_2} = E[x_2] = \int_{-\infty}^{\infty} x_2 f_{x_2}(x_2) dx_2$$

which unfortunately does not appear to have an analytical solution, so we resort to numerical integration. Plots of $f_{x_2}(x_2)$ are shown in Fig. 4; clearly, for distant features the p.d.f. is non-Gaussian, non-symmetric and exhibits a long tail. The tail shifts the mean away from the true range and hence we see the source of bias in stereo.

2.3 Bias Reduction

Naturally, we would like an unbiased method for calculating range. Recall that the distribution on \hat{d} is approximately Gaussian, the mean of which we take to approximate some true underlying state, d . If the true value of d was known, then the true unbiased range could be calculated with $x_2 = s_2(d)$, but due to the variation in \hat{d} , $s_2(\hat{d})$ is, as we have seen, slightly biased. However, if the variation of \hat{d} around d is small, then a Taylor series expansion of s_2 may provide a better estimate [2],

$$\tilde{s}_2(d) \approx s_2(d) + \frac{\partial s_2}{\partial d} \Big|_d (\hat{d} - d) + \frac{1}{2} (\hat{d} - d)^2 \frac{\partial^2 s_2}{\partial d^2} \Big|_d$$

Taking the expectation, noting that $E[\hat{d} - d] = 0$, that $E[(\hat{d} - d)^2]$ is the definition of variance, and replacing d with \hat{d} we get

$$s_2(\hat{d}) \approx \tilde{s}_2(\hat{d}) - \frac{1}{2} \text{var}(\hat{d}) \frac{\partial^2 s_2}{\partial d^2} \Big|_{\hat{d}} \quad (4)$$

which is the new range equation that we use to correct for bias. Note that this formulation requires accurate knowledge of the measurement variance; which is reasonable. Looking at Fig. 5 the improvement is immediately visible for small disparities; in fact, for the ranges shown,

bias is reduced by more than an order of magnitude. Note that higher order series approximation, which should theoretically provide a better estimate, will depend on higher order moments, $E[(\hat{d} - d)^n]$, $n > 2$. But if the input distribution has negligible higher order moments, then the second term in (4) makes use of all the available information. By considering the variance and how it impacts the range distribution, this bias-correction method largely removes the bias from long range stereo.

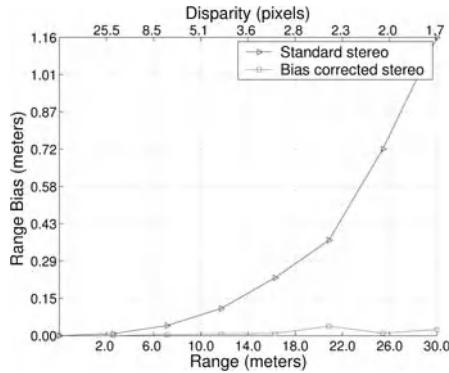


Fig. 5. Bias-corrected stereo compared to traditional stereo.

3 3-D Estimation

In this section we uncover another type of bias that results from filtering a sequence of 3-D estimates produced by triangulation and linearized error propagation. To alleviate this we develop a non-linear Gauss-Newton iterative measurement update using image space measurements instead of 3-D measurements. Finally, the statistical efficiency of the 3-D measurement update and the Gauss-Newton update are compared to the Cramer-Rao lower bound.

3.1 3-D Measurement Update

Let $\mathbf{x} \in \mathbb{R}^3$, $\hat{\mathbf{x}} \in \mathbb{R}^3$, $\mathbf{z}_{3D} \in \mathbb{R}^3$ denote the current state, current state estimate and the current 3-D observation, respectively. For the case at hand, the current observation, \mathbf{z}_{3D} , is the vector found via bias-corrected stereo. The state estimate and observation are independent realizations of multivariate Gaussian distributions: $\mathbf{z}_{3D} \sim N(s(\mathbf{z}), \mathbf{R}_{3D})$ and $\hat{\mathbf{x}} \sim N(\mathbf{x}, \hat{\mathbf{P}})$ where \mathbf{R}_{3D} and $\hat{\mathbf{P}}$

are the measurement and state error covariance matrices, respectively. The error covariance matrix \mathbf{R}_{3D} is found via error propagation of image errors

$$\mathbf{R}_{3D} = \frac{\partial s}{\partial \mathbf{z}} \mathbf{R} \frac{\partial s^T}{\partial \mathbf{z}}, \quad \mathbf{R} = \begin{bmatrix} \Sigma_l & 0 \\ 0 & \Sigma_r \end{bmatrix} \quad (5)$$

where s is the bias corrected stereo equation and Σ_l and Σ_r are 2x2 error covariance matrices from the left and right images, respectively.

In this scenario the sensor model $h_{3D} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ is the vector function that returns a predicted measurement for \mathbf{z}_{3D} given \mathbf{x} . If \mathbf{x}_{ws} is the global position of the stereo head with orientation matrix \mathbf{R}_{ws} then the generative sensor model is, $h_{3D}(x) = \mathbf{R}_{ws}^T x - \mathbf{R}_{ws}^T x_{ws}$. Note that while we focus here on the stationary case and solving issues of bias and filter convergence, this formulation extends to the mobile sensor case (an issue we are actively working on that is beyond the scope of this paper). Following standard notation [14] the Kalman Filter update equations for this system are

$$\begin{aligned} \hat{\mathbf{x}}_{k+1}^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}(\mathbf{z}_{3D} - h_{3D}(\hat{\mathbf{x}}_k^-)) \\ \hat{\mathbf{P}}_{k+1}^+ &= (\mathbf{I} - \mathbf{K}\mathbf{H}_k)\hat{\mathbf{P}}_k^- \\ \mathbf{K} &= \hat{\mathbf{P}}_k^- \mathbf{H}_k^T (\mathbf{H}_k \hat{\mathbf{P}}_k^- \mathbf{H}_k^T - \mathbf{R}_{3D})^{-1} \end{aligned} \quad (6)$$

where \mathbf{H}_k is the Jacobian of h_{3D} . Filtering with this setup leads to the situation depicted in Fig. 6, which clearly shows what is called *apparent divergence* - i.e. convergence to the wrong result[7]. This can be explained by the fact that linearized error propagation in (5) gives quadratically larger range variance for more distant features. This means that the weighted average in (6) will always place more confidence in closer measurements and $\hat{\mathbf{x}}^+$ will be biased toward the short measurements. In essence, linearized error propagation leads to over confidence for shorter measurements, which in turn leads to serious filter bias.

3.2 Gauss-Newton Measurement Update

Instead of using the triangulated point \mathbf{z}_{3D} as the observation, let the observation again be the vector of pixel coordinates $\mathbf{z} = [u_1, v_1, u_2, v_2]^T$. Thus our sensor model $h : \mathbb{R}^3 \mapsto \mathbb{R}^4$ is the vector function that projects \mathbf{x} into the left and the right images,

$$h(\mathbf{x}) = \begin{bmatrix} h_l(\mathbf{x}) \\ h_r(\mathbf{x}) \end{bmatrix} \quad (7)$$

where $h_l : \mathbb{R}^3 \mapsto \mathbb{R}^2$ and $h_r : \mathbb{R}^3 \mapsto \mathbb{R}^2$ are the left and right camera projection functions. Depending on the camera models in use, h_l and h_r can be formulated in a variety of ways [18, 19]. We only require that these functions and their first derivatives are available, and otherwise leave them unspecified.

For convenience we choose to formulate the measurement update as an iterative Gauss-Newton method, which is equivalent to an iterated Extended

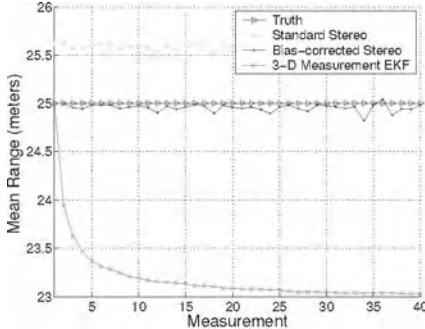


Fig. 6. Kalman Filter of a sequence of 40 3-D stereo measurements averaged over 10,000 trials.

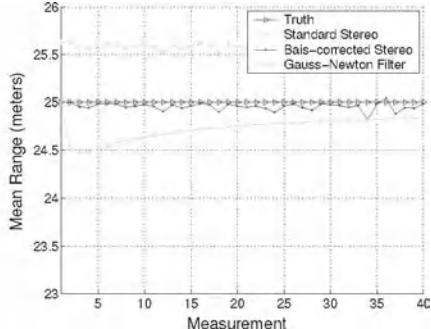


Fig. 7. Gauss-Newton filter of a sequence of 40 image measurements averaged over 10,000 trials.

Kalman Filter [1]. To integrate prior information about $\hat{\mathbf{x}}$ we write the current state estimate and current observation as a single measurement vector

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z} \\ \hat{\mathbf{x}} \end{bmatrix}, \quad g(\mathbf{x}) = \begin{bmatrix} h(\mathbf{x}) \\ \mathbf{x} \end{bmatrix}$$

For the first measurement, the filter is initialized with $\hat{\mathbf{x}} = \mathbf{z}_{3D}$ and $\hat{\mathbf{P}} = \mathbf{R}_{3D}$ which are calculated by bias-corrected triangulation and linearized error propagation in as in section 3.1. Since the current observation and state estimate are realizations of independent normal distributions we have $\mathbf{Z} \sim N(g(\mathbf{x}), \mathbf{C})$ where $\mathbf{C} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix}$. Given the measurement \mathbf{Z} , we can write the likelihood function

$$\mathcal{L}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^7 |\mathbf{C}|}} \exp\left(-\frac{1}{2}(\mathbf{Z} - g(\mathbf{x})) \mathbf{C}^{-1} (\mathbf{Z} - g(\mathbf{x}))\right) \quad (8)$$

where $|\cdot|$ is the determinant. The maximum likelihood estimate for this expression is $\hat{\mathbf{x}}^+ = \operatorname{argmax}_{\mathbf{x}} \mathcal{L}(\mathbf{x})$, whose solution is equivalent the solution minimizing the negative log-likelihood, $\operatorname{argmin}_{\mathbf{x}} \ell(\mathbf{x})$,

$$\ell(\mathbf{x}) = \frac{1}{2}(\mathbf{Z} - g(\mathbf{x})) \mathbf{C}^{-1} (\mathbf{Z} - g(\mathbf{x})) + k \quad (9)$$

where k is a constant. If we let $\mathbf{S}^T \mathbf{S} = \mathbf{C}^{-1}$ and

$$r(\mathbf{x}) = \mathbf{S}(\mathbf{Z} - g(\mathbf{x})) \quad (10)$$

then (9) is a non-linear least squares problem to minimize $r(\mathbf{x})^T r(\mathbf{x})$. The Gauss-Newton method to solve non-linear problems of this form is the sequence of iterates [5]

$$\mathbf{x}_{i+1} = \mathbf{x}_i - (\mathbf{J}(\mathbf{x}_i)^T \mathbf{J}(\mathbf{x}_i))^{-1} \mathbf{J}(\mathbf{x}_i)^T r(\mathbf{x}_i) \quad (11)$$

where \mathbf{J} is the Jacobian of (10). Noting that $\mathbf{J} = -\mathbf{S}\mathbf{G}_i$ where \mathbf{G}_i is the Jacobian of $g(\mathbf{x}_i)$, (11) becomes

$$\mathbf{x}_{i+1} = (\mathbf{G}_i^T \mathbf{C}^{-1} \mathbf{G}_i)^{-1} \mathbf{G}_i^T \mathbf{C}^{-1} (\mathbf{Z} - g(\mathbf{x}_i) + \mathbf{G}_i \mathbf{x}_i)$$

which is the familiar normal equation solution. Once iterated to convergence the covariance $\hat{\mathbf{P}}^+$ can then be approximated using $\hat{\mathbf{P}}^+ = (\mathbf{G}^T \mathbf{C}^{-1} \mathbf{G})^{-1}$. As noted in [1], this is equivalent to the iterated Extended Kalman Filter measurement update.

Filtering with this setup leads to the situation depicted in Fig 7. Typically, the measurement update converges after 3 to 4 iterations. Compared to the 3D measurement update, the fact that the Gauss-Newton (IEKF) method converges without bias is not surprising considering that we avoid the intermediate stereo triangulation and linearized error propagation for calculating the 3-D error covariance matrix.

3.3 Estimator Efficiency

Having derived a bias-corrected estimator it is important to address its efficiency, that is, how well it approximates a minimal variance estimate of the parameters. The information inequality, $\text{cov}_{\mathbf{x}}(\mathbf{x}) \geq \mathcal{I}_{\mathbf{z}}(\mathbf{x})^{-1}$, defines such a bound, which is called Cramer-Rao lower bound[4]. Here the Fisher information matrix $\mathcal{I}_{\mathbf{z}}(\mathbf{x})$ is given by the symmetric matrix whose ij^{th} element is the covariance between first partial derivatives of the *measurement* log-likelihood function,

$$\mathcal{I}_{\mathbf{z}}(\mathbf{x})_{i,j} = \text{cov}_{\mathbf{x}} \left(\frac{\partial \ell_{\mathbf{z}}}{\partial \mathbf{x}_i}, \frac{\partial \ell_{\mathbf{z}}}{\partial \mathbf{x}_j} \right) \quad (12)$$

The measurement log-likelihood function is $\ell_{\mathbf{z}}(\mathbf{x}) = \frac{1}{2}(\mathbf{z} - h(\mathbf{x}))\mathbf{R}^{-1}(\mathbf{z} - h(\mathbf{x})) + k$. For a multivariate normal distribution (12) reduces to[17, 20]

$$\mathcal{I}_{\mathbf{z},i,j}(\mathbf{x}) = \frac{\partial h}{\partial \mathbf{x}_i}^T \mathbf{R}^{-1} \frac{\partial h}{\partial \mathbf{x}_j}$$

For n independent identically distributed (i.i.d.) measurements the Fisher information is simply $n\mathcal{I}$. An estimator that achieves the CRLB is said to be efficient. Fig. 8 shows range variance convergence for the Gauss-Newton estimator; this demonstrates that the Gauss-Newton stereo estimator is efficient.

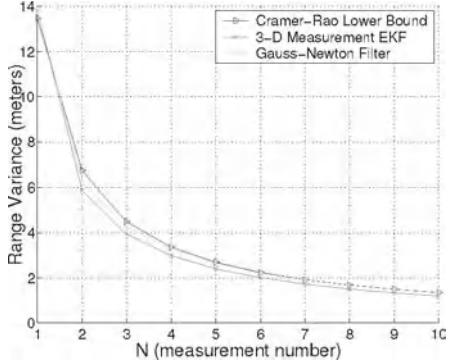


Fig. 8. Gauss-Newton and 3-D EKF estimator efficiency compared against the Cramer-Rao lower bound over a sequence of 10 measurements of a feature 25m away. At each step estimator variance is found via Monte-Carlo simulation over 10,000 trials.

4 Conclusion

In our efforts to improve long range stereo by filtering image sequences we have come across two problems: the first is that stereo triangulated range estimates are statistically biased. To address this we have re-expressed the stereo triangulation equations using a second order series expansion. This new formulation reduces bias in stereo triangulation by more than an order of magnitude. The second problem is that temporal filtering of 3D stereo measurements also leads to biased estimates. The solution to this problem is to filter with image coordinates as measurements instead of triangulated 3D coordinates. Finally, using the Fisher information inequality we show that the bias-corrected Gauss-Newton stereo estimator approaches the minimum variance Cramer-Rao lower bound. While the scope of this paper is constrained to address stereo bias and estimator efficiency, our ultimate goal is to filter feature points from a moving platform. This is a task that requires a solid solution to the simultaneous localization and mapping problem, which we are actively exploring.

References

- [1] F.W. Bell, B.M. Cathey. The iterated Kalman filter update as a Gauss-Newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, Feb 1993.
- [2] D.R. Cox and D.V. Hinkley. *Theoretical Statistics*. Chapman and Hall, 1979.
- [3] Kostas Daniilidis and Minas E. Spetsakis. *Understanding noise sensitivity in structure from motion*, chapter 4, pages 61–88. Lawrence Erlbaum Associates, 1996.
- [4] Morris H. DeGroot and Mark J. Schervish. *Probability and Statistics*. Addison Wesley, 2001.
- [5] Jr Dennis J.E. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Soc for Industrial & Applied Math, 1996.
- [6] W. Forstner. A feature based correspondence algorithm for image matching. *International Archives of Photogrammetry & Remote Sensing*, 26(3):150–166, 1986.
- [7] Arthur Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.
- [8] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, Manchester, 1988.
- [9] Andrew H Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

- [10] Il-Kyun Jung and Simon Lacroix. Simultaneous localization and mapping with stereovision. In *International Symposium on Robotics Research*, pages 315–324, 2003.
- [11] L. Lucchese and S.K. Mitra. Using saddle points for subpixel feature detection in camera calibration targets. In *Proceedings of the 2002 Asia Pacific Conference on Circuits and Systems*, volume 2, pages 191– 195, 2002.
- [12] L. Matthies and P. Grandjean. Stochastic performance modeling and evaluation of obstacle detectability with imaging range sensors. *IEEE Transactions on Robotics and Automation*, Vol.10(6):783–792, Dec 1994.
- [13] L. Matthies and S. Shafer. Error modelling in stereo navigation. *IEEE Journal of Robotics and Automation*, 3(3):239–248, 1987.
- [14] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. Academic Press, Inc, 1979.
- [15] Clark F. Olson, Larry H. Matthies, Marcel Schoppers, and Mark W. Maimone. Stereo ego-motion improvements for robust rover navigation. In *In Proceedings IEEE Conference on Robotics and Automation (ICRA)*, volume 2, pages 1099– 1104, 2001.
- [16] Amit Roy-Chowdhury and Rama Chellappa. Statistical error propagation in 3d modeling from monocular video. In *2003 Conference on Computer Vision and Pattern Recognition Workshop*, volume 8, pages 89 –, Madison, Wisconsin, June 2003.
- [17] Harold W Sorenson. *Parameter Estimation: Principles and Problems*. Marcel Dekker, Inc., 1980.
- [18] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.
- [19] Y. Yakimovsky and R. Cunningham. A system for extracting three-dimensional measurements from a stereo pair of tv cameras. In *Computer Graphics and Image Processing*, volume 7, pages 1995–2010, 1978.
- [20] G.S. Young and R. Chellappa. Statistical analysis of inherent ambiguities in recovering 3-d motion from a noisy flow field. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 14(10):995–1013, 1992.

Fusion of Stereo, Colour and Contrast

A. Blake, A. Criminisi, G. Cross, V. Kolmogorov, and C. Rother

Microsoft Research Cambridge, 7 JJ Thomson Avenue, Cambridge, UK.
www.research.microsoft.com/vision/cambridge

1 Introduction

Stereo vision has numerous applications in robotics, graphics, inspection and other areas. A prime application, one which has driven work on stereo in our laboratory, is teleconferencing in which the use of a stereo webcam already makes possible various transformations of the video stream. These include digital camera control, insertion of virtual objects, background substitution, and eye-gaze correction [9, 8].

Digital camera control: Here the foreground part of a scene is isolated using stereo, and used to drive the digital pan/zoom/tilt controls of a camera, to keep the subject well framed in the virtual view.

Insertion of virtual objects: Knowing the depth structure of a scene, virtual objects can be inserted live into the video stream, in a way that respects the space occupancy of existing, real objects.

Background substitution: Having isolated the background of a scene using stereo, it can be manipulated — for example blurred, re-colored or replaced entirely, without touching foreground elements. This demands foreground layer separation to near Computer Graphics quality, including α -channel determination as in video-matting [6], but with computational efficiency sufficient to attain live streaming speed.

Eye-gaze correction: A particularly challenging application is to combine video streams from a pair of cameras, stereoscopically, to generate a virtual camera in locations that are inaccessible to a real physical camera. In this way, a virtual camera can be placed in the centre of the screen of each of two computers, so that a pair of subjects in conversation can gaze at one another directly, eye to eye. This problem is particularly hard in practice because the baseline separating the left and right cameras has to be large (fig. 1), resulting in more substantial differences to be resolved between the two images.

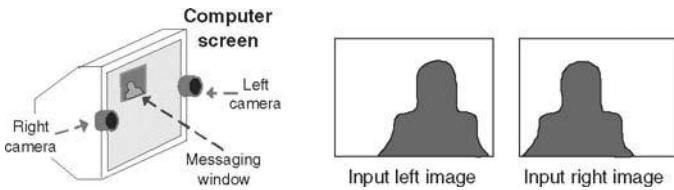


Fig. 1. Stereo cameras for teleconferencing. Two cameras are placed on the frame of a computer monitor. A virtual camera can be synthesised right over the window for viewing the remote participant is (marked in blue) on the middle of the computer screen. Viewing the subject through the virtual camera, rather than the left or right, ensures direct eye contact.

Stereo algorithms have been developed over the past 25 years that are competent at computing depth “densely” — that is at all pixels — in 3D scenes. Earlier algorithms used Dynamic Programming (DP) to compute optimal matches [15, 7] but lately two new algorithms — *Belief Propagation* and *Graph Cut* — have come to head the league table of stereo performance [18]. Stereo “occlusion” is a further cue, arising for those parts of a scene that are visible in one eye (or camera) but not the other. Occlusion needs to be accurately detected, as it is a strong cue for discontinuity of surfaces, and some modern algorithms are capable of doing this [10, 1, 13, 9, 8]. However, some problems remain. In particular, the strength of stereo cues degrades over low-texture regions such as blank walls, sky or saturated image areas. In general, it is difficult to deal with this problem, but in the particular application of stereo to foreground/background segmentation, a powerful remedy is at hand in the form of cue fusion. Recently color and contrast have been shown to be powerful cues for segmentation [4, 17], even in the absence of texture. Segmentation based on color/contrast alone is nonetheless beyond the capability of fully automatic methods. This suggests a robust approach that exploits fusion of a variety of cues for segmentation. Here we propose a model and algorithms for fusion of stereo with color and contrast, and a prior for intra-layer spatial coherence.

2 Probabilistic Models for Stereo Matching

First we outline the probabilistic structure of the stereo matching and color/contrast models. A notation is set out for state variables and observables. Then an energy E or cost-function is defined to characterise well matched images. The energy E also defines a probabilistic model, by acting as the Gibbs energy in a *Conditional Random Field* (CRF) [14].

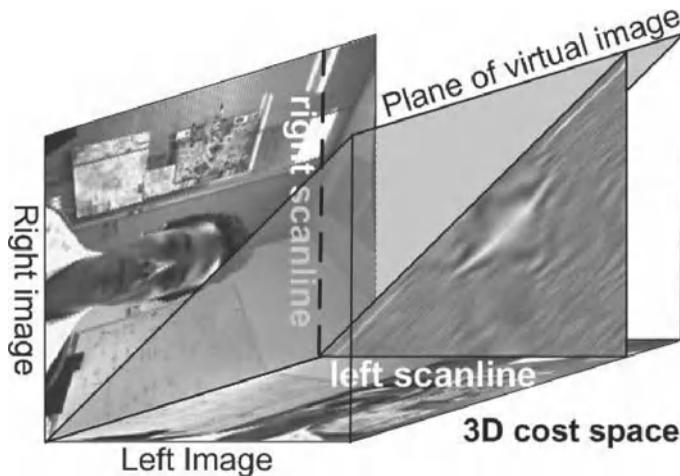


Fig. 2. Matching a pair of rectified stereo images. *Rectification means that pixels in a left scanline all match to pixels in the corresponding right scanline. A match function over a triangular domain is shown that scores the likelihood of all feasible pixel pairs over a particular pair of corresponding epipolar lines — bright means high match likelihood.*

2.1 Notation

Pixels in the rectified left and right images are indexed by m and n respectively, so the images are denoted

$$\mathbf{L} = \{L_m, m = 1, \dots, N\}, \mathbf{R} = \{R_n, n = 1, \dots, N\}.$$

We refer jointly to the data as $\mathbf{z} = (\mathbf{L}, \mathbf{R})$. Rectification is a projective warping transformation applied to left and right images that brings their respective scanlines into direct correspondence. Thus, in rectified images, all pixels on a horizontal (“epipolar”) line in the left image match to pixels in the corresponding epipolar line in the right image. This geometrical normalisation greatly simplifies the complexity of matching pixels. A pair of rectified images is illustrated in figure 2 and the stereo problem is to establish a match between pixels in the left image and corresponding pixels in the right image. Typically, most pixels in each of the images are matched in this way. Those that remain unmatched are the *occluded* pixels, arising for instance where a particular point in the background of a scene is masked by a foreground object in the left view, but visible in the right view.

The mapping between left and right images is expressed in terms of state variables \mathbf{x} and *disparities* \mathbf{d} . The array \mathbf{x} of state variables can be defined symmetrically with respect to left and right image coordinate frames, in so-called *cyclopean* coordinates k . The array then comprises components $\mathbf{x} = \{x_k\}$ which

take values $x_k \in \{M, O\}$, according to whether the pixel is matched or occluded. A further elaboration of the state space, employs values $x_k \in \{F, B, O\}$ according to whether the pixel is a foreground match, a background match or occluded. This subdivision of the scene into foreground and background layers offers the opportunity for imposing further constraints, both prior and driven by data.

Stereo *disparity* is an inverse measure of three-dimensional depth, and is defined to be $d = m - n$. The disparity values along one epipolar line are expressed as $\mathbf{d} = \{d_k, k = 1, \dots, 2N - 1\}$. Note this means that

$$m = \frac{(k + d_k)}{2} \quad \text{and} \quad n = \frac{(k - d_k)}{2}, \quad (1)$$

so that k, d forms the cyclopean coordinate system for the space of epipolar matches, which is symmetric and this is well known to be helpful for probabilistic modeling of stereo matching [1].

This sets up the notation for a complete match of two images as the combined vector (\mathbf{d}, \mathbf{x}) of disparities and states. Now a posterior distribution for (\mathbf{d}, \mathbf{x}) , conditioned on image data, can be defined.

2.2 Generative Model

A Gibbs energy $E(\mathbf{z}, \mathbf{d}, \mathbf{x}; \theta)$ is defined to specify the posterior over the inferred sequence (\mathbf{d}, \mathbf{x}) , given the image data \mathbf{z} , as:

$$p(\mathbf{x}, \mathbf{d} \mid \mathbf{z}) \propto \exp -E(\mathbf{z}, \mathbf{d}, \mathbf{x}; \theta). \quad (2)$$

Here θ is a vector of parameters for the model, which will need to be set according to their relation to physical quantities in the stereo problem, and by learning from labeled data. The posterior could, for instance, be globally maximised to obtain an estimated segmentation $\hat{\mathbf{x}}$ and estimated stereo disparities $\hat{\mathbf{d}}$.

The model (2) can be regarded simply as a Conditional Random Field (CRF) [14], without any generative explanation/decomposition in terms of priors over (\mathbf{x}, \mathbf{d}) and data likelihoods. However, simpler forms of the model do admit a generative decomposition, and this is very helpful also in motivating the structure of a fuller CRF model that is not so naturally decomposed. One reasonable generative model has a Gibbs energy with the following decomposition:

$$E(\mathbf{z}, \mathbf{x}, \mathbf{d}; \theta) = V(\mathbf{x}, \mathbf{d}; \theta) + U^M(\mathbf{z}, \mathbf{x}, \mathbf{d}; \theta) + U^C(\mathbf{z} \mid \mathbf{x}; \theta), \quad (3)$$

in which the role of each of the three terms is as follows.

Prior: an MRF prior for (\mathbf{x}, \mathbf{d}) has an energy specified as a sum of unary and pairwise potentials:

$$V(\mathbf{x}, \mathbf{d}; \theta) = \sum_{(k, k') \in \mathcal{N}} [F(x_k, x_{k'}, \Delta d_k, \Delta d_{k'})] + \sum_k G_k(x_k, d_k), \quad (4)$$

where $\Delta \mathbf{d}$ is the *disparity gradient* along epipolar lines, that is

$$\Delta d_k = d_k - d_{k-1}. \quad (5)$$

Typically, $F(\dots)$ discourages excessive disparity gradient within matched regions. Pixel pairs $(k, k') \in \mathcal{N}$ are the ones that are deemed to be neighbouring in the pixel grid. The first component $F(\dots)$ of the prior Gibbs energy V in (4) should incorporate an Ising component that favours coherence in the segmentation variables $x_k, x_{k'}$. It should also favour continuity of disparity over matched regions, and do so anisotropically — more strongly along epipolar lines than across them.

Optionally, when the extended state-space $x_k \in \{\text{F}, \text{B}, \text{O}\}$ is used, the $G_k(\dots)$ term is included to implement “disparity-pull”, the tendency of foreground elements to have higher disparity than background ones. The specific form of $G_k(\dots)$ can be set by taking

$$G_k(x_k, d_k) = -\log p(d_k | x_k), \quad (6)$$

and determining the conditional density $p(d_k | x_k)$ from the observed statistics of some labelled data. Various models could be used here, but in our experiments a simple, constant disparity, separating surface is used, so that $d > d_0$ characterises foreground, with uniform distributions for $p(d_k | x_k)$ over each of the possible states $x \in \{\text{F}, \text{B}, \text{O}\}$.

Stereo likelihood, represented by the U^M term, evaluates the stereo-match evidence in the data \mathbf{z} , both to distinguish occlusion ($x_k = \text{O}$) from full visibility ($x_k \in \{\text{F}, \text{B}\}$) and, given visibility, to determine disparity d_k .

Color likelihood, represented by the U^C term, uses probability densities in colour space, one density for the background and another for the foreground, to apply evidence from pixel colour to the segmentation x_k of each pixel. This term is optional, used only with the extended state-space $x_k \in \{\text{F}, \text{B}, \text{O}\}$.

2.3 Contrast Dependence

One further elaboration, due to Boykov and Jolly [4], incorporates the evidence from image contrast for segmentation — see also ‘line processes’ [11], ‘weak constraints’ [3] and ‘anisotropic diffusion’ [16]. It proves important in refining segmentation quality, at the cost of obscuring somewhat the clear generative distinction between prior and likelihood [2]. The Ising component F in (4) is made contrast dependent, disabling the penalty for breaking coherence in \mathbf{x} wherever image contrast is high. Segmentation boundaries tend, as a result, to align with contours of high contrast. The MRF model (3) is extended in this way to a CRF

$$E(\mathbf{z}, \mathbf{x}, \mathbf{d}; \theta) = V(\mathbf{z}, \mathbf{x}, \mathbf{d}; \theta) + U^M(\mathbf{z}, \mathbf{x}, \mathbf{d}; \theta) + U^C(\mathbf{z} \mid \mathbf{x}; \theta), \quad (7)$$

in which dependence on data \mathbf{z} is now incorporated in to the $V(\dots)$ term, so that this no longer represents a pure prior distribution.

3 Inference

Two inference problems are considered for the model of the previous section. The first is the full inference of disparity \mathbf{d} and state \mathbf{x} , necessary when the three-dimensional structure of a scene is required explicitly. This would be the case in many robotics applications, and for the gaze-correction function in the teleconferencing application described in section 1. The other three teleconferencing applications however, require only segmentation — the inference of \mathbf{x} but not of \mathbf{d} .

3.1 Inferring Disparity

To compute both disparity and state, the posterior is maximised with respect to \mathbf{d} and \mathbf{x} :

$$(\hat{\mathbf{x}}, \hat{\mathbf{d}}) = \arg \max_{\mathbf{x}, \mathbf{d}} p(\mathbf{x}, \mathbf{d} \mid \mathbf{z}). \quad (8)$$

Here we take the short form of the state vector $x_k \in \{M, O\}$, and use only stereo cues, without colour. This problem is not formally tractable but could be regarded as tractable in practice because it can be solved approximately by the α -expansion form of graph-cut [5], over the variables \mathbf{x}, \mathbf{d} jointly (provided the energy function E is chosen to meet the necessary regularity conditions). In practice α -expansion over (\mathbf{x}, \mathbf{d}) jointly is computationally burdensome, one or two orders of magnitude slower than real-time, for a conventional video stream on a current single processor architecture. A faster solution can be computed by neglecting vertical constraints in the model. All vertical cliques in V (4) are removed, resulting in a posterior density consisting simply of a set of one-dimensional Hidden Markov Models (HMMs), one HMM along each epipolar line. For the coherence encouraged by V , constraints can be imposed only horizontally, and the vertical constraint is lost. Nonetheless there is some implicit transfer of information vertically via the overlap of the patches used in the stereo match likelihood [8]. In exchange for the lost vertical constraint, the problem becomes exactly tractable by dynamic programming and DP can be performed along scanlines, jointly with respect to disparities and state variables [9, 8]. This can be achieved in real time.

3.2 Inferring Segmentation

An alternative aim to computing full disparity and state, is to compute only the state, and this useful with the extended state $x_k \in \{F, B, O\}$, so that

the image is segmented into foreground, background and occluded regions. Then colour distributions, associated with background and foreground, can be brought into the model. For this problem, the posterior should, in principle, be marginalised with respect to \mathbf{d} , and then maximised with respect to \mathbf{x} to estimate a segmentation:

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} \sum_{\mathbf{d}} p(\mathbf{x}, \mathbf{d} \mid \mathbf{z}). \quad (9)$$

This problem is intractable with the Gibbs energy model (7) above. This paper proposes two approaches to simplifying the Gibbs energy model, to make inference of segmentation \mathbf{x} practically tractable and efficient.

LDP. In Layered Dynamic Programming [12], vertical constraints are neglected as above for full stereo. The marginalised form of the problem (9) is not tractable even without vertical constraints, so it is necessary to stick with the full problem (8), and simply discard the unwanted disparities. This is not ideal because, in principle, statistical information is wasted on the computation of disparities.

The difference then between LDP and full DP stereo is simply that the extended state $x_k \in \{F, B, O\}$ is used, with appropriate energy terms to represent foreground and background constraints, both prior and from data, in terms of colour properties of the foreground and background layers of the scene.

LGC. In Layered Graph Cut [12], the prior term $F(\dots)$ in (4) is made independent of disparity \mathbf{d} . Now the posterior density can be marginalised exactly over \mathbf{d} in the original inference problem (9). Marginalization gives the posterior density $p(\mathbf{x} \mid \mathbf{z})$ for segmentation only, which can be maximised by graph-cut with α -expansion. Parameter learning has not been made tractable, but some guidance comes from priors and likelihoods estimated for LDP, transplanted (and simplified) to the LGC model.

In summary, we have two approximate models for the original problem. One, LDP, has the advantage of practical tractability not only for inference but in fact also for parameter learning [12]. It has the disadvantage though that vertical constraints have been neglected. On the other hand LGC retains vertical constraints at least for segmentation, but neglects all direct constraints on continuity of disparity. It has the advantage of solving the original max-sum form of the inference problem, rather than just the max-max approximation, but the disadvantage that parameter estimation remains intractable. In terms of practical efficiency and efficacy, the two algorithms, LDP and LGC, perform remarkably similarly [12], despite having very different structures.

4 Some Results from Stereo Matching and Segmentation

Results of full stereo matching, used to generate virtual camera views, are illustrated in figure 3. The top line show real left and right cameras, used

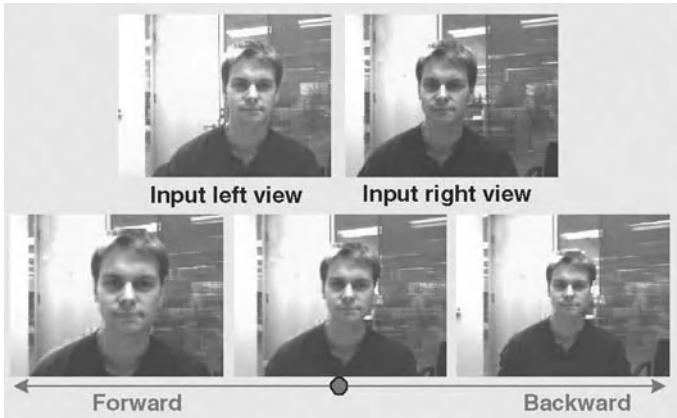


Fig. 3. Virtual camera.

as input to stereo matching. Computed disparities are then used, either implicitly or explicitly, to recover the shape of the scene. The scene can then be reprojected onto the image plane of a virtual camera. In the bottom centre, an interpolated cyclopean view is shown, of the sort that can be used for gaze correction — the subject is looking directly forwards in this view. It is critically important for the quality of the virtual image, that not only disparity but also the occlusion information in \mathbf{x} is available [9, 8]. Bottom left and right images in the figure show the views when the virtual camera is moved respectively backward and forward in space.

Results of stereo segmentation, fusing stereo, colour and contrast, are shown in figure 4. Left and right images are processed using the LGC algorithm above, to separate the foreground subject from its background [12]. The extracted foreground can then be applied to a new background and this is illustrated in the figure for three frames of a test video. Special measures — so-called *border matting* [17] — are taken so that the extracted foreground sequence can be composited, free of “aliasing” artefacts, onto the background. Border matting deals with mixed pixels — pixels that contain both foreground and background colour, occurring typically around the boundary of an object. If this is neglected, discolouration occurs around boundaries, where traces of the original background colour remain stuck to the foreground subject.

The paper has made a rapid tour of some recent progress in algorithms for stereo vision. Highlights include: a probabilistic framework; the full treatment of occlusion via an appropriate representation of state; fusion of cues, specifically stereo, colour and contrast. Many details have been omitted in this account, and the reader is directed to [8, 12] for full details.

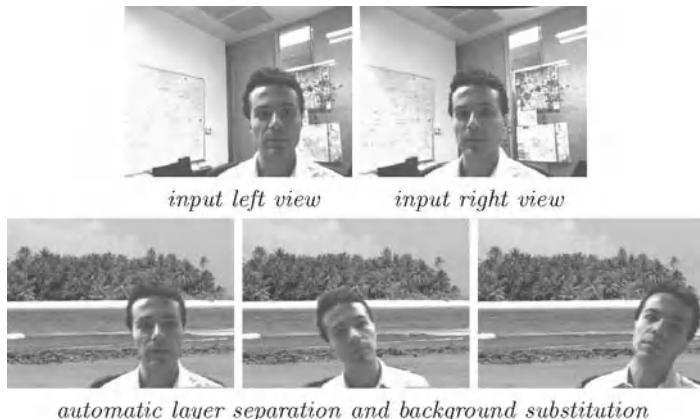


Fig. 4. Background substitution.

References

1. P.N. Belhumeur. A Bayesian approach to binocular stereopsis. *Int. J. Computer Vision*, 19(3):237–260, 1996.
2. A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *Proc. European Conf. Computer Vision*, pages 428–441. Springer-Verlag, 2004.
3. A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, USA, 1987.
4. Y.Y. Boykov and M-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proc. Int. Conf. on Computer Vision*, pages 105–112, 2001.
5. Y.Y. Boykov, O. Veksler, and R.D. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(11), 2001.
6. Y-Y Chuang, B. Curless, D.H. Salesin, and R. Szeliski. A Bayesian approach to digital matting. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages CD-ROM, 2001.
7. I.J. Cox, S.L. Hingorani, and S.B. Rao. A maximum likelihood stereo algorithm. *Computer vision and image understanding*, 63(3):542–567, 1996.
8. A. Criminisi, J. Shotton, A. Blake, and P.H.S. Torr. Efficient dense stereo and novel view synthesis for gaze manipulation in one-to-one teleconferencing. Technical Report MSR-TR-2003-59, Microsoft Research Cambridge, 2003.
9. A. Criminisi, J. Shotton, A. Blake, and P.H.S. Torr. Gaze manipulation for one to one teleconferencing. In *Proc. Int. Conf. on Computer Vision*, pages 191–198, 2003.
10. D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *Int. J. Computer Vision*, 14:211–226, 1995.
11. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

12. V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Bi-layer segmentation of binocular stereo video. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2005.
13. V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *Proc. European Conf. Computer Vision*, pages 82–96, 2002.
14. J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proc. Int. Conf. Machine Learning*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
15. Y. Ohta and T. Kanade. Stereo by intra- and inter-scan line search using dynamic programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.
16. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.
17. C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.
18. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, 47(1–3):7–42, 2002.

Automatic Single-Image 3d Reconstructions of Indoor Manhattan World Scenes

Erick Delage, Honglak Lee, and Andrew Y. Ng

Stanford University, Stanford, CA 94305 {edelage,hllee,ang}@cs.stanford.edu

Summary. 3d reconstruction from a single image is inherently an ambiguous problem. Yet when we look at a picture, we can often infer 3d information about the scene. Humans perform single-image 3d reconstructions by using a variety of single-image depth cues, for example, by recognizing objects and surfaces, and reasoning about how these surfaces are connected to each other. In this paper, we focus on the problem of automatic 3d reconstruction of indoor scenes, specifically ones (sometimes called “Manhattan worlds”) that consist mainly of orthogonal planes. We use a Markov random field (MRF) model to identify the different planes and edges in the scene, as well as their orientations. Then, an iterative optimization algorithm is applied to infer the most probable position of all the planes, and thereby obtain a 3d reconstruction. Our approach is fully automatic—given an input image, no human intervention is necessary to obtain an approximate 3d reconstruction.

1 Introduction

When viewing a single image such as that in Figure 1, most humans have little trouble estimating the 3d shape of the scene. Given only a single image, depths are inherently ambiguous, and thus 3d reconstruction cannot be achieved using naive, geometry-only approaches such as a straightforward implementation of stereopsis (binocular vision). In this paper, we consider the task of monocular (single camera) 3d reconstruction, specifically of indoor scenes consisting mainly of orthogonal planes. Our motivation for studying the monocular 3d reconstruction problem is two-fold. First, although one may envision systems that use both monocular and binocular cues, as a scientific endeavor we find it most enlightening to focus exclusively on monocular vision; specifically, this allows us to try to elucidate how monocular cues—which have heretofore been little-exploited in automatic 3d reconstructions—can be used. Second, we consider monocular 3d reconstruction to be interesting and important in its own right. For example, unlike stereo vision, it works well even at large distances

(if, say, the images are taken through a zoom lens). In contrast, stereo vision is fundamentally limited by the baseline distance between the two cameras, and performs poorly when used to estimate depths at ranges that are very large relative to the baseline distance.



Fig. 1. Single camera image of a corridor.

Apart from stereopsis, there are many other algorithms that use multiple images to estimate depths, such as structure from motion [23] and shape from defocus [8]. These methods suffer from similar problems to stereopsis when estimating depths at large ranges. A number of researchers have attempted to recover 3d information from a single image. Shape from shading [25] is one well-known approach, but is not applicable to richly structured/textured images such as that in Figure 1. For such indoor images, methods based on “3d metrology” hold some promise. Given sufficient human labeling/human-specified constraints, efficient techniques can be used to generate a 3d reconstruction of the scene. [5, 6, 21, 22] However, these methods tend to require a significant amount of human input (for example, specifying the correspondences between lines in the image and the edges of a reference model), and are thus limited in their applicability.

Recent work strongly suggests that 3d information can be efficiently recovered using Bayesian methods that combine visual cues with some prior knowledge about the geometry of a scene. For example, Kosaka and Kak [13] give a navigation algorithm that allows a monocular robot to track its position in a building by associating visual cues, such as lines and corners, with the configuration of hallways on a floor plan. However, this approach would fail in a new environment in which such a floor plan is not available beforehand. A more flexible algorithm, due to Han and Zhu [11], used models both of man-made “block-shaped objects” and of some natural objects, such as trees and grass. Unfortunately, this approach has so far been applied only to fairly sim-

ple images, and seems unlikely to scale in its present form to complex, textured images as shown in Figure 1. Saxena, Chung and Ng [19] apply an MRF to directly estimating depths from a monocular image, focusing mainly on unstructured (for example, outdoor) scenes. (See also [18].) The “Manhattan world” assumption [3, 4] (i.e., that the environment contains only orthogonal planes, as in many urban environments) has been used to develop algorithms for estimating camera calibration parameters [20] and camera pose [3, 4] from complex images. In this paper, we exploit this same assumption to obtain single-image 3d reconstructions.

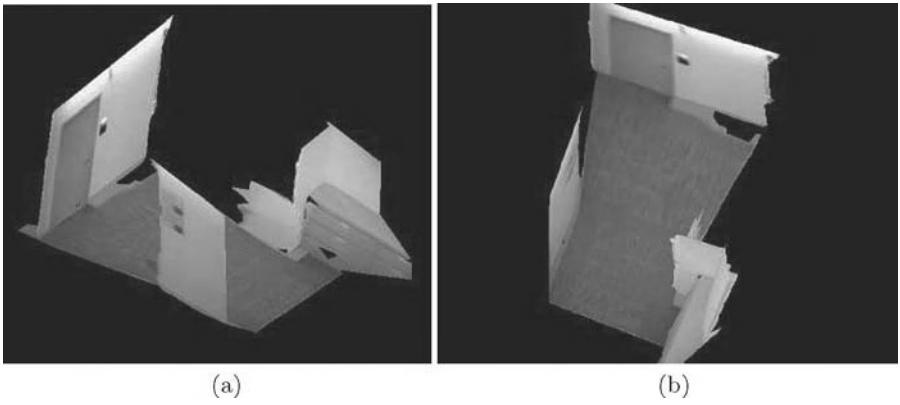


Fig. 2. 3d reconstruction of a corridor from single image presented in figure 1.

Our approach uses a Markov random field (MRF) to estimate whether each point in an image represents a surface or an edge, and also the orientation of the surface or edge. Using this information, we then use an iterative algorithm to try to infer the 3d reconstruction of the scene. Figure 2 shows an example of our algorithm’s output, generated fully automatically from the image in Figure 1. To our knowledge, our work represents the first fully automatic algorithm for 3d reconstruction from single indoor images.

The remainder of this paper is structured as follows. In Section 2, we describe the basic geometric calculations used by our algorithms. Section 3 presents the MRF model; and Section 4 then describes how we compute a 3d reconstruction from the MRF’s output. In Section 5, we present experimental results.

2 Preliminaries

We make the following assumptions:

1. The image is obtained by perspective projection, using a calibrated camera¹ with calibration matrix K . Thus, as presented in Figure 3, a point \mathbf{Q} in the 3d world is projected to pixel coordinate \mathbf{q} (represented in homogeneous coordinates) in the image if and only if:²

$$\mathbf{Q} \propto K^{-1}\mathbf{q}. \quad (1)$$

2. The objects in the image are composed of planes in each of three mutually orthogonal orientations. Thus, the image also contains three vanishing points corresponding to three different directions (one of them orthogonal to the floor plane).³
3. The camera's vertical axis is orthogonal to the floor plane, and the floor is in the lower part of the image.(Figure 3)⁴
4. The camera center (origin of the coordinate frame) is at a known height above the ground.⁵

Assumption 2 is often called the Manhattan world assumption [3].

In an image that has no occluding edges, the assumptions above are sufficient to ensure that the full 3d geometry of a scene is exactly specified, given only a segmentation of the scene into surfaces (together with labels indicating the surfaces' orientations). Thus, knowledge of the segmentation and orientations is sufficient to unambiguously reconstruct the 3d location of every pixel in the image. This result is a completely straightforward consequence of perspective geometry. Still assuming the absence of occluding edges, we now describe how this 3d reconstruction can be obtained.

First, by perspective projection, the 3d location \mathbf{Q}_i of a pixel at position \mathbf{q}_i in the image plane must satisfy:

$$\mathbf{Q}_i = \lambda_i K^{-1}\mathbf{q}_i \quad (2)$$

¹ A calibrated camera means that the orientation of each pixel relative to the optical axis is known.

² Here, K , \mathbf{q} and \mathbf{Q} are as follows:

$$K = \begin{bmatrix} f & 0 & \Delta_u \\ 0 & f & \Delta_v \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Thus, \mathbf{Q} is projected onto a point \mathbf{q} in the image plane if and only if there is some constant λ so that $\mathbf{Q} = \lambda K^{-1}\mathbf{q}$.

³ Vanishing points in the image plane are the points where lines that are parallel in the 3d space meet in the image. In a scene that has mainly orthogonal planes—such as in many indoor scenes—most edges (in the 3d world) will lie in one of three possible directions, and thus there will be three vanishing points in the image.

⁴ Small misalignments of the camera's vertical axis can also be easily compensated for (e.g., see [3, 4]).

⁵ If the height of the camera is unknown, then the 3d reconstruction will be determined only up to an unknown scaling factor.

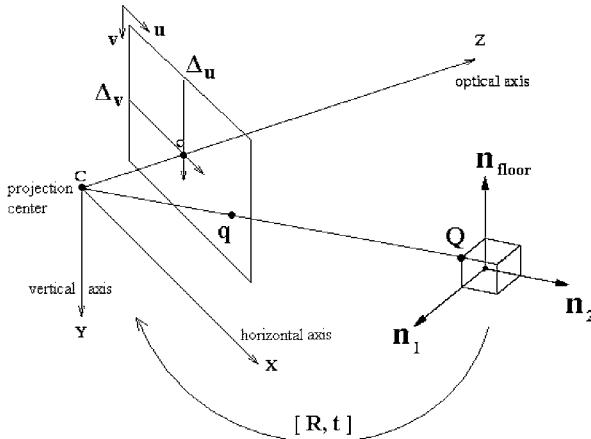


Fig. 3. Coordinate system used by algorithm.

for some λ_i . Thus, \mathbf{Q}_i is restricted to a specific line that passes through the origin of the camera. Further, if this point lies on some plane p that has normal vector \mathbf{n}_p , then we have

$$\mathbf{n}_p \cdot \mathbf{Q}_i = \lambda_i \mathbf{n}_p \cdot (K^{-1}\mathbf{q}_i) = d_p, \quad (3)$$

where d_p is the distance of the plane from the camera center (the origin of the 3d coordinate frame). Thus, λ_i can be exactly determined given only d_p ; and therefore estimating the position of every pixel in the image reduces to the problem of finding d_p for all planes p .

Since we assumed that there are no occluding edges, every two adjacent pixels in the image are also physically adjacent in 3d.⁶ Since each point \mathbf{q}_i (with variable λ_i) is part of some plane, each variable λ_i is constrained by at least one equation of the form in Equation (3). Moreover, if there are no occluding edges in the image, then the points lying on the boundary of two adjacent/connected planes participate in two different constraints (one for each of the two neighboring planes). By incorporating assumption 4, we also know the distance d_p from the floor plane to the camera. Except in degenerate cases, this is sufficient to ensure that, treating the λ_i and d_p as variables, the system of equations given in Equation (3) are sufficiently constrained to have a unique solution.

The process described above required knowledge of the segmentation of the scene into planes as well as knowledge of the orientation of the planes. In Section 3, we describe an algorithm for estimating these quantities. Furthermore, the assumption that there are no occluding edges will often fail to hold

⁶ Section 4 will address the case of occluding edges.

in indoor scenes; in Section 4, we describe a reconstruction algorithm that applies even in the presence of occluding edges.

3 Markov Random Field Model

Given an image of a scene comprising planes in three mutually orthogonal directions, there are standard algorithms for recovering the three vanishing points in the image. (E.g., [17, 20]) We use [20] to identify these vanishing points; by doing so, we also identify the three possible orientations for the planes $\mathbf{n}_{\text{floor}}$, \mathbf{n}_1 , and \mathbf{n}_2 (one orthogonal to each of the vanishing point directions).

In our Manhattan world, the edges (boundaries) of a plane cannot be oriented in the same direction as its normal. If there is no occlusion, this gives us a constraint on the possible directions for the edges of a surface. (For example, the floor should not be bordered by edges that point upwards in the direction $\mathbf{n}_{\text{floor}}$). Our MRF model will incorporate this constraint.

Our MRF is structured as a 320*240 grid (each node corresponding to a different position in the image). Each node corresponds to a random variable that takes on one of 6 possible values, that indicate whether the node is on a line pointing toward one of the three vanishing points (labels e_1 , e_2 , e_3), or whether it lies on a plane whose normal is oriented in one of the three orthogonal directions (labels p_1 , p_2 , p_3). Figure 4 shows the 6 labels. The MRF models the joint probability distribution of this 320*240 grid of label values; and will be used to infer the most likely set of labels given a new image.

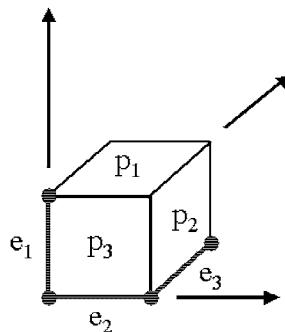


Fig. 4. The 6 possible labels for the MRF nodes (points in the 2d image).

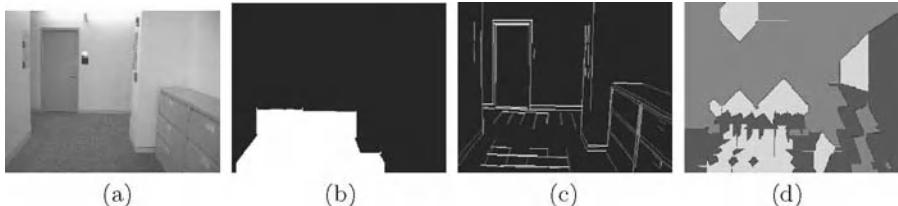


Fig. 5. Example of features extracted from an image. (a) The input image. (b) A mask identifying the floor pixels. (c) Lines extracted using [16] and classified according to their orientations. (d) Labeling of each pixel with the direction of the edge in (c) which is closest in the same row or column.

3.1 MRF Features

This section briefly describes the image features used in our MRF model.

Edge Statistics Features

Statistics about edges were computed using the Canny edge detector [2], the phase congruence [15], and Sobel edge filter [10]. Using the orientation of intensity gradients, we also determined for each location the most likely vanishing point of each edge. Line extraction algorithms from [14] and [16] were used to obtain a list of lines in the image (generating two different sets of features). Each line was also identified according to its vanishing point.⁷ We also created additional features based on the nearby edges' orientations.⁸

Segmentation-Based Features

Surfaces often have fairly uniform appearances in texture and color, and thus image segmentation algorithms provide another set of useful features. Specifically, pixels that are members of the same segmented group should usually be labeled with the same orientation. We used a graph-based segmentation algorithm [9] to generate a partition of the image, and assigned a unique identifier to each partition output by the segmentation algorithm. For each pair of adjacent nodes in the grid, we also generated a pairwise/relational feature in our MRF model indicating whether the nodes were members of the same partition of the image.

⁷ Lines which diverged from all three vanishing points were discarded. Some lines whose 3d orientations were ambiguous were assigned to two vanishing points.

⁸ At a given position in the image, we add an extra feature corresponding to the orientation of the closest line (measured either in the same row or column) in the image. (See Figure 5d.) We also created additional features corresponding to the second and third closest lines.

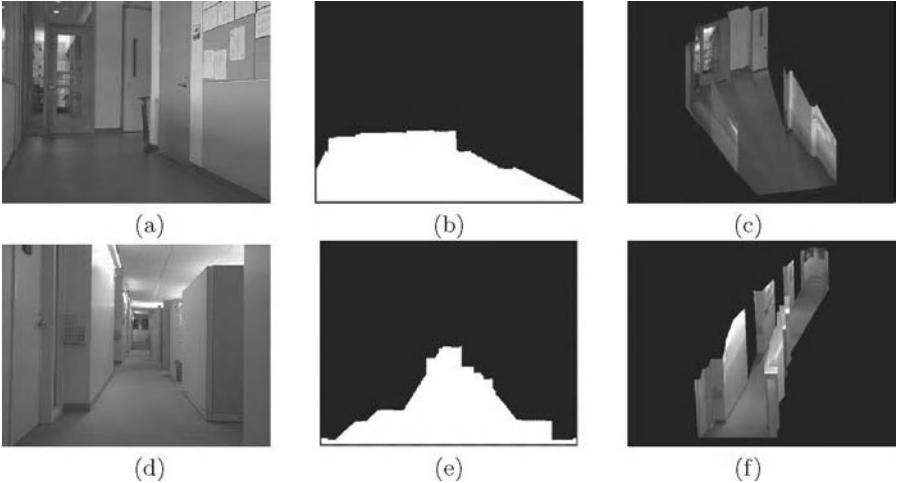


Fig. 6. Results from DBN floor segmentation algorithm of [7]. (a),(d) original image. (b),(e) floor mask. (c),(f) 3d reconstruction (obtained assuming presence only of floor and walls in image).

Floor Segmentation Features

Since many planes (e.g., most walls) are connected to the floor, correct labeling of the floor plane plays an important role in 3d reconstruction. Building on our earlier work [7], we used a dynamic Bayesian network (DBN) to identify the floor boundary in the image plane. Our DBN is a probabilistic model that incorporates a number of local image features, and tries to reason about the chroma of the floor, the position of the floor boundary in each column of the image, and the local direction of the floor boundary. The DBN output is then used to generate a “floor mask” feature indicating whether each pixel was identified as part of the floor.⁹ (See Figure 5b.)

In [7], it was shown that if the image contains only the floor and vertical walls, then (under mild assumptions) knowledge of this floor boundary is sufficient to give a complete 3d reconstruction of the scene. The basic idea is that, given the camera height and orientation, every point in the ground plane can be reconstructed exactly. Then, because the position of each point on the boundary between the floor and each wall is now known (because these points also comprise part of the ground plane), we also now know the 3d position of the lower-edge of each wall. This is sufficient to exactly reconstruct the position of each wall. Figure 6 shows some examples of results obtained using this procedure. We note, however, that this procedure does not apply to scenes

⁹ Two additional features were created using the DBN output: one to identify edges of the floor boundary; the other to identify sharp changes in direction of the floor boundary (which are often indicative of a transition between two wall planes).

that have other orthogonal surfaces (e.g., the top surfaces of desks and filing cabinets), such as in the Manhattan worlds considered in the present paper.

3.2 MRF Parameterization

As discussed previously, each node can take on one of 6 possible label values: 3 for plane orientations (labels p_1, p_2, p_3) and 3 for edge orientations (labels e_1, e_2, e_3).¹⁰ We used a grid-structured Markov random field. Figure 7 shows the structure of the MRF. We use V to denote the set of nodes in the model, and E to denote the edges. Let $y_v \in \{p_1, p_2, p_3, e_1, e_2, e_3\}$ denote the value associated with vertex $v \in V$, and let x_v denote the vector of features computed at position v in the image (and similarly $x_{u,v}$ be computed from positions u and v). The MRF defines a joint probability distribution over all label assignments y :

$$P_\theta(y|x) = \frac{1}{Z_\theta(x)} \exp \left(- \sum_{v \in V} \Psi_1(y_v, x_v; \theta_1) - \sum_{(u,v) \in E} \Psi_2(y_u, y_v, x_{u,v}; \theta_2) \right). \quad (4)$$

Here, Ψ_1 is the potential function for individual nodes, Ψ_2 gives the pairwise potentials in the MRF, $\theta = [\theta_1, \theta_2]$ are the parameters of the model, and $Z_\theta(x)$ is the partition function.

Using the features described in Section 3.1, we chose $\Psi_1(y_v, x_v; \theta_1)$ to be a weighted linear combination of features indicative of the label at a vertex v :¹¹

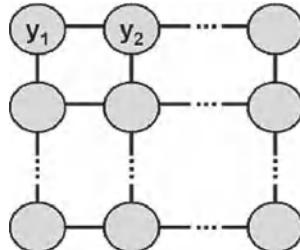


Fig. 7. Markov random field model over the image.

¹⁰ A plane with orientation p_i has a normal in direction e_i . Thus, a plane with orientation p_1 would typically be bordered by edges of type e_2 and e_3 .

¹¹ For example, given a specific edge-based feature $C_1(v, x_v)$ from Section 3.1 (one that is indicative of whether an edge at position v heads towards e_1), we create the following MRF features:

$$\Psi_1(y_v, x_v; \theta_1) = \theta_1^\top \cdot \Phi(y_v, x_v). \quad (5)$$

Similarly, we used

$$\Psi_2(y_u, y_v, x_{u,v}; \theta_2) = \theta_2^\top \cdot \Phi(y_u, y_v, x_{u,v}), \quad (6)$$

where $\Phi(y_u, y_v, x_{u,v})$ were chosen to be features indicative of whether y_u and y_v are likely to be the same label (e.g., the segmentation-based feature of Section 3.1). We also included features in the pairwise potential that measure “consistency” between the plane and the edge orientations.¹² For example, these features can be used to help capture the fact (discussed earlier) that a plane with normal p_i is unlikely to be bordered by edges of orientation e_i .

Putting all the features together, $\Phi(y_v, x_v)$ was a 75 dimension vector, and $\Phi(y_u, y_v, x_{u,v})$ was a 9 dimension vector.

3.3 Training and Inference

In order to train the model parameters θ_1 and θ_2 , we hand-labeled two images with their ground-truth labels y . This set of two images made up our training set. Unfortunately, maximum likelihood parameter learning is intractable in grid-structured MRF models; thus we learned the parameters using an objective similar to pseudo-likelihood.¹³ [1]

$$\begin{aligned} \Phi_1(y_v, x_v) &= C_1(v, x_v) \times 1\{y_v = e_1\} \\ \Phi_2(y_v, x_v) &= C_1(v, x_v) \times 1\{(y_v = e_2) \vee (y_v = e_3)\} \\ \Phi_3(y_v, x_v) &= C_1(v, x_v) \times 1\{(y_v \neq e_1) \wedge (y_v \neq e_2) \wedge (y_v \neq e_3)\}, \end{aligned}$$

¹² For example:

$$\begin{aligned} \Phi_1(y_u, y_v, x_{u,v}) &= 1\{y_u = \text{plane} \wedge y_v = y_u\} \\ \Phi_2(y_u, y_v, x_{u,v}) &= 1\{y_u = \text{plane} \wedge y_v = \text{plane} \wedge y_u \neq y_v\} \\ \Phi_3(y_u, y_v, x_{u,v}) &= 1\{y_u = \text{edge} \wedge y_v = \text{edge}\} \\ \Phi_4(y_u, y_v, x_{u,v}) &= \sum_{i=1}^3 1\{y_u = p_i \wedge y_u = e_i\} \\ \Phi_5(y_u, y_v, x_{u,v}) &= \sum_{i=1}^3 1\{y_u = p_i \wedge y_v = \text{edge} \wedge y_u \neq e_i\} \end{aligned}$$

¹³ In our experiments, straightforward pseudo-likelihood (or generalized pseudo-likelihood [12] using small clusters of nodes) did not work well. Our parameters were actually learned using a product approximation over 3-node networks. More formally, we used:

$$\max_{\theta} \prod_{(u,v,w) \in \mathbf{F}} \hat{P}_{\theta}(y_u, y_v, y_w | x),$$

Finally, after learning the parameters, the inference task in our Markov random field is to compute the most likely set of labelings, given a feature vector x from a new image:

$$\hat{y} = \arg \max_y P_\theta(y|x), \quad (7)$$

Exact inference in a grid-structured MRF is intractable. We approximated this using the algorithm of Wainwright et al. [24].

4 Using the MRF Output for 3d Reconstruction

We now address the problem of 3d reconstruction given an image in which the planes have been segmented and labeled with their orientations, for example by our MRF. Sturm and Maybank [22] proposed an algorithm for a similar problem, and demonstrated good 3d reconstruction given human-labeled images. However, their algorithm is not directly applicable to an image labeled by our MRF, as it requires that occlusion vs. non-occlusion edges be labeled (i.e., labels indicating whether two adjacent planes in the image are physically connected in 3d). This is difficult to infer from local image features, and is not part of the information output by our MRF. Their algorithm has also been tested only on instances with perfectly correct human-generated labels. We now present an algorithm, a modification and generalization of Sturm and Maybank’s algorithm, for 3d reconstruction from an image given possibly noisy labels of the planes and edges.

If we examine an individual “edge” point \mathbf{q}_i that is on the boundary between two planes p and p' , this point can either be part of an occluding edge between the two planes or part of an edge that physically connects the two planes p and p' . Therefore, in the latter case we would want to find a 3d reconstruction where the following distance is small:

$$\Delta_{i,p,p'} = \|\mathbf{Q}_{i,p} - \mathbf{Q}_{i,p'}\|_2.$$

Here, $\mathbf{Q}_{i,p}$ (respectively $\mathbf{Q}_{i,p'}$) is the 3d position in the plane of p (respectively p') that would appear at position \mathbf{q}_i in the image plane. Thus, we can informally think of $\Delta_{i,p,p'}$ as the distance between (two specific points on) the planes p and p' .

Thus argument above applies if an edge is known to be non-occluding. However, it is usually not obvious if an edge is indeed occluding, and thus

where

$$\hat{P}_\theta(y_u, y_v, y_w | x) = \frac{1}{\hat{Z}_\theta(x)} \exp \left(- \sum_{i \in \{u, v, w\}} \Psi_1(y_i, x_i; \theta_1) - \sum_{(i,j) \in \{(u,v), (v,w)\}} \Psi_2(y_i, y_j, x_{i,j}; \theta_2) \right).$$

Above, \mathbf{F} is set of randomly sampled regions of three connected vertices.

occlusion vs. non-occlusion must be inferred. We model the distance $\Delta_{i,p,p'}$ using a Laplacian probability distribution parameterized by $\alpha_{p,p'}$:

$$P_{\alpha_{p,p'}}(\Delta_{i,p,p'}) = \alpha_{p,p'} \exp(-\alpha_{p,p'} \Delta_{i,p,p'}), \quad \forall i \in R_{p,p'}, \quad (8)$$

where $R_{p,p'}$ is the set of (indices of) points that are on the boundary between the planes p and p' .

To form a 3d reconstruction, we will try to maximize the log-likelihood of d , λ , \mathbf{Q} and α , given the MRF labeling of the planes and edges. More formally, we have:

$$\begin{aligned} & \text{maximize}_{d,\lambda,\mathbf{Q},\alpha} \sum_{(p,p') \in B} \sum_{i \in R_{p,p'}} \log P_{\alpha_{p,p'}}(\|\mathbf{Q}_{i,p} - \mathbf{Q}_{i,p'}\|_2) \\ & \text{subject to} \quad \mathbf{Q}_{i,p} = K^{-1} \mathbf{q}_i \lambda_{i,p}, \quad \forall (i,p) \\ & \quad d_p = \mathbf{n}_p^T K^{-1} \mathbf{q}_i \lambda_{i,p}, \quad \forall (i,p) \\ & \quad d_{\text{floor}} = c, \end{aligned} \quad (9)$$

where B is the set of pairs (p,p') of planes that share a common boundary in the image.

We apply an efficient alternating maximization algorithm to this optimization problem. For fixed α , maximizing the objective over d , λ and \mathbf{Q} reduces to a linear program:

$$\begin{aligned} & \text{minimize}_{d,\lambda} \sum_{(p,p') \in B} \sum_{i \in R_{p,p'}} w_{i,p,p'} |\lambda_{i,p} - \lambda_{i,p'}| \\ & \text{subject to} \quad d_p = \mathbf{n}_p^T K^{-1} \mathbf{q}_i \lambda_{i,p}, \quad \forall (i,p) \\ & \quad d_{\text{floor}} = c, \end{aligned} \quad (10)$$

where $w_{i,p,p'} = \alpha_{p,p'} \|K^{-1} \mathbf{q}_i\|_2$. For fixed d , λ and \mathbf{Q} , we can maximize over α in closed form:¹⁴

$$\alpha_{i,j} = \frac{\sum_{i \in R_{p,p'}} 1}{\sum_{i \in R_{p,p'}} \|\mathbf{Q}_{i,p} - \mathbf{Q}_{i,p'}\|_2}. \quad (11)$$

We iterate updating d , λ and \mathbf{Q} ; and updating α , until convergence.¹⁵

Sturm and Maybank's method—which relied on known occlusion edges—can roughly be viewed as a variant of our algorithm in which a Gaussian

¹⁴ Using a heuristic reminiscent of Laplace smoothing, we actually add 0.5 to the denominator, and 5 to the numerator. This smooths the estimates, and also prevents a small denominator from causing $\alpha_{p,p'}$ from growing without bound. To help the search procedure, we also used a heuristic in which $\alpha_{\text{floor},p'}$ (and $\alpha_{p,p'}$ for horizontal edges) were initialized to be large. Edges that appeared clearly to be occluding, specifically ones parallel to the normal of a plane, were discarded from the optimization (or, less formally, had $\alpha_{p,p'}$ set to an infinitesimally small value).

¹⁵ Other details: During the reconstruction, we also discard the ceiling plane. Also, all planes that were reconstructed as lying outside a reasonable range (a 10m \times 10m \times 50 m box in front of the camera) were considered outliers, and also discarded.

(instead of Laplacian) model with a fixed variance parameter (rather than the variable α) is used. [19] found Laplacians to be a superior model than Gaussians for modeling differences between distances. In our experiments (described in Section 5), we also find the Laplacian to outperform the Gaussian.

5 Experimental Results

We applied our algorithm to a test set of 15 images obtained using a calibrated digital camera in 8 different buildings (all of which had fairly different interior decoration themes from each other, and from the building from which the training set images were taken). Since the test set buildings contained a diverse range of orthogonal geometries (boxes, doors, hallways, cabinets, etc.), we believe that the results we present are indicative of the algorithm’s performance on images of new (Manhattan world) buildings and scenes.

Figures 9 shows the labeling obtained by the MRF on 6 images from the test set, as well as the resulting 3d reconstructions. Even in fairly complex environments or ones that do not perfectly respect the Manhattan world assumption, the algorithm is still able to label most of the planes correctly, and obtain reasonable 3d reconstructions.

We also evaluate the algorithm more formally. First, using a hand-labeling of the test set images, we measure the labeling error rate of the MRF. The overall accuracy of the MRF is 79.6%. Given that there are 6 possible labels for each pixel, random guessing would have obtained 16.7% accuracy. Table 1 shows a further breakdown of these results by planes and edges.¹⁶ Although our precision on edges was surprisingly low, this appears to be a consequence of only a very small fraction of the pixels being edge pixels, and did not seem to significantly affect the final reconstruction performance.

Table 1. MRF labeling errors on test set images

	planes	edges
Recall	80.6%	65.7%
Precision	89.1%	29.0%

Using a careful hand-labeling of the test set images (including both plane orientations and occluding edges), we also generated a full ground-truth 3d reconstruction of the test set scenes. We then measured the average errors in the reconstructed distances, for pixels at different ground-truth distances from the camera. These statistics do not include planes that were discarded during the

¹⁶ Recall is the fraction of plane/edge labels that we labeled correctly. Precision is, out of all the times we predicted a specific label, the fraction of times that the prediction was correct.

reconstruction, and thus might reflect a slightly overoptimistic performance metric, but nonetheless represents a fair comparison between the different algorithms.¹⁷ The results are shown in Figure 8. We also compare the Laplacian model with a Gaussian one (that, similar to our procedure for learning $\alpha_{p,p'}$, tries to adapt its variance parameters), and with an implementation of Sturm and Maybank's algorithm that assumes there are no occluding edges. When performing 3d reconstruction using our MRF's output labels, the Laplacian model appears to perform best.

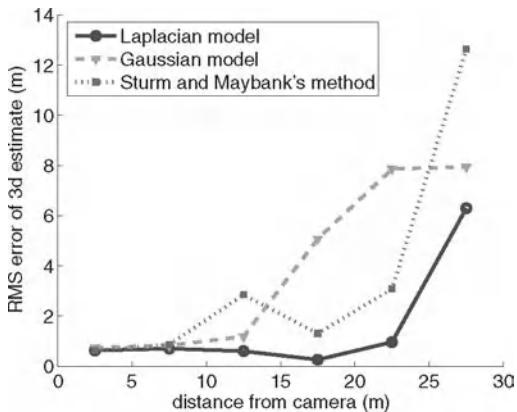


Fig. 8. Errors in 3d reconstructions, for pixels at different ground-truth distances from the camera.

6 Summary

We have presented an algorithm for fully automatic 3d reconstruction of indoor Manhattan world scenes from a single image. Our method uses an MRF to label each pixel as belonging to one of three plane orientations or one of three edge orientations. Given the MRF model's outputs, we use a Laplacian probabilistic model to infer a 3d reconstruction. Our experimental results show the algorithm performing well on a number of indoor scenes, even ones very different from the training set images. The work presented in this paper

¹⁷ See footnote 15 for details. Given the MRF output, all three algorithms discard (the same) 4% of pixels as belonging to the ceiling; 22% of pixels labeled as edges (whose distance is truly ambiguous, since they can be reconstructed as lying on either of two planes); and under 1% as outliers (reconstructed as lying outside the box described in footnote 15).

was restricted to Manhattan worlds, and it remains an important problem to generalize these ideas to other scenes. More generally, we believe that monocular depth estimation holds significant promise, and it remains an important problem to develop algorithms that exploit other single-image cues for depth estimation.

Acknowledgments

We give warm thanks to Sebastian Thrun, Ashutosh Saxena, Pieter Abbeel, and Morgan Quigley for helpful discussions. This work was supported by the DARPA LAGR program under contract number FA8650-04-C-7134.

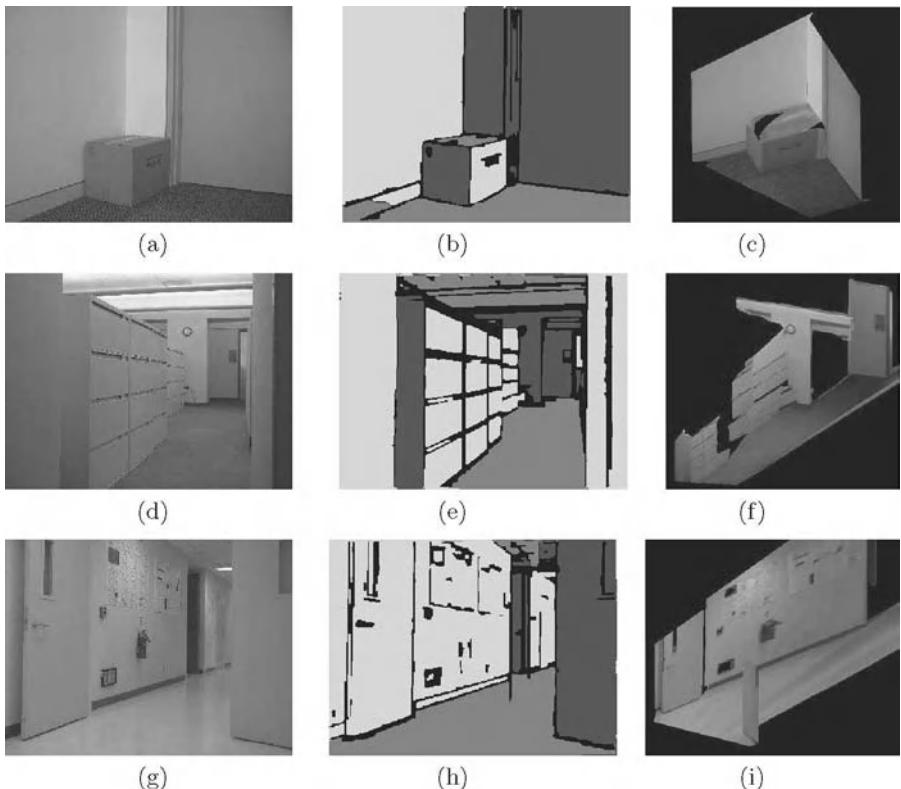
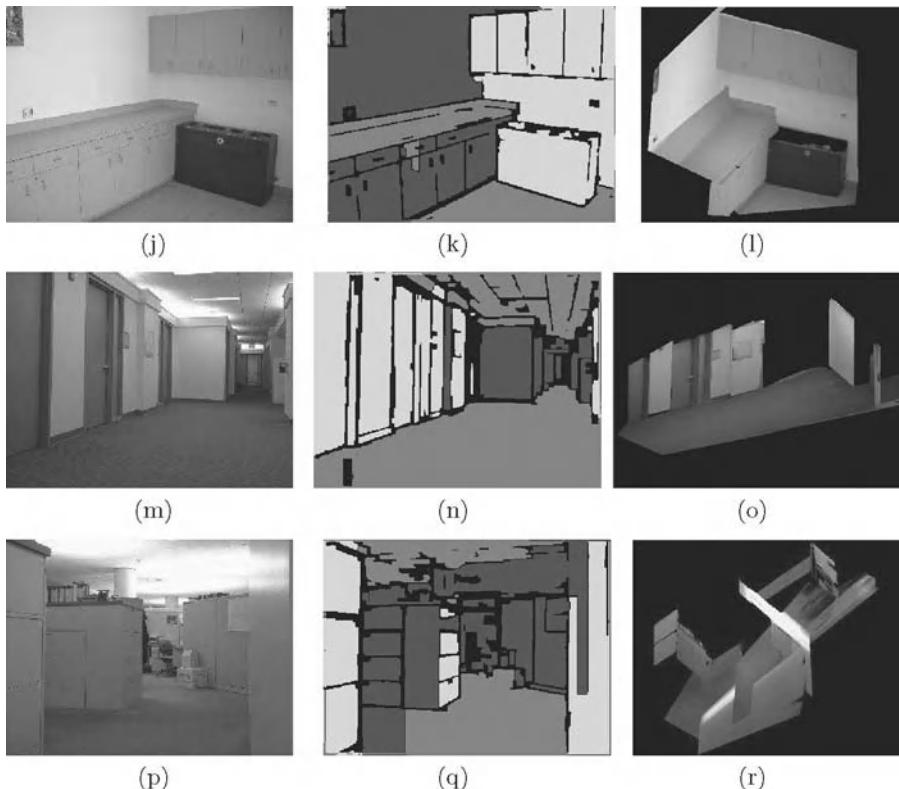


Fig. 9. Inferred 3d reconstructions of test set indoor scenes. Left column: Input image. Middle column: Labeling generated by MRF (red, green and blue correspond to the three plane orientations; black corresponds to all three edge orientations). Right column: Resulting 3d reconstructions.

**Fig. 9.** (continued)

References

1. J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, 1974.
2. J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
3. J. Coughlan and A.L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *IEEE International Conference on Computer Vision*, 1999.
4. J. Coughlan and A.L. Yuille. Manhattan world. *Neural Computation*, 15:1063–1088, 2003.
5. A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40:123–148, 2000.
6. P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. In *SIGGRAPH*, 1996.
7. E. Delage, H. Lee, and A. Y. Ng. A dynamic Bayesian network model for autonomous 3d reconstruction from a single indoor image. Unpublished manuscript, 2005.

8. P. Favaro and S. Soatto. Shape and radiance estimation from the information divergence of blurred images. In *European Conference on Computer Vision*, 2000.
9. Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59, 2004.
10. R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.
11. F. Han and S. C. Zhu. Bayesian reconstruction of 3d shapes and scenes from a single image. In *IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis*, pages 12–20, 2003.
12. F. Huang and Y. Ogata. Generalized pseudo-likelihood estimates for Markov random fields on lattice. *Annals of the Institute of Statistical Mathematics*, 2002.
13. A. Kosaka and A. C. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *CVGIP: Image Understanding*, 56:271–329, 1992.
14. J. Kosecka and W. Zhang. Video compass. In *European Conference on Computer Vision*, 2002.
15. P. Kovesi. Image features from phase congruency. *Videre: A Journal of Computer Vision Research*, 1, 1999.
16. P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. School of Computer Science & Software Engineering, The University of Western Australia.
Available from: <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>.
17. E. Lutton, H. Maitre, and J. Lopez-Krahe. Contribution to the determination of vanishing points using hough transform. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16:430–438, 1994.
18. J. Michels, A. Saxena, and A. Y. Ng. High-speed obstacle avoidance using monocular vision and reinforcement learning. In *International Conference on Machine Learning*, 2005.
19. A. Saxena, S. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Neural Information Processing Systems*, 2005.
20. G. Schindler and F. Dellaert. Atlanta World: An expectation-maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.
21. H.-Y. Shum, M. Han, and R. Szeliski. Interactive construction of 3d models from panoramic mosaics. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 1998.
22. P. F. Sturm and S. J. Maybank. A method for interactive 3d reconstruction of piecewise planar objects from single images. In *British Machine Vision Conference*, 1999.
23. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.
24. M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Trans. Information Theory*, 49(5):1120–1146, 2003.
25. R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21:690–706, 1999.

Robot Design and Control

Session Overview

Robot Design and Control

Claire J. Tomlin

University of California at Berkeley and Stanford University
tomlin@stanford.edu

Control theory to date has achieved tremendous success in the analysis and synthesis of single control systems, as well as in the development of control laws for simple groups of systems which are connected together by point-to-point wires (assumed reliable) so that information is received and processed synchronously at each subsystem. Many of these advances have been fueled by challenges in robotics: force feedback in haptic devices led to new formulations of stable control laws, coordination algorithms for robot swarms has likewise led to a theory of control for networked systems.

New vistas and challenges in robotics are continuing to push the envelope in control design. In an earlier session, new advances in the area of human-robot interaction were described. Here, the implications of this for control are discussed: the requirement for robots to interact in safe and meaningful ways with the handicapped, with soldiers, and in elder care, require new ways of thinking about a control theory that interacts seamlessly with humans, that is “self-aware”, that can provide guarantees about the limits of its operation for safety. This also leads us to think about how to endow robots with “human” qualities. And, more and more, robotics are being asked to function in extreme environments: at the micro-scale level where the sheer size demands engineering innovations, to the battlefield where robustness, ease of transport, and ease of use, are paramount. This session features four exciting new research directions spanning the aspects discussed above.

One is Enough! by Lauwers, Kantor and Hollis describes *Ballbot*, a cylindrical robot the height of an adult human, designed to be agile and responsive in its interaction with people in their environments. Multi-wheel, statically stable robots of this height would be clumsy and slow, with low centers of gravity, wide bases of support, and low accelerations. Maneuvering in tight spaces, even planning a trajectory through a partially closed door, is challenging for such robots. Ballbot is a statically unstable, but dynamically stable, agile cylindrical robot whose body is supported by a single, omni-directional spherical wheel. In this paper, the challenge of designing the control law which dynamically stabilizes this system is presented, and results shown.

A Steerable, Untethered, $250 \times 60 \mu\text{m}$ MEMS Mobile Micro-Robot by Donald, Levey, McGray, Paprotny, and Rus describes the smallest micro-robot ever – 1-2 orders of magnitude smaller in size than previous designs. The robot system is a cantilevered arm mounted on untethered component actuators moving on a uniform electrical grid. Here, the challenge is to design a control scheme which is simple enough to be robust at this scale. The paper discusses how the actuators are used to define two control modes, forward motion and turning, which may be coupled together to produce a control logic capable of controlling the device globally with impressively small error.

Some Issues in Humanoid Robot Design by Takanishi, Ogura, and Itoh addresses a range of new ideas and the resulting design issues for robots that interact with humans. These include the study of human motion and the human muscular-skeletal system to inspire new design structures and control laws for robot walking; new thoughts about controlling very high DOF systems; the mechanisms for realistic joint motion; and finally, the expression of emotion through arms, hands, and face (eyes, eyelids, eyebrows, lips and jaws). Exciting results on their two humanoid robots, WABIAN-2 and WE-4RII, are presented.

Finally, **That which does not stabilize, will only make us stronger**, by Kazerooni and Steger describes the design and control of BLEEX, the Berkeley Lower Extremity Exoskeleton. BLEEX is a human exoskeleton which fits around the torso, legs, and shoes of a human body (the pilot) to effectively carry 75 lb of payload while allowing the pilot to walk up to 2.9 mph – such a system has huge implications for soldiers who are continually asked to carry such heavy loads over long distances. The paper focuses on the novel control design: how does one design a control scheme which does not require direct measurements from the pilot, yet moves in such a way to shadow, with very little delay, the pilots movements so that the pilot feels very little force from the payload. This motivates a design which is counter to today's control design techniques of minimizing system sensitivity to errors or disturbances: Kazerooni has designed a control scheme which does just the opposite, and he demonstrates its success in this paper.

One Is Enough!

Tom Lauwers, George Kantor, and Ralph Hollis

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
`{tlauwers,kantor,rhollis}@ri.cmu.edu`

Summary. We postulate that multi-wheel statically-stable mobile robots for operation in human environments are an evolutionary dead end. Robots of this class tall enough to interact meaningfully with people must have low centers of gravity, overly wide bases of support, and very low accelerations to avoid tipping over. Accordingly, we are developing an inverse of this type of mobile robot that is the height, width, and weight of a person, having a high center of gravity, that balances dynamically on a *single spherical wheel*. Unlike balancing 2-wheel platforms which must turn before driving in some direction, the single-wheel robot can move directly in any direction. We present the overall design, actuator mechanism based on an inverse mouse-ball drive, control system, and initial results including dynamic balancing, station keeping, and point-to-point motion.

1 Motivation

A significant, but frequently overlooked problem is that statically-stable wheeled mobile robots can easily become dynamically *unstable*. If the center of gravity is too high, or the robot accelerates/decelerates too rapidly, or is on a sloping surface, the machine can tip over. A robot must be tall enough to be able to interact with people and the human environment at a reasonable height. On the other hand, it must be skinny enough to easily make its way around without bumping into things or getting into peoples' way.

What is needed are robots that are safe; agile and capable of graceful motion; slender enough to easily maneuver in cluttered, peopled environments; and which readily yield when pushed around. It is surmised that intelligent machines of this sort can only be achieved with *dynamic stability*. This idea follows the model of humans and other animals which are intrinsically dynamically stable.

2 Background

A two-wheeled robot with inverse pendulum control developed in Japan was demonstrated in 1994 [2]. The two-wheeled design eliminated the need for a third castor-

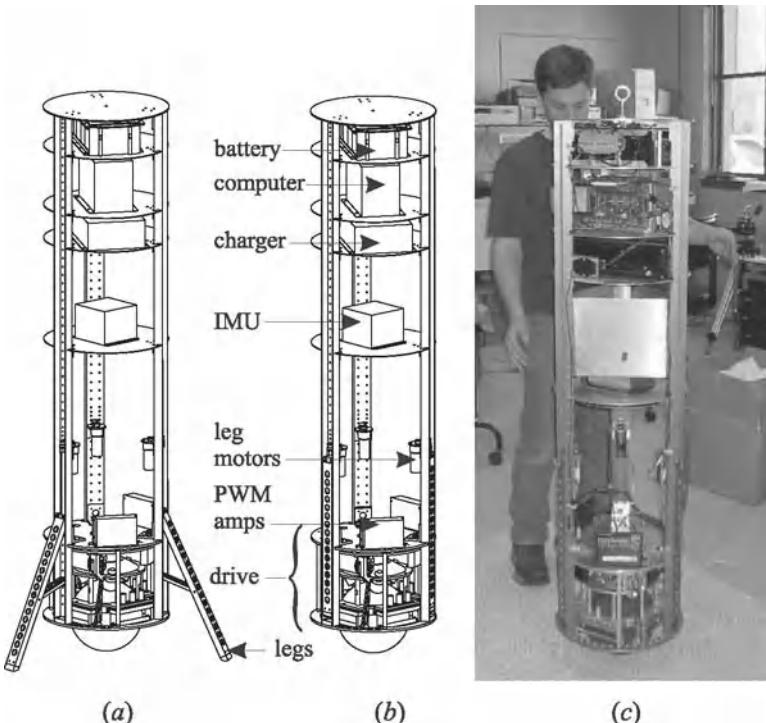


Fig. 1. Ballbot design and realization: (a) with three legs deployed, (b) with legs retracted into body, (c) balancing and station keeping.

ing wheel. The same group introduced a one-wheel balancing robot [3]. The wheel is a prolate ellipsoid like a rugby ball and is driven with an axle along the major axis. The body of the robot has a hinge above and perpendicular to this axis. The robot balances in the forward/backward directions by application of wheel torque in the manner of the two-wheeled design, and balances from side to side by leaning the body left or right at the actuated hinge. Recently, balancing wheel chairs¹ and balancing 2-wheel “Segway personal mobility devices”² have been introduced. The 2-wheel RMP robotic platforms [4] based on the Segway are the subject of much recent development in robotic locomotion.

The previous work on dynamically-stable rolling machines provides inspiration for our current research, yet is distinctly different. For example, there is no previous work proposing a balancing rolling machine whose body is supported by a single omni-directional spherical wheel. The previous rolling/balancing machines cannot immediately drive in a given direction without first re-orienting the drive mechanism. For example, a two-wheel balancing machine such as the Segway RMP cannot maneuver in tight spaces by moving sideways; a robot based on such a machine could

¹ Independence Technology, <http://www.indetech.com>.

² Segway human transporter, <http://www.segway.com>.

not open and close a door without knowing the precise location of the hinges in order to establish the correct turning radius. The rugby-ball robot cannot turn in place, but can only turn in a wide arc.

3 System Description

Ballbot, shown in Fig. 1, is a reconfigurable research platform developed and constructed to validate the notion of a dynamically stable robot resting atop a single, spherical drive wheel. It was designed to meet two goals: approximate the dimensions of a human being, and create a platform that is easily reconfigured for various present and future research efforts. The body is a cylinder 1.5 m tall, with a diameter of 400 mm and a weight of 45 kg. Three aluminum channels, held together by circular decks, define the structure of Ballbot's body. Three retractable landing legs are attached to the lower third of the channels, which when deployed allow Ballbot to remain standing after being powered down. Active components, such as computing, power, and sensing, are mounted on the decks, allowing these elements to be placed at varying positions along Ballbot's axis. Figures 1(a) and (b) show the design and Fig. 1(c) shows its present configuration successfully balancing and station keeping.

Ballbot is designed to be entirely self-contained; power is supplied by a 48V lead acid battery with operating time of several hours, and computing is performed onboard by a 200 MHz Pentium processor. Communication with Ballbot is through an 802.11b wireless link. A Crossbow Technology VG700CA-200 Inertial Measuring Unit (IMU) emulating a vertical gyro provides Kalman-filtered pitch and roll angles and rates with respect to gravity. The drive motors are connected to Copley Model 412 PWM amplifiers, with 1024 cpr encoders feeding motor shaft position back to the computer. Additionally, 1024 cpr encoders are placed on the passive rollers to measure ball rotation. The IMU and encoders provide all data required for full-state feedback control.

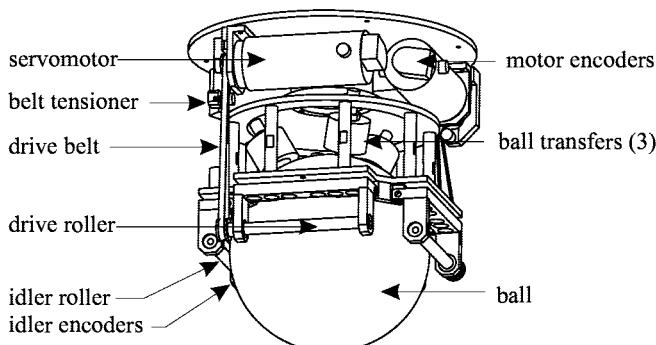


Fig. 2. Ballbot inverse mouseball drive mechanism

The drive mechanism, shown in Fig. 2, is essentially the inverse of a mouse ball drive: instead of the mouse ball driving the mouse rollers to provide computer input, rollers drive the ball to produce motion. The ball is a 200 mm diameter hydroformed steel shell covered with a 3.2 mm thick urethane outer layer. We have fabricated balls with urethane formulations of several different durometers. The ball is actuated by a pair of 12.7 mm diameter smooth stainless steel rollers placed orthogonally at the sphere's equator. These rollers are linked through timing belts to two high torque DC servomotors. Opposite the drive rollers are two spring-loaded passive idler rollers that apply force at the ball's equator to maintain contact between the drive rollers and the ball. This arrangement represents a compromise since some slippage is always present. For example, if one roller is being driven, the orthogonal roller must be slipping. This simultaneously demands both a high-friction and low-friction material for the ball. On the other hand, it is always desirable to have high friction between the ball and the floor. The drive works well but a fair amount of ball wear has been experienced, and we are at present still seeking a satisfactory compromise solution. The entire drive mechanism is attached to the body with a large diameter thrust bearing, allowing a third actuator (currently not installed) to re-orient the body in yaw. Finally, the entire Ballbot body rests on top of the ball on three commercial low friction, omni-directional ball transfer devices.

4 Simplified Ballbot Model

For the purposes of developing a stabilizing controller, we introduce and derive equations of motion for a simplified model of Ballbot. In this model, the Ballbot ball wheel is a rigid sphere, the body is rigid, and the control inputs are torques applied between the ball and the body. There is no slip between the wheel and the floor. Friction between the wheel and the floor and between the wheel and the body is modeled as viscous damping. Further, we assume that the motion in the median sagittal plane and median coronal plane is decoupled and that the equations of motion in these two planes are identical. As a result, we can design a controller for the full 3D system by designing independent controllers for the two separate and identical planar systems.

Figure 3 is a diagram depicting the planar model. The Lagrangian formulation is used to derive the nonlinear equations of motion for the simplified model (see, e.g., [1]). The first step is to compute the kinetic energy K_b of the ball:

$$K_b = \frac{I_b \dot{\theta}^2}{2} + \frac{m_b(r_b \dot{\theta})^2}{2},$$

where I_b , m_b , and r_b are, respectively, the moment of inertia, mass, and radius of the ball. The potential energy of the ball is $V_b = 0$. The kinetic energy K_B and potential energy V_B of the body are

$$K_B = \frac{m_B}{2} \left(r_b^2 \dot{\theta}^2 + 2r_b \ell (\dot{\theta}^2 + \dot{\phi}^2) \cos(\theta + \phi) + \ell^2 (\dot{\theta} + \dot{\phi})^2 \right) + \frac{I_B}{2} (\dot{\theta} + \dot{\phi})^2,$$

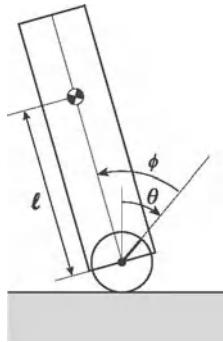


Fig. 3. Planar simplified Ballbot model used for controller design.

$$V_B = m_B g \ell \cos(\phi + \theta),$$

where I_B is the moment of inertia of the body about the center of the ball, ℓ is the distance between the center of the ball and the center of mass of the body, m_B is the mass of the body, and g is the acceleration due to gravity. The total kinetic energy is $K = K_b + K_B$ and the total potential energy is $V = V_b + V_B$.

Define the system configuration vector $q = [\theta \ \phi]^T$. The Lagrangian \mathcal{L} is a function of q and \dot{q} and is defined to be $\mathcal{L}(q, \dot{q}) = K - V$.

Let τ be the component of the torque applied between the ball and the body in the direction normal to the plane. To model the viscous friction terms, define the vector

$$D(\dot{q}) = \begin{bmatrix} \mu_\theta \dot{\theta} \\ \mu_\phi \dot{\phi} \end{bmatrix},$$

where μ_θ and μ_ϕ are the viscous damping coefficients that model ball-ground and ball-body friction, respectively. Using this notation, the Euler-Lagrange equations of motion for the simplified Ballbot model are

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \begin{bmatrix} 0 \\ \tau \end{bmatrix} - D(\dot{q}).$$

After computing the derivatives in the Euler-Lagrange equations and rearranging terms, the equations of motion can be expressed as

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) + D(\dot{q}) = \begin{bmatrix} 0 \\ \tau \end{bmatrix}. \quad (1)$$

The mass matrix $M(q)$ is

$$M(q) = \begin{bmatrix} \Gamma_1 + 2m_B r_b \ell \cos(\theta + \phi) & \Gamma_2 + m_B r_b \ell \cos(\theta + \phi) \\ \Gamma_2 + m_B r_b \ell \cos(\theta + \phi) & \Gamma_2 \end{bmatrix},$$

where

$$\begin{aligned}\Gamma_1 &= I_b + I_B + m_b r_b^2 + m_B r_b^2 + m_B \ell^2, \\ \Gamma_2 &= m_B \ell^2 + I_B.\end{aligned}$$

The vector of Coriolis and centrifugal forces is

$$C(q, \dot{q}) = \begin{bmatrix} -m_B r_b \ell \sin(\theta + \phi) (\dot{\theta} + \dot{\phi})^2 \\ 0 \end{bmatrix}$$

and the vector of gravitational forces is

$$G(q) = \begin{bmatrix} -m_B g \ell \sin(\theta + \phi) \\ -m_B g \ell \sin(\theta + \phi) \end{bmatrix}.$$

To put these equations into standard nonlinear state space form, define the state vector to be $x = [q^T \ \dot{q}^T]^T$ and define the input $u = \tau$. This together with Eq. 1 yields

$$\dot{x} = \left[M(q)^{-1} \left(\begin{bmatrix} 0 \\ u \end{bmatrix} - C(q, \dot{q}) - G(q) - D(\dot{q}) \right) \right] \triangleq f(x, u).$$

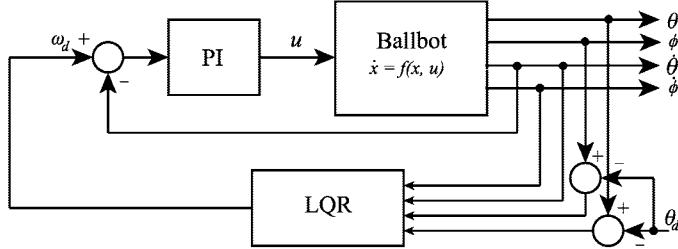


Fig. 4. Structure of stabilizing linear feedback controller.

5 Stabilizing Feedback Controller

The linear controller used to stabilize Ballbot has two loops: an inner loop that feeds ball velocity $\dot{\theta}$ back into a PI controller, and an outer loop linear quadratic regulator (LQR) that uses full state feedback. This architecture is shown in Fig. 4. The proportional gain k_p and integral gain k_i in the PI controller are chosen and experimentally tuned so that the actual ball velocity $\dot{\theta}$ tracks the desired ball velocity ω_d . The integral term adds an extra state to the system. Define the augmented state vector $x_a = [x^T \ x_5]^T$. The closed loop equations of motion of the inner loop can then be written as

$$\dot{x}_a = \begin{bmatrix} f(x, k_p(\omega_d - \dot{\theta}) + k_i(x_5 - \theta)) \\ \omega_d \end{bmatrix} \triangleq f_a(x_a, \omega_d).$$

The outer loop is designed by linearizing the inner loop equations of motion and applying LQR. Note that the simplified Ballbot system is at equilibrium whenever $\sin(\theta + \phi) = 0$ and $\dot{\phi} = \dot{\theta} = 0$. The objective is to design a controller that will balance Ballbot with the body straight up and hold it in a fixed position $\theta = 0$, which is equivalent to stabilizing the equilibrium point at $x_a = 0$. We begin by linearizing the equations of motion about this point:

$$\dot{x}_a = \underbrace{\frac{\partial f_a}{\partial x_a} \Big|_{x_a=0, \omega_d=0}}_A x_a + \underbrace{\frac{\partial f_a}{\partial \omega_d} \Big|_{x_a=0, \omega_d=0}}_B \omega_d.$$

Working out the partial derivatives yields

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ M_*^{-1} \begin{bmatrix} -m_B g \ell & -m_B g \ell & \mu_\theta & 0 & 0 \\ -m_B g \ell - k_i & -m_B g \ell - k_p & \mu_\phi & k_i \end{bmatrix}, \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ 0 \\ M_*^{-1} \begin{bmatrix} 0 \\ k_p \end{bmatrix} \\ 1 \end{bmatrix},$$

where M_* is simply the mass matrix $M(q)$ evaluated at $\theta = \phi = 0$.

Now LQR can be used to find a linear state feedback controller that stabilizes the system about $x_a = 0$ and minimizes the cost function

$$J = \int (x_a(t)^T Q x_a(t) + R \omega_d(t)^2) dt.$$

We choose the structure of Q to be

$$Q = \begin{bmatrix} \gamma_b + \gamma_B & \gamma_B & 0 & 0 & 0 \\ \gamma_b & \gamma_B & 0 & 0 & 0 \\ 0 & 0 & \gamma_b + \gamma_{\dot{B}} & \gamma_{\dot{B}} & 0 \\ 0 & 0 & \gamma_{\dot{B}} & \gamma_{\dot{B}} & 0 \\ 0 & 0 & 0 & 0 & \gamma_5 \end{bmatrix},$$

where γ_b , γ_B , $\gamma_{\dot{B}}$, and γ_5 can be loosely thought of as controlling the relative convergence rates of the ball angle, body angle, ball angular velocity, body angular velocity, and x_5 , respectively. In practice, these parameters were hand tuned based on simulation results. For a given choice of Q and R , Matlab's LQR command can be

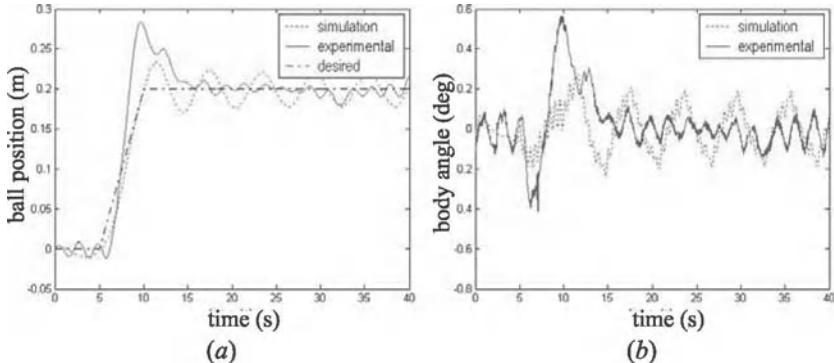


Fig. 5. Moving between two locations in a straight line: (a) ball position in meters, (b) body angle in degrees.

used to compute the associated gain matrix K , which defines the stabilizing feedback control law $\omega_d = -Kx_a$.

When implementing the controller on the actual robot, we were forced to deviate slightly from the controller presented above. We found that there is a practical limit on the magnitude of the gain k_4 that multiplies $\dot{\phi}$. Exceeding this limit induces an oscillation not present in the simplified Ballbot model. We hypothesize that this oscillation is due to flexibility in the body frame and the mechanics of the soft urethane layer that couples the drive roller to the ball. The K matrix generated by the LQR algorithm gives a k_4 that exceeds the practical limit, so we manually adjusted k_4 to an allowable level. Unfortunately, with this limit on k_4 , it is not possible to directly stabilize Ballbot, which explains the need for the inner PI loop. Also, the gain k_5 turns out to be negligible, so it is set to zero in the experiments.

6 Initial Results

A number of tests were conducted to characterize physical system performance, and to make comparisons with simulation. During operation on a hard tiled floor, it was found that the machine was able to balance robustly, strongly resisting attempts to tip it over when a person applied torques to the body. However, it was not able to simultaneously balance and station keep. When operated on a carpeted surface, Ballbot was able to do both, presumably due to the extra damping afforded by the carpet material.

In the test run shown in Fig. 5, Ballbot was commanded to move from a starting position in a straight line to a goal position. There is an initial retrograde ball motion causing the body to lean toward the goal position, followed by a reverse motion to stop at the goal. As mentioned in the previous section, differences between simulation and experiment might derive from unknown frictional and spring forces. The divergence when station keeping is at most about 40 mm in position, and 0.5° in tilt.

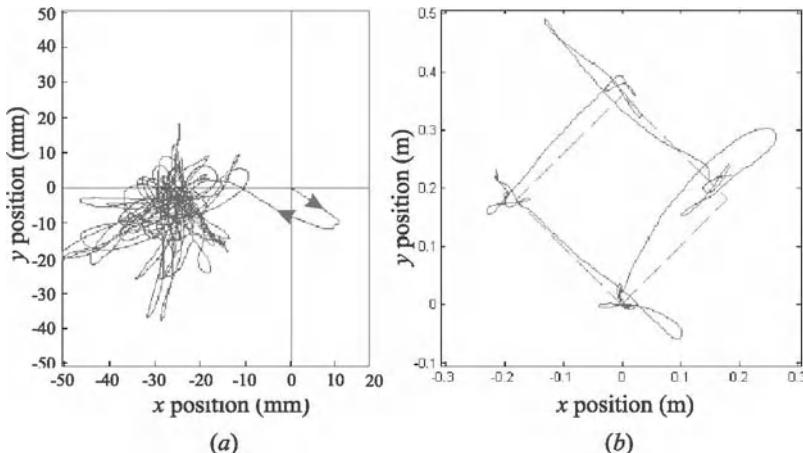


Fig. 6. Plots of the ball path during (a) balancing and station keeping, and (b) attempting to move in a square.

To see the typical motion jitter experienced during operation, one may plot the paths taken as the ball moves around on the carpeted floor. Figure 6(a) shows data taken from a 99 s run where Ballbot was released slightly out of balance, which was rapidly corrected by ball motion, followed by station keeping within a roughly circular region of about 40 mm diameter. Figure 6(b) shows Ballbot's attempt to track a square trajectory.

7 Discussion

Our results are preliminary and there is much that remains to refine Ballbot's model and control. Nevertheless, it would appear that Ballbot and its progeny might well represent the vanguard of a new type of wheeled mobile robot capable of agile, omnidirectional motion. Such robots, combined with the research community's ongoing work in perception, navigation, and cognition, could yield truly capable intelligent mobile robots for use in physical contact with people. If realizable and economically viable, they might well function as aids to elderly or disabled persons; provide guidance and assistance in public spaces; help with education and entertainment; perform domestic cleaning and housekeeping; or fetch and carry everyday objects. The more immediate goal of our research is simply to gain a deeper understanding of how such dynamic agility can be achieved in mobile machines interacting with people and operating in normal home and workplace environments.

Acknowledgment. This work was supported in part by NSF grant IIS-0308067.

References

1. A.M. Bloch. *Nonholonomic Mechanics and Control*. Springer, 2003.
2. Y.-S. Ha and S. Yuta. Trajectory tracking control for navigation of self-contained mobile inverse pendulum. In *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pages 1875–1882, 1994.
3. R. Nakajima, T. Tsubouchi, S. Yuta, and E. Koyanagi. A development of a new mechanism of an autonomous unicycle. In *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pages 906–12, Grenoble, France, September 7-11 1997.
4. H. G. Nguyen, J. Morrell, K. Mullens, A. Burmeister, S. Miles, N. Farrington, K. Thomas, and D. Gage. Segway robotic mobility platform. In *SPIE Proc. 5609: Mobile Robots XVII*, Philadelphia, PA, October 2004.

A Steerable, Untethered, $250 \times 60 \mu\text{m}$ MEMS Mobile Micro-Robot

Bruce R. Donald^{1,2,*}, Christopher G. Levey³, Craig D. McGray¹,
Igor Paprotny¹, and Daniela Rus⁴

¹ Dartmouth Computer Science Department, Hanover, NH 03755

² Dartmouth Chemistry Department, Hanover, NH 03755

³ Dartmouth Thayer School of Engineering, Hanover, NH 03755

⁴ Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Boston, MA

Summary. We present a steerable, electrostatic, untethered, MEMS micro-robot, with dimensions of $60 \mu\text{m}$ by $250 \mu\text{m}$ by $10 \mu\text{m}$. This micro-robot is **1 to 2 orders of magnitude** smaller in size than previous micro-robotic systems. The device consists of a curved, cantilevered steering arm, mounted on an untethered scratch drive actuator. These two components are fabricated monolithically from the same sheet of conductive polysilicon, and receive a common power and control signal through a capacitive coupling with an underlying electrical grid. All locations on the grid receive the same power and control signal, so that the devices can be operated without knowledge of their position on the substrate and without constraining rails or tethers. Control and power delivery waveforms are broadcast to the device through the capacitive power coupling, and are decoded by the electromechanical response of the device body. Individual control of the component actuators provides two distinct motion gaits (forward motion and turning), which together allow full coverage of a planar workspace (the robot is globally controllable). These MEMS micro-robots demonstrate turning error of less than $3.7^\circ/\text{mm}$ during forward motion, turn with radii as small as $176 \mu\text{m}$, and achieve speeds of over $200 \mu\text{m/sec}$, with an average step size of 12 nm . They have been shown to operate open-loop for distances exceeding 35 cm without failure, and can be controlled through teleoperation to navigate complex paths.

1 Introduction

This paper addresses the design, fabrication, and control of micro-robots that are small enough to locomote on the surface of an integrated circuit, and to interact with parts as small as individual MEMS components. While there

* Corresponding author (e-mail: brd@cs.dartmouth.edu)

are many MEMS devices with sizes measured in tens of microns, the smallest micro-robotic systems yet produced have dimensions on the order of millimeters or centimeters. A primary reason for this, is that existing micro-robot architectures employ a rigid chassis on which to mount power, locomotion, steering, communication, and control systems. While these active components often include thin-film MEMS actuators, the chassis is a macro-scale part such as, for example, a silicon die. For this reason, these micro-robots are often referred to as “walking chips” [1, 2, 3, 4, 5].

We build on our previous work [6, 7], which describes MEMS untethered scratch drive actuators that can move only along *straight-line* (linear) trajectories, and demonstrate a new generation of micro-robots that are *steerable along arbitrary trajectories* (i.e., globally controllable [8] in $\mathbb{R}^2 \times S^1$). Our new devices integrate, in a single thin film device body, not only power delivery, locomotion, and communications, but now also steering and control systems. This allows us to build a micro-robot that is one to two orders of magnitude smaller in length than previous systems, and many thousands of times smaller in overall mass. The device is capable of two distinct motions: it can either translate forwards, or turn through an arc with a fixed minimum radius of approximately 175 μm . Alternation of these two motion primitives allows for execution of turns with any arbitrary radius larger than the minimum. These two operations are sufficient to provide our device with global controllability.

Figure 1 shows the structure of this device. It consists of an untethered scratch drive actuator [6, 7] (**A**), with a cantilevered steering arm (**B**) that protrudes from one side.

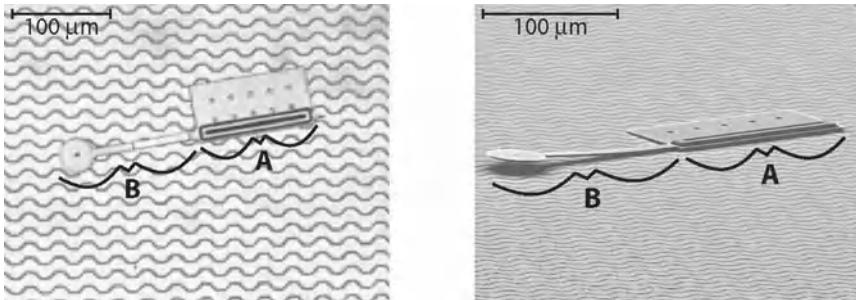


Fig. 1. Optical (left) and electron (right) micrographs of an electrostatic micro-robot. The device consists of an untethered scratch drive actuator (**A**) [6, 7], with a cantilevered steering arm (**B**) that protrudes from one side. The untethered scratch drive is used for propulsion, while the steering arm can be raised or lowered to turn. An array of insulated interdigitated electrodes (lighter-colored background) provides electrical power and control signals to the device.

The untethered scratch drive is used for locomotion, while the steering arm is used to raise or lower a stylus into contact with the substrate. When this

stylus is in contact with the substrate, it provides enough friction to cause the device as a whole to turn. The device receives its electrical power and control signals through a grid of insulated interdigitated electrodes that cover the device's operating environment. Since the control signal and electrical power are both available to a device anywhere within this environment, the device can move freely, unconstrained by wires or rails that power most electrostatic MEMS devices. The operating environment used for the devices presented in this paper extends across 6.25 square millimeters of surface, and could be made even larger if desired.

Previous approaches to micro-robot control rely on providing signals to all sub-systems continuously and simultaneously. In macro-scale robotic systems, instructions are generally only transmitted once, and are then stored on-board the device until they are replaced with a new instruction. While macro-scale devices typically implement this data storage with electronics, a thin-film MEMS device can utilize the simpler alternative approach of storing state information in the electromechanical flexure of the active components. The devices described in the present paper are controlled through *electromechanical state-based* component addressing. We exploit the hysteresis of the components by applying sequences of voltages in a control waveform. First, the desired behavior (forward motion or turning) is specified by an electrical pulse, and is stored in the elastic flexure of the device steering arm. Then, a continuous AC drive waveform is applied to actuate the scratch drive and produce motion. This is achieved by nesting the electromechanical hysteresis loops of the scratch drive within the hysteresis loop of the actuator steering arm.

The micro-robots are composed of polycrystalline silicon using a multi-user foundry MEMS process [9]. After receipt from the foundry, the die are coated with 830 nm of thermally evaporated chromium to create a well-controlled stress gradient in the cantilevered steering arms, determining the cantilevers' tip deflection. Full details of the fabrication process are presented in Appendix A.

The performance of the devices was tested under both open-loop and tele-operated control. Micro-probes connect the electrodes to a function generator and amplifier. During teleoperation, a human operator switches between two different waveforms produced by the function generator in order to control the motion of the untethered micro-robotic device (see Figure 1). A camera records the device's motion through an optical microscope, allowing the operator to make the necessary adjustment to guide the device along the desired path. Section 6 discusses the reliability of the basic motion primitives, and shows some examples of more complex paths produced through teleoperation. The device has the ability to push and manipulate other MEMS-scale components [6, 7].

The introduction of a micro-robotic device with size less than $250 \mu\text{m}$ could extend and enable the set of micro-robot applications that have been previously identified. These include security and surveillance [3]; exploration of

hazardous environments; and biomedical research [10]. Of particular interest are those applications which allow a controlled environment for micro-robot operation, in which cleanliness and surface smoothness can be carefully maintained, and in which an ambient power source can be conveniently applied. Such applications include the manipulation and assembly of hybrid microsystems [10, 3]; micro-scale self-reconfiguring robotics [11]; and MEMS infosecurity self-assembly [12], where autonomous locomotion of micro-devices is a primary requirement. We envision that the devices, designs, and control systems presented in this paper will enable these applications for micro-robots.

2 Related Work

Previous work has produced a number of micro-robots with dimensions on the order of millimeters or centimeters [1, 13, 3, 5]. Past systems have delivered power through vibration [14], photo-thermal transduction [1], inductive coupling [2], and electrically through gold bonding wire [3]. The capacitively-coupled electrostatic power delivery mechanism that we described in [6, 7] is well-suited to the untethered devices presented in the current paper. While the linear untethered scratch drive actuators presented in [6, 7] provide some building blocks for our current power delivery system, the devices in [6, 7] could only be driven in straight lines. The present paper describes the design, fabrication, and control challenges in making *untethered steerable micro-robots* that can execute complex paths and are globally controllable [8]. These capabilities are essential for micro-robotic applications.

In previous micro-robotic devices, steering systems have been implemented primarily through differential operation of matched pairs or arrays of actuators [1, 14]. In these devices, each actuator contributes a small propulsive force to the device as a whole, which then moves as the vector sum of the forces provided by the component actuators. The device described in the present paper uses only two actuators: one for propulsion, and a second one to raise and lower a stylus into frictional contact with the substrate. This simplifies the overall device, reduces its size, and allows for precisely-controlled turning motions, even in the presence of small surface abnormalities.

3 Power Delivery

In [6, 7], we presented a mechanism for delivering power to linear untethered MEMS actuators, via a capacitive coupling across a thin film of thermal silica. In this mechanism, a silicon substrate is covered with rows of insulated interdigitated electrodes. When a conductive actuator, such as a scratch drive, rests on top of these electrodes (as shown on Figure 1), it forms the capacitive circuit with the underlying electrodes. In this way, a voltage is applied to the actuator, regardless of its position and orientation relative to the underlying

electrodes (i.e. no position-restricting wires or tethers are required.) Each time the voltage is cycled, the scratch drive moves forward by a small increment, known as the *step size*. The frequency at which this cycle occurs is known as the *stepping frequency*, and the speed of the actuator is the product of its stepping frequency and its average step size. To deliver power to our steerable MEMS micro-robots, we have used substrates covered with zirconia-insulated gold electrodes. The edges of the electrodes are jagged (as shown in Figure 1) to help maintain a consistent voltage on the actuator, regardless of the actuator's pose. Further details regarding the power delivery mechanism are provided in Appendix B.

4 Steering

The micro-robot controls its direction by raising and lowering its steering arm. Figure 2 shows a close-up view of this actuator. It consists of a $133\text{-}\mu\text{m}$ -long curved cantilever beam, with a disc at its tip. At the center of the disc, a $0.75\text{-}\mu\text{m}$ -high dimple serves as the stylus for frictional contact. The dimple has a radius of $1.5 \mu\text{m}$, and the surrounding disc has a radius of $18 \mu\text{m}$.

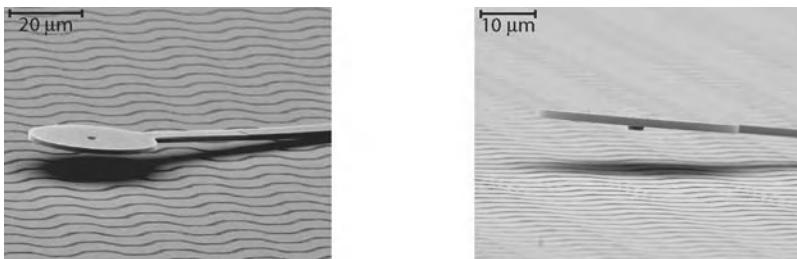


Fig. 2. Electron micrographs of the steering arm sub-system. The stylus used for frictional contact consists of a $0.75 \mu\text{m}$ dimple, visible beneath the end of the arm. An $18 \mu\text{m}$ radius disc increases the electrostatic force on the arm, which is curled upwards to increase the gap between the stylus and the substrate.

When the steering arm is in the raised position, the device as a whole behaves like a linear untethered scratch drive actuator [6, 7]. In this case, when an oscillating voltage is applied, the device will move forward in a straight line. To actuate the device forward, we chose a pulsed waveform with peak and minimum electrode-to-electrode voltages of 112 V and 39 V, respectively.

Note that the voltage applied to the electrode array differs from the potential between the scratch-drive actuator and the substrate, due to the nature of the power delivery mechanism. The power delivery mechanism forms a capacitive circuit between the electrodes and the scratch drive actuator (see Figure 8, p. 15, Appendix B). This circuit sets the potential of the scratch-drive actuator plate to approximately the midpoint between the potential of

the inter-digitated electrodes. Since the "even" electrodes (see Figure 8) are held at 0 V while the pulsed drive waveform is only applied to the "odd" electrodes (Figure 8), the resulting voltage affecting the scratch-drive actuator is approximately half of the voltage applied to the electrode field. Thus, the potential affecting the scratch-drive actuator during the application of the power delivery waveform has an effective peak and minimum voltages of approximately 56 V and 19 V, respectively.

The drive waveform is adequate to actuate the scratch drive actuator, but does not disturb the steering arm, regardless of whether the arm is in its raised or lowered position. For this reason, the same drive waveform can be applied either when the device is going straight or when it is turning. The behavior of the device is changed only by the position of its stylus. Further details regarding the design of the drive waveform are provided in Section 5, and also in [15, p. 17, Sec. C].

Before the micro-robot can turn, the stylus at the tip of its steering arm must be lowered into contact with the substrate. When the stylus is lowered, it creates friction at the contact point, causing the device to turn. Then, the drive waveform is applied. The frictional force acting on the stylus as the scratch drive actuates causes the device to turn. If the maximum available force of friction on the stylus exceeds the force applied on it by the scratch drive, then the stylus will not move, and the device will pivot around it.

There are two considerations that must be taken into account in the design of the steering arm. First, the arm must be stiff enough that the peak voltage of the drive waveform (112 V) does not inadvertently pull it into contact with the substrate. Second, it must be flexible enough that, once in contact, the minimum voltage of the drive waveform (39 V) does not allow it to release from the substrate. The voltage at which the beam will snap down into contact with the electrodes is called the *snap-down voltage*. When the voltage is subsequently decreased, the tip of the cantilever will remain in contact with the substrate until another instability is reached, and it snaps upward. This latter instability is known as the *release voltage*.

The steering arms on the micro-robots presented in this paper have snap-down and release voltages of approximately 60 V, and 15 V, respectively.⁵ In this way, the stylus can be raised and lowered at will, independent of forward motion. This allows the power delivery waveform to be used both to control the state of the arm as well as to provide energy to propel the device forward. Further details of the design of the steering arm are provided in Appendix C.

5 Control

This section describes the instruction set of the MEMS micro-robots, and shows how to encode it in a control waveform to specify device behavior. The

⁵ These values correspond to electrode-to-electrode voltages of 120 V and 30 V, respectively.

devices presented in this paper can, at any given time, be in one of the four distinct states shown in Figure 3. The stylus can be either up or down, and the scratch drive can be either flexed or relaxed.

We can now define four voltages that comprise the instruction set of the micro-robot. We have:

$\mathbf{V_1 = 0 \text{ V}}$	$\mathbf{V_2 = 39 \text{ V}}$	$\mathbf{V_3 = 112 \text{ V}}$	$\mathbf{V_4 = 140 \text{ V}}$
Raises Steering Arm			Lowers Steering Arm
Relaxes Scratch Drive	Relaxes Scratch Drive	Flexes Scratch Drive	Flexes Scratch Drive

To be able to operate the scratch drive independently of the position of the stylus, we need the drive waveform to fit within the voltage range defined by the steering arm's snap-down and release voltages. Since V_2 and V_3 fall between the snap-down and release voltages of the steering arm, application of these two voltages will not change the state of the steering arm actuator.

With these four instructions, we can model the system as the finite state machine shown in Figure 3. Here, the set of discrete DC voltages, $\{V_1, V_2, V_3, V_4\}$ comprises the transitions, and the zero-voltage state, S_0 , is the start state. The pair of voltages, (V_2, V_3) comprises the drive waveform discussed in Section 4. The blue and red transitions in Figure 3 correspond to the two motion operations of the device that can occur when the drive waveform is applied.

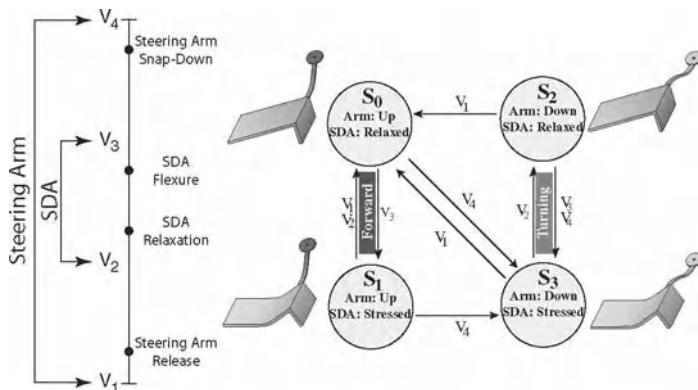


Fig. 3. The state transition diagram of the micro-robot. Four voltages, $V_1 < V_2 < V_3 < V_4$, are used in constructing control waveforms. SDA = untethered scratch drive actuator.

It is easy to see from this state transition diagram that all four system states can be reached, and to compute the voltage sequence required to achieve each one. This leads directly to the control waveforms shown in Figure 4. Both waveforms begin by selecting the system state associated with the desired motion, and then applying the drive waveform. After 250 steps, the waveform

polarity is reversed to minimize charge-trapping in the dielectric. The desired state is refreshed, and then the drive waveform is continued.

The control system can be extended to include an arbitrary number of states. Further details regarding the extensibility of the control system can be found in [15, p. 19, Sec. E].

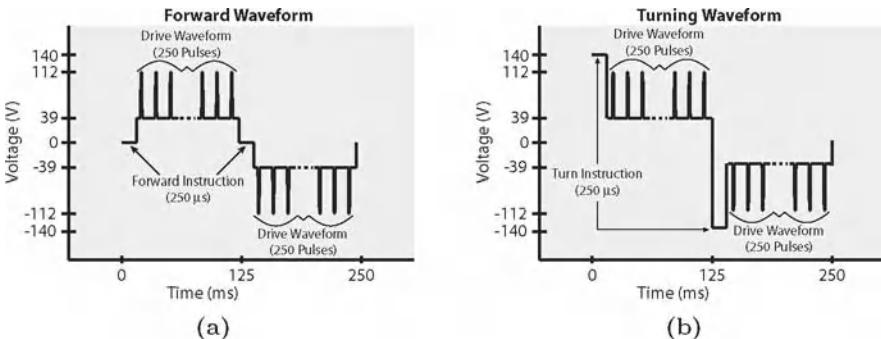


Fig. 4. Control waveforms used for driving the micro-robots at a stepping frequency of 4 kHz. **a:** The forward waveform lowers the device voltage to zero before initiating the drive waveform, ensuring that the steering arm will be in the raised position. **b:** The turning waveform increases the device voltage to 140 V (or -140 V) before initiating the drive waveform, ensuring that the steering arm will be in the lowered position. The polarity of the control waveform is reversed every 250 pulses to limit the effects of parasitic charging. The state of the steering arm is refreshed each time this occurs. In the control waveform segments shown here, the instructions are refreshed at 0 and 125 ms, when the polarity of the control waveform is reversed.

6 Performance

We tested the performance of the micro-robots in a variety of ways. First, we examined the reliability of the two motion primitives (forward motion and turning) with ten test runs of each motion primitive for each of five test devices. Second, we looked at how the radius of curvature can be controlled by time-sequence multiplexing the motion primitives. Third, we demonstrated teleoperated control of the devices by piloting them through clockwise and counter-clockwise rectangular paths. Last, we demonstrated device endurance by continuous operation in turning mode until accumulated error forced the device off of the operating environment.

This section quantifies results from 271 open-loop test runs of five devices, and presents representative segments of additional paths traversed during teleoperation. In all of these test runs, the devices were run under an optical microscope while recording their motions with a digital video camera.

Table 1. Turning rate of individual devices at 4 kHz stepping frequency.

Signal	Open-Loop Turning Rate (std. dev.) [degrees/mm]				
	Device 1	Device 2	Device 3	Device 4	Device 5
Forward :	-14.6 (7.8)	9.7 (9.6)	7.8 (14.9)	-21.7 (10.5)	0.4 (13.6)
Turning :	353 (4.9)	338 (4.6)	250 (9.7)	365 (6.5)	321 (8.1)

Device headings and positions were later extracted by image analysis with precision of $\pm 2^\circ$ and $\pm 1.6 \mu\text{m}$ respectively. The position of the device was defined at the center of the scratch drive plate, and its heading was defined by the orientation of the scratch drive bushing.

To test the forward motion, each of five devices was operated with the waveform shown in Figure 4a for 10 10-second trials with a 4 kHz stepping frequency, during each of which the devices traveled an average of $566 \mu\text{m}$. For consistency, all of these trials were run approximately parallel to the electrodes. The turning rate of an individual test run is defined by the slope of the best-fit line to the device heading over the course of the test run, plotted as a function of distance traveled.

Figure 5a shows turning rates for all ten forward test runs of one device. Over all devices, the average turning rate was $-3.7^\circ/\text{mm}$, with a standard deviation of $13.9^\circ/\text{mm}$. Average turning rates for individual devices are shown in Table 1, with standard deviations in parentheses. The errors shown accumulated open-loop, in the absence of an error-correcting scheme, and can be corrected through closed-loop control.

To test the turning motion, the devices were operated with the waveform shown in Figure 4b with a stepping frequency of 4 kHz for 10 trials of one full revolution each. Figure 5b shows the deviation from initial heading for all test runs of one of these devices. For all devices combined, the average turning rate was $325^\circ/\text{mm}$, which corresponds to a radius of curvature of $176 \mu\text{m}$. The standard deviation of the turning rate across all 4 kHz turning runs of all devices was $45.3^\circ/\text{mm}$. Standard deviations for individual devices are considerably lower, and appear in Table 1.

The forward and turning behaviors can be combined to produce turning radii with intermediate values. To demonstrate this, we drove a device with a signal composed of turning waveforms interleaved with forward waveforms, at a stepping frequency of 8 kHz. We tested waveforms with ratios of 50% turning and 75% turning, and compared these to the results of the test runs with 100% turning described above. There were ten full-revolution test runs at each of these turning ratios. Figure 5.c shows sample paths from tests runs executed at 50%, 75%, and 100% turning ratio, along with a plot of curvature vs. turning ratio averaged across all trials. The devices have been experimentally shown to achieve speeds in excess of $200 \mu\text{m/s}$, with an average step-size of 12 nm. (See Appendix D for more details)

With a human operator observing the device behavior, and controlling the waveforms sent to the device, it is possible to direct the devices through tele-

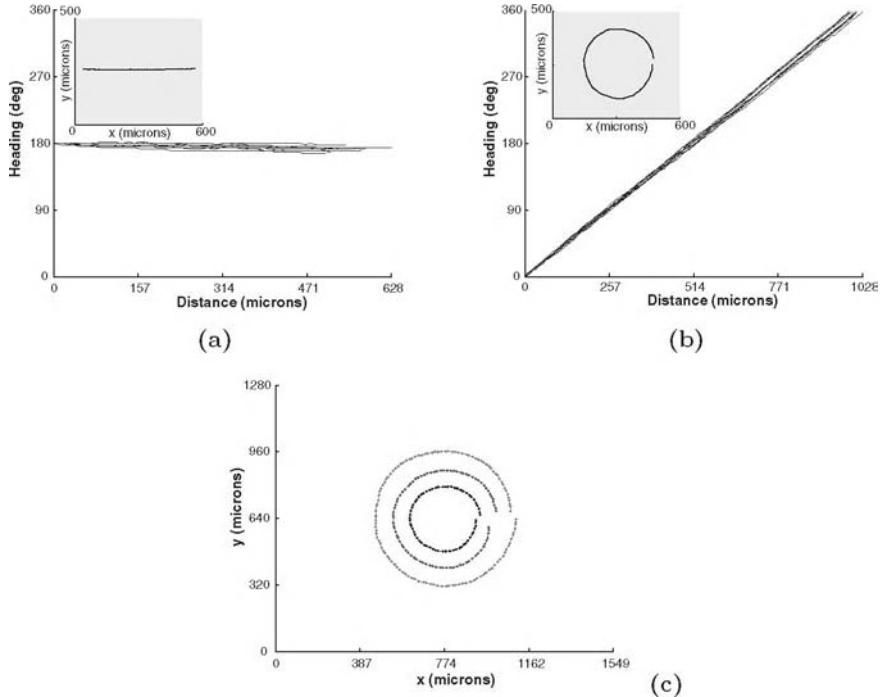


Fig. 5. Open-loop test runs of an electrostatic MEMS micro-robot. **a:** The change in device heading over the course of each of ten trials with the forward waveform at a stepping frequency of 4 kHz. The inset shows a representative path traversed by the device during one of the ten trials. The average turning rate for these trials was $-14.6^\circ/\text{mm}$. **b:** Device heading as a function of distance for each of ten trials with the turning waveform at a stepping frequency of 4 kHz. The inset shows a representative path. The average turning rate was $353^\circ/\text{mm}$, corresponding to a radius of curvature of $162 \mu\text{m}$. **c:** Representative paths traversed by a device with waveforms composed of different amounts of turning and forward control signals. In red: 50% turning. In blue: 75% turning. In black: 100% turning.

operation. Figure 6 shows clockwise and counter-clockwise rectangular paths traversed by one of these devices under teleoperated control. Digital videos of our devices are available on-line here [16].

For additional tests regarding the reliability of the device see Appendix E.

7 Conclusions

This paper presented an electrostatic MEMS micro-robot that is 1 to 2 orders of magnitude smaller than previous micro-robotic systems. This device was

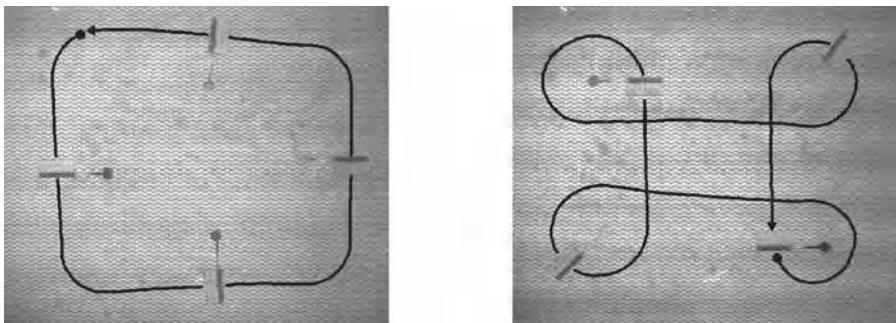


Fig. 6. Sample paths traversed by one of the micro-robots under teleoperated control. **Left:** Traversal of a counter-clockwise rectangular path by turning corners at minimum turning radius. **Right:** Clockwise paths were achieved by looping at the corners.

shown to perform in a robust and repeatable manner, and could be controlled through teleoperation to traverse complex paths.

The devices are powered through a capacitive coupling with an interdigitated electrode array, so that the devices need not be restricted by the wires and rails that power most electrostatic MEMS devices. Careful design of the mechanical structure of the micro-robot body allows the power signal to double as the control signal. The control information received from this signal is stored as electro-mechanical state information on-board the robot, so that the device can exhibit different behaviors in response to the same drive waveform, based on a previously-encoded state.

The communication and control system utilized in these micro-robots exploits electromechanical hysteresis to store state information within the micro-robot body, and is analogous to a four-state finite state machine.

Useful extensions to the complexity of the control system could include the ability to turn in both directions, to move in “reverse”, or to manipulate other objects in the environment. One particularly interesting extension would be the parallel operation of multiple micro-robots for cooperative tasks.

Acknowledgements

This work was supported by award number 2000-DT-CX-K001 to B.R.D., from the Office for Domestic Preparedness, Department of Homeland Security, USA. The electron micrographs were taken at the Ripple Electron Microscopy Laboratory, with the help of Chuck Daghlian. The vacuum microprobe used for device transport was developed with the help of Richard Rohl and the National Science Foundation’s Research Experience for Undergraduates program. We thank Ursula Gibson for the use of equipment in her lab, and for

many helpful discussions. We also thank Devin Balkcom, Karl Böhringer, Jessica Gomez, and Satish Prabhakaran for their advice and suggestions.

References

1. S. Baglio, S. Castorina, L. Fortuna, and N. Savalli. Development of autonomous, mobile micro-electro-mechanical devices. In *IEEE International Symposium on Circuits and Systems. Proceedings*, volume IV, pages 285–288, 2002.
2. P. Basset, A. Kaiser, P. Bigotte, D. Collard, and Lionel Buchaillot. A large stepwise motion electrostatic actuator for a wireless microrobot. In *Proceedings of the IEEE International Conference on Micro Electro Mechanical Systems*, pages 606–609, 2002.
3. Paul E. Kladitis and Victor M. Bright. Prototype microrobots for micro-positioning and micro-unmanned vehicles. *Sensors and Actuators A (Physical)*, A80(2):132–137, March 2000.
4. Richard Yeh, Ezekiel J. J. Kruglick, and Kristofer S. J. Pister. Surface-micromachined components for articulated microrobots. *Journal of Microelectromechanical Systems*, 5(1):10–17, March 2001.
5. S. Hollar, A. Flynn, C. Bellew, and K. S. J. Pister. Solar powered 10 mg silicon robot. In *The Sixteenth Annual International IEEE Conference on Micro Electro Mechanical Systems, Proceedings*, pages 706–711, January 2003.
6. Bruce R. Donald, Christopher G. Levey, Craig D. McGraw, Daniela Rus, and Mike Sinclair. Power delivery and locomotion of untethered micro-actuators. *Journal of Micro-Electromechanical Systems*, 12(6):947–959, December 2003.
7. Bruce R. Donald, Christopher G. Levey, Craig D. McGraw, Daniela Rus, and Mike Sinclair. Untethered micro-actuators for autonomous micro-robot locomotion: Design, fabrication, control, and performance. In *11th International Symposium of Robotics Research (ISRR), Proceedings*, pages 19–22, October 2003.
8. H. Hermes. On local and global controllability. *SIAM Journal of Control*, 12:252–261, 1974.
9. K. W. Markus, D. A. Koester, A. Cowen, R. Mahadevan, V. R. Dhuler, D. Roberson, and L. Smith. MEMS infrastructure: The multi-user MEMS processes (MUMPs). In *Proceedings of the SPIE - The International Society for Optical Engineering, Micromachining and Microfabrication Process Technology*, volume 2639, pages 54–63, 1995.
10. P. Dario, R. Valeggi, M. C. Carrozza, M. C. Montesi, and M. Cocco. Microactuators for microrobots: a critical survey. *Journal of Micromechanics and microengineering*, page 141, September 1992.
11. K. Kotay and D. Rus. Locomotion versatility through self-reconfiguration. *Robotics and Autonomous Systems*, 26:217, 1999.
12. Bruce R. Donald. *MEMS for Infosecurity*. Award Number 2000-DT-CX-K001, The Office for Domestic Preparedness, Department of Homeland Security, USA, 2000–2006.
13. S. Baglio, S. Castorina, L. Fortuna, and N. Savalli. Technologies and architectures for autonomous "MEMS" microrobots. In *IEEE International Symposium on Circuits and Systems. Proceedings*, volume II, pages 584–587, 2002.

14. Hiroyuki Miura, Takashi Yasuda, Yayoi Kubo Fujisawa, and Isao Shimoyama. Insect-model based microrobot. In *Transducers*, pages 392–395, June 1995.
15. Bruce Donald, Christopher Levey, Craig McGraw, Igor Paprotny, and Daniela Rus. A steerable, untethered, 250×60 micron MEMS mobile micro-robot. Technical Report TR2005-564, Dartmouth College, Computer Science Department, www.cs.dartmouth.edu/reports/abstracts/TR2005-564/, 2005.
16. Bruce Donald, Christopher Levey, Craig McGraw, Igor Paprotny, and Daniela Rus. An untethered, electrostatic, globally-controllable MEMS micro-robot. supplementary videos. Technical Report TR2005-553, Dartmouth College, Computer Science Department, www.cs.dartmouth.edu/reports/abstracts/TR2005-553/, 2005.

Appendix

Appendix **A** describes in detail the fabrication process used to manufacture the micro-robots and the electrical grids. In appendix **B** we provide an extended description of the power delivery mechanism. In appendix **C** we provide an analysis of the design parameters used to define the steering arm, and the steering waveform. Appendix **D** contains data on device speed and step-size. In appendix **E** we summarize a reliability test of prolonged device operation. Please note that the appendix contains an additional reference section.

A Details of the Fabrication Process

The first steps in fabricating the devices were performed through the PolyMUMPs process [9]. This multi-user surface micromachining process consists of three layers of polycrystalline silicon, separated by two sacrificial layers of phosphosilicate glass. The untethered scratch drive actuators and steering arms are both formed from the top layer of polysilicon, as shown in Figure 7.

The bushing is formed by combining the conformalities that result from the PolyMUMPs Dimple Etch and Via Etch, and is approximately $1.5 \mu\text{m}$ high. Similarly, the stylus at the end of the steering arm is formed from the Dimple Etch conformality, and is $0.75 \mu\text{m}$ high. After the PolyMUMPs process is complete, we coat the devices with a patterned layer of evaporated chromium. The tensile residual stress in the chromium curves the steering arms upwards. This curvature allows the arm to remain suspended above the substrate, even when sufficient voltage is applied to actuate the scratch drive.

The electrical grids used as operating environments for the devices were fabricated entirely in-house, and consist of an array of metal interdigitated electrodes on a silicon substrate. The electrodes are insulated from the substrate by a $3 \mu\text{m}$ - thick layer of thermal silica, and are coated with $0.5 \mu\text{m}$ of zirconium dioxide, followed by a 300 nm passivation layer of evaporated silica. This dielectric layer allows power delivery to devices placed on top of the electrical grids by capacitive coupling with the underlying electrodes. Fabrication of these electrical grids is illustrated in Figure 7. Once fabrication is complete, the devices are transferred onto the grids with a vacuum microprobe.

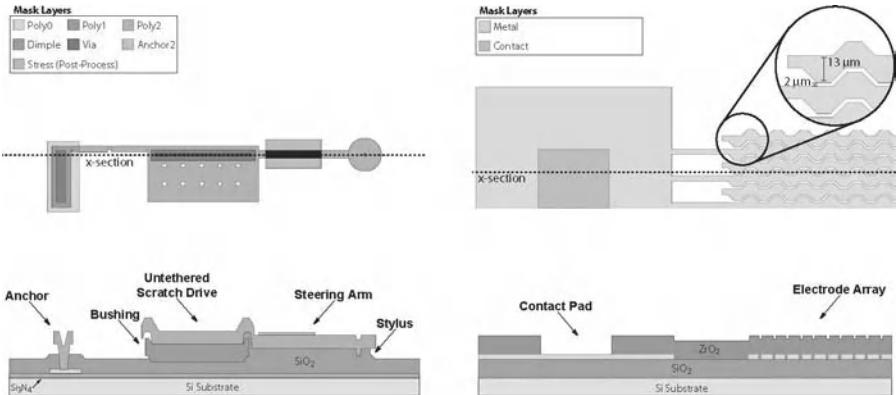


Fig. 7. Fabrication of electrostatic micro-robots. **Left:** Layout and cross-section of an electrostatic micro-robot just prior to sacrificial release. The device utilizes the first and second released polysilicon layers from the PolyMUMPs process [9], plus an additional layer added during post-processing. This layer consists of 830 nm of chromium with a tensile residual stress of approximately 550 MPa, and is lithographically patterned with the “Stress” mask (gray, above). After release, the stress in this layer curls the steering arm out of plane. **Right:** Layout and cross-section of the electrical grids used as the micro-robots’ operating environments. The first mask layer defines the metal electrodes, while the second layer defines contact holes through the electrode insulation. The metal electrodes are sandwiched between a layer of thermal silica, and a deposited layer of zirconium dioxide.

Processing details regarding the fabrication of the devices and the electrical grids are presented in Appendices A.1 and A.2.

A.1 Actuator Fabrication

Figure 7 shows the layout of one of the devices. The scratch drive plate is 1.5 μm thick, and is defined by a 120 μm by 60 μm rectangle on the third polysilicon layer (Poly2) of the PolyMUMPs process. The scratch drive bushing is 1.5 μm high, and is composed of a Dimple Etch conformality beneath a sheet of Poly1 that is anchored to the Poly2 layer with the Poly1-Poly2-Via etch. The steering arm is 133 μm long, 8 μm wide, and has an 18- μm -radius disc at its tip. A stylus is defined in the center of this disc by a 1.5- μm radius dimple that protrudes 0.75 μm beneath the bottom surface. The base of the steering arm is curled so that the tip of the arm is approximately 7.5 μm higher than the scratch drive plate. Since the PolyMUMPs process does not include a layer with enough stress to create this curvature, a layer of tensile chromium is deposited and patterned in the following post-processing sequence.

The devices are received from the foundry on 1-cm² silicon die. After the protective coating of photoresist is removed, the die are soaked in buffered hydrofluoric acid to under-etch the top polysilicon layer. This produces a re-entrant surface profile which enables lift-off of subsequent layers.

After rinsing and drying, the die are coated with 830 nm of chromium by thermal evaporation. As deposited, the chrome has an intrinsic tensile residual stress of approximately 550 MPa, which will produce the necessary curvature in the steering arms upon sacrificial release. The chrome is lithographically patterned with the “Stress” layer, shown in Figure 7, and etched in a perchloric-acid-based chrome etchant to transfer the pattern.

Once the chrome pattern has been defined, the sacrificial release etch is performed by soaking in 49% hydrofluoric acid. In addition to releasing the polysilicon structures, this undercuts the excess chrome and detaches it from the substrate. After rinsing in DI water, the die are dehydrated by soaking in isopropyl alcohol, and are then transferred to an ozone-friendly fluorocarbon solvent (based on 2,3-dihydrodecafluoropentane and isopropanol). Slow removal from this solvent ensures very little spotting or unnecessary stiction.

The devices are initially attached to substrate anchors by notched sacrificial beams. These beams are broken with a tungsten microprobe tip to release the devices, as described in [6, p. 951], prior to transferring them to the power-delivery substrates with a vacuum microprobe.

A.2 Substrate Fabrication

Figure 7 shows the layout of one of the electrical grids used as operating environments for the micro-robots. These grids consist of interdigitated metal electrodes microfabricated on oxidized silicon substrates. An insulating coating of zirconium dioxide provides a high-impedance dielectric coupling between the electrodes and the devices. Fabrication of these electrical grids was accomplished with the following process sequence.

The sequence begins with a set of 3-inch ⟨100⟩ n-type (phosphorus-doped) silicon wafers. The wafers are cleaned, and oxidized for 20 hours at 1100°C in oxygen, followed by an additional 14 hours of wet oxidation using water vapor in a nitrogen carrier gas.

After cooling, the wafers are patterned with the “Metal” pattern shown in Figure 7, using a bi-layer photodefinable resist suitable for liftoff. Metallization is then conducted by resistive boat evaporation at 10^{-6} Torr. Three metal layers are evaporated onto the patterned substrates. The middle layer consists of 500 nm of gold, and serves as the conductive bulk of the electrodes. Above and below this are two layers of chromium, each 50 nm thick, which serve as adhesion layers between the gold, the oxidized substrate, and the zirconium dioxide which will be subsequently deposited to insulate the electrodes.

After metallization, each wafer is cleaved into four 1-inch die. These die are sonicated in photoresist stripper at 45°C, to lift off the resists and unwanted metal, leaving only the interdigitated electrodes and their associated contact pads. To insulate the electrodes, the die are then coated with 5100 nm of zirconium dioxide, deposited by electron beam evaporation from zirconia powder according to the protocol described in [17]. The process used for zirconia deposition is critical to the quality of the dielectric, and to device performance. Since ZrO_2 dissociates during evaporation, it is important to facilitate re-

combination at the substrate surface. To do so, the substrates are heated to 100° C, and oxygen gas is introduced into the chamber to a pressure of 10^{-4} Torr. Throughout the deposition process, the chamber pressure is maintained to within $\pm 10\%$ by manually adjusting the oxygen flow. When the zirconia deposition is complete, the chamber is pumped back down to 2×10^{-6} Torr, and a 300 nm silica layer is then evaporated. We have found empirically that this over-layer of silica improves the walking performance of scratch drive actuators on zirconia-insulated substrates.

Once the die have been insulated, they are patterned with the “Contact” mask shown in Figure 7, and etched in a 5:1 buffered hydrofluoric acid solution for 5 minutes. This etches through the zirconia insulation, stopping on the contact pads.

After rinsing and drying, the substrates are ready for use. The devices are transferred to the electrode arrays by vacuum microprobe, and tungsten-tipped micro-probes are used to provide power to the interdigitated electrodes. In the next section, we discuss the delivery of electrical power from these insulated arrays of electrodes to untethered MEMS devices.

B Detailed Explanation of the Power Delivery Mechanism

When a conductive actuator, such as a scratch drive, rests on top of these electrodes, it forms the capacitive circuit shown in Figure 8. In this way, a voltage is applied to the actuator, regardless of its position and orientation relative to the underlying electrodes (i.e. no position-restricting wires or tethers are required.)

Once a voltage has been applied to a scratch drive actuator as described above, the actuator will deform as shown in Figure 8 [18,20]. Hayakawa *et. al.* [21] have calculated the length of the curved region of the scratch drive, ℓ , as follows:

$$\ell = \left(\frac{3Et^3h^2d}{4\kappa\epsilon_0 V^2} \right)^{\frac{1}{4}} \quad (1)$$

where κ is the dielectric constant of the insulator, ϵ_0 the permittivity of free space, h the bushing height, V the applied voltage, E the Young’s modulus of the plate material, d the insulator thickness, and t the thickness of the actuator plate. When the voltage is decreased, the flexure in the scratch drive plate relaxes, as shown in Figure 8. Each time the voltage is cycled, the scratch drive moves forward by a small increment, known as the *step size*. The frequency at which this cycle occurs is known as the *stepping frequency*, and the speed of the actuator is the product of its stepping frequency and its average step size.

Equation (1) shows that there is a trade-off between voltage, insulator thickness, and the relative permittivity of the dielectric. Hence, to improve device performance at a given voltage, we would like an insulator with a high

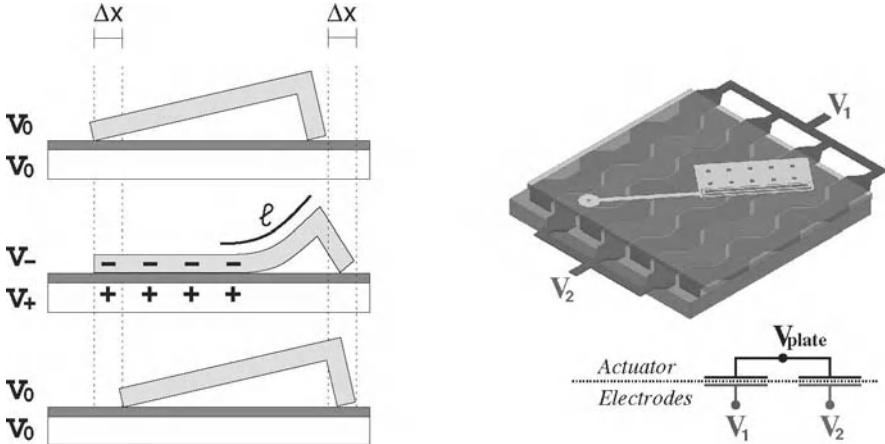


Fig. 8. **Left:** A schematic of the operation of a tethered scratch drive actuator [18,19]. The length of the curved region of the plate, ℓ , and the step size, Δx , are determined by the voltage. **Right:** A schematic of a capacitively-coupled power delivery mechanism for untethered actuators [6]. The potential induced on the actuator, V_{plate} , is approximately the mean of V_1 and V_2 .

value of κE_{br} , where E_{br} is the dielectric strength. For this reason, the high- κ dielectrics under investigation by the semiconductor industry [22] should also be good materials to use as the insulating layer in contact-mode electrostatic MEMS. One such material that performs well in this regard is e-beam-evaporated zirconium dioxide.

To deliver power to our MEMS micro-robots, we have used insulated substrates covered with zirconia-insulated gold electrodes. The edges of the electrodes are jagged (as shown in Figures 1 and 7) to help maintain a consistent voltage on the actuator, regardless of the actuator's pose. The electrodes are thin enough that the capacitance between adjacent electrodes is negligible, so the largest source of parasitic capacitance is between the electrodes and the underlying silicon substrate. This was sufficiently small for the purposes of the experiments conducted in this paper, and could easily be made much smaller by replacing the silicon substrate with an insulating material such as quartz, or with silicon-on-insulator (SOI) techniques. With these parasitic capacitances removed, the bulk of the delivered power can be focused only on those areas where a device is present.

C Design of the Steering Arm Actuator

The snap-down voltage of a cantilever beam is one of the earliest problems studied in the field of MEMS. First presented by Nathanson *et. al.* in 1967 [23], the electromechanical analysis of cantilever snap-down has since been

refined in numerous papers [24-27]. For simplicity, we will use Nathanson's model here.

Nathanson used a lumped energy minimization model to calculate the snap-down voltage of a cantilever beam as follows:

$$V_{SD} \approx \sqrt{\frac{8Kg_0^3}{27\epsilon_0 A}} \quad (2)$$

where K is the spring constant of the cantilever beam, g_0 is the zero-voltage gap between the cantilever and the electrode, and A is the total area of the cantilever. A similar analysis can be used to calculate the release voltage:

$$V_R \approx \sqrt{\frac{2Kg_1^2(g_0 - g_1)}{\epsilon_0 A}} \quad (3)$$

where g_1 is the contact gap between the cantilever and the electrode, as defined, for example, by a dimple.

These values are, of course, somewhat approximate, but they serve to illustrate the following interesting limitation. As mentioned earlier, we would like the micro-robot's steering arm to have a high snap-down voltage, and a low release voltage. In other words, we would like to be able to increase the ratio of the snap-down voltage to the release voltage. We'll call this the *snap ratio*. From the above two equations, the snap ratio is as follows:

$$R_s = \frac{V_{SD}}{V_R} \approx \sqrt{\frac{4g_0^3}{27g_1^2(g_0 - g_1)}} \quad (4)$$

We note that the snap ratio is independent of the beam's spring constant and area, but depends strongly on the gaps g_0 and g_1 .

Due to the largely planar nature of current microfabrication techniques, it can be difficult to parameterize z-axis geometries such as a cantilever's zero-voltage gap, without introducing a new processing step (e.g. a new material layer or etch mask) for each desired parameter value. One way to be able to parameterize these z-axis geometries is to deform parts out-of-plane using stress gradients of bi-layer materials. Tsai *et. al.* presented a general technique for controlling part curvature, using a top layer of silicon nitride with tensile residual stress [28,29]. We have adapted this approach for use with a chromium stress layer as described in Appendix A.1.

By curving the steering arm out-of-plane, we can increase the snap-down voltage well above the peak voltage of the scratch drive actuator's drive waveform, while keeping its release voltage well below the minimum of the drive waveform.

D Speed and Step-Size Data

Table 2 presents the speed of 5 devices using a stepping frequency of 2,4,8 and 16 KHz. The maximum recorded device speed was 224 $\mu\text{m/s}$, using a

Table 2. Speed of individual devices.

Signal	Open-Loop Speed (std. dev.) [$\mu\text{m/sec}$]				
	Device 1	Device 2	Device 3	Device 4	Device 5
Forward (4 kHz):	55 (4.4)	58 (2.7)	55 (5.9)	49 (4.0)	66 (2.6)
Turning (2 kHz):	25 (0.3)	28 (0.7)	24 (0.8)	16 (0.4)	29 (0.6)
Turning (4 kHz):	51 (0.8)	53 (0.5)	47 (1.2)	34 (1.0)	59 (0.7)
Turning (8 kHz):	80 (7.6)	93 (1.7)	97 (4.7)	70 (1.1)	105 (2.2)
Turning (16 kHz):	224 (2.9)	147 (1.7)	204 (7.4)	133 (1.3)	197 (6.0)

stepping frequency of 16 KHz. The step-size can be obtained by dividing the measured speed by the frequency of the drive waveform. The step-sizes for the experiments displayed in Table 2 range from 8 nm to 16 nm, with an overall average step-size of 12 nm.

E Reliability Testing of Device Operation

For all experiment described in Section 6 and below, the humidity was maintained at below 15% RH by a continuous stream of dry nitrogen. Drive waveforms were produced using an Agilent 33120A arbitrary waveform generator, and amplified with a Trek PZD700-1 high-voltage power amplifier with a gain of 200.

To test the reliability of the devices during prolonged operation, we operated one device until the point of failure. The device was piloted to the center of the operating environment, and the turning waveform was applied at a stepping frequency of 4 kHz. Over the course of the next seventy-five minutes, the device executed 215 full rotations, open-loop, without operator intervention, for a total distance traveled of over 35 centimeters. The device eventually stopped when accumulated position error forced it off of the 2.5-mm-wide operating environment. When the device was pushed back onto the operating environment with a microprobe, it continued to operate correctly.

Refs. 1-16 are in the main text on pages 12-13

References

- [17] J. A. Dobrowolski, P. D. Grant, R. Simpson and A. J. Waldorf, "Investigation of the Evaporation Process Conditions on the Optical Constants of Zirconia Films," *Applied Optics*, vol. 28, no. 18, pp. 3997–4005, 1989.
- [18] T. Akiyama and K. Shono, "Controlled Stepwise Motion in Polysilicon Microstructures," *Journal of Microelectromechanical Systems*, vol. 2, no. 3, pp. 106–110, 1993.
- [19] T. Akiyama, D. Collard and H. Fujita, "Scratch Drive Actuator with Mechanical Links for Self-Assembly of Three-Dimensional MEMS," *Journal of Microelectromechanical Systems*, vol. 6, pp. 10–17, 1997.

- [20] T. Akiyama, D. Collard and H. Fujita, "A quantitative analysis of Scratch Drive Actuator using buckling motion," in *IEEE International Conference on Micro Electro Mechanical Systems. Proceedings*, January 1995, pp. 310–315.
- [21] K. Hayakawa and A. Torii and A. Ueda, "An Analysis of the Elastic Deformation of an Electrostatic Microactuator," *Transactions of the Institute of Electrical Engineers of Japan, Part E*, vol. 118-E, no. 3, pp. 205–211, March 1998.
- [22] T. P. Ma, "Opportunities and Challenges for High-k Gate Dielectrics," in *2004 IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits. Proceedings*, pp. 1–4.
- [23] H.C. Nathanson, W. E. Newell, R. A. Wickstrom and J. R. Davis, "The Resonant Gate Transistor," *IEEE Transactions on Electron Devices*, vol. 14, pp. 117–133, 1978.
- [24] P. M. Osterberg and S. D. Senturia "M-TEST: A Test Chip for MEMS Material Property Measurement Using Electrostatically Actuated Test Structures," *Journal of Microelectromechanical Systems*, vol. 6, no. 2, pp. 107–118, 1997.
- [25] P. M. Osterberg "Electrostatically Actuated Microelectromechanical Test Structures for Material Property Measurement," Ph.D.dissertation, Massachusetts Institute of Technology, 1995.
- [26] S. Pamidighantam, R. Puers, K. Baert, and H. A. C. Tilmans, "Pull-in voltage analysis of electrostatically actuated beam structures with fixed-fixed and fixed-free end conditions," *Journal of Micromechanics and Microengineering*, vol. 12, pp. 458–464, October 2002.
- [27] K. Petersen, "Dynamic Micromechanics on Silicon: Techniques and Devices," *IEEE Transactions on Electron Devices*, vol. 25, no. 10, pp. 1241–1250, October 1978.
- [28] C.-L. Tsai and A. K. Henning, "Out-of-plane microstructures using stress engineering of thin films," in *Proceedings of Microlithography and Metrology in Micromachining.*, vol. 2639 ,1995, pp. 124–132.
- [29] C.-L. Tsai "Three dimensional stress engineered microstructures," Ph.D.dissertation, Thayer School of Engineering, Dartmouth College, 1997.

Some Issues in Humanoid Robot Design

Atsuo Takanishi¹, Yu Ogura², and Kazuko Itoh¹

¹ Department of Mechanical Engineering, Waseda University, Japan

² Advanced research institute for science and engineering, Waseda University, Japan

1 Introduction

Even though the market size is still small at this moment, applied fields of robots are gradually spreading from the manufacturing industry to the others in recent years. One can now easily expect that applications of robots will expand into the first and the third industrial fields as one of the important components to support our society in the 21st century. There also raises strong anticipations in Japan that robots for the personal use will coexist with humans and provide supports such as the assistance for the housework, care of the aged and the physically handicapped, since Japan is the fastest aging society in the world.

Consequently, humanoid robots and/or animaloid robots have been treated as subjects of robotics researches in Japan such as a research tool for human/animal science, an entertainment/mental-commit robot or an assistant/agent for humans in the human living environment.

Over the last couple of years, some manufactures started to develop prototypes or even to sell mass production robots for the purposes mentioned above, such as the SONY's pet robot AIBO and the small size humanoid robot QRIO, the TMSUK's tele-humanoid robot TMSUK04 and the TMSUK-SANYO's home utility robot ROBORIOR, the HONDA's humanoid robot ASIMO, the TOYOTA's partner humanoid robots, the NEC's information agent robot PaPeRo, etc. Most of those robots have some lifelikeness in their appearances and behaviors. Moreover, AIST, METI of Japan launched some national projects, such as Humanoid Research Project (HRP) in 1998 and the New Generation Robot Project in 2004 to develop humanoid robots and service robots, to accelerate the market growth of personal and service robots in the near future.

On the other hand, Waseda University, where we belong to, has been one of the leading research sites on humanoid robot research since the late Prof. Ichiro Kato and his colleagues started the WABOT (WAseDa roBOT) Project and produced the historically first humanoid robots WABOT-1 that could bipedal-walk in 1973 and the musician robot WABOT-2 that could play the electric organ in 1984. One

of the most important aspects of our research philosophy is as follows: By constructing anthropomorphic/humanoid robots that function and behave like a human, we are attempting to develop a design method of a humanoid robot having human friendliness to coexist with humans naturally and symbiotically, as well as to scientifically build not only the physical model of a human but also the mental model of it from the engineering view point.

Based upon the research philosophy mentioned above, we have been doing researches on humanoid robots, such as the Biped Walking Robots as WL(Waseda Leg) series and WABIAN(WAseda BIpedal humANoid) series, Mastication Robots as WJ(Waseda Jaw) series, Flute Player Robots as WF(Waseda Flutist) series, Emotion Expression Robots(Waseda Eye) series, Speech Production Robots as WT(Waseda Talker) series, etc. In this paper we introduce the mechanical design of the latest bipedal humanoid robot WABIAN-2 and the emotion expression humanoid robot WE-4RIII as shown in the Figure 1 and 2.

2 Bipedal Humanoid Robot WABIAN-2

In retrospect, many researchers have studied the control and mechanism of biped robots in recent years (Sakagami et al. 2002), (Nishiwaki et al. 2000), (Nishiwaki

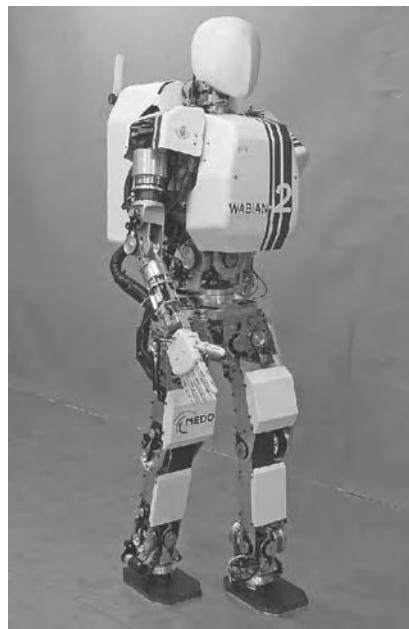


Fig. 1. Bipedal humanoid robot WABIAN-2



Fig. 2. Emotion expression humanoid robot WE-4RII

et al. 2002), (Löffler et al. 2003). These humanoid robots assimilated dynamic and stable walks. However, there are a few studies on human-like upper body. The Japanese National Institute of Advanced Industrial Science and Technology with cooperation of Kawada Industries, Inc., have developed HRP-2 and HRP-2P, which have 2-DOF trunk system and implement falling down motion in a positive way and rising from a lying position (Kaneko et al. 2002), (Fujiwara et al. 2003). This robot effectively bent its trunk in the experiments. The humanoid research group of Waseda University has also been studying biped humanoid robots since 1966. Research on the WABIAN (WA seda BIpedal humANoid) series had set walking with 3-DOF trunk motion and walking with 3-axis ZMP (Zero Moment Point) compensation using the trunk (Lim H et al. 1999), (Lim H et al. 2002). In advance of this study, we already have developed a new biped walking robot named WABIAN-2/LL (WA seda BIpedal humANoid-2 Lower Limb). Moreover, we have developed an algorithm that enables the robot to stretch its knees in steady walking avoiding singularity by using waist motion, and carried out stretch walking experiment by using this robot (Ogura Y et al. 2004), (Ogura Y et al. 2004). WABIAN-2/LL without upper limb originally developed as a lower limb system for a humanoid type robot WABIAN-2(WA seda BIpedal humANoid-2). In this chapter, we propose this new humanoid robot WABIAN-2 which has two 7-DOF legs, a 2-DOF waist, a 2-DOF trunk, and two 7-DOF arms. In the development of the robot, new design principle for a robot which can be used as a walking assist machine for a handicapped or elderly is set as the first goal of this study.

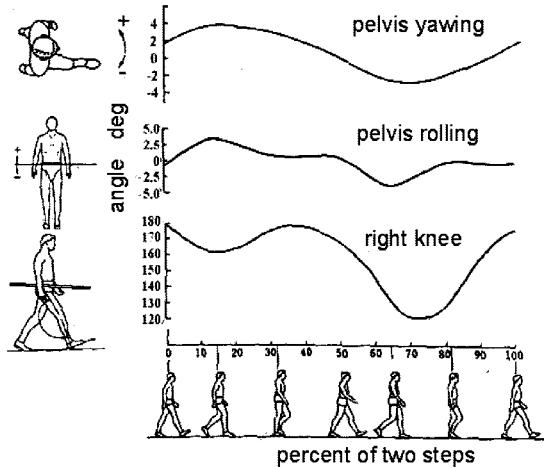


Fig. 3. Human's pelvis and knee motion (Klopsteg et al. 1963)

2.1 Design Concept

2.1.1 Human Motion

Human body mechanism basically comprises bones as rigid links, cartilage that lines the joints, muscles and tendons that actuate each part of the body. It is impossible to replace all of this muscular-skeletal system by current mechanical components.

Therefore, we determined that the primary goal of the mechanical design is to develop a robot that can imitate equivalent human motion.

Klopsteg et al. have proposed the result of the gate analysis of humans (Klopsteg et al. 1963). Figure 3 shows the pelvis and the knee motion plotted in the steady walking phase. The data is based on experimental results of 8 people walking motion who do not have physical or mental handicaps. In the result, human's pelvis motion in steady walking is observed in frontal plane (defined as roll motion in this study) and horizontal plane (defined as yaw motion). Waist motion in side plane (defined as pitch motion) is seldom observed. According to this a humanoid robot which can perform walks similar to human should be able to move its hips in the roll and yaw axes. These hip movements have to be independent in its trunk position.

Moreover, a study of gait analysis and bio mechanics has reported about pelvis motion. In steady walking, Pubic symphysis, the two hipbones combined by a cartilage, moves like a crank joint. According to this motion, the two hipbones are

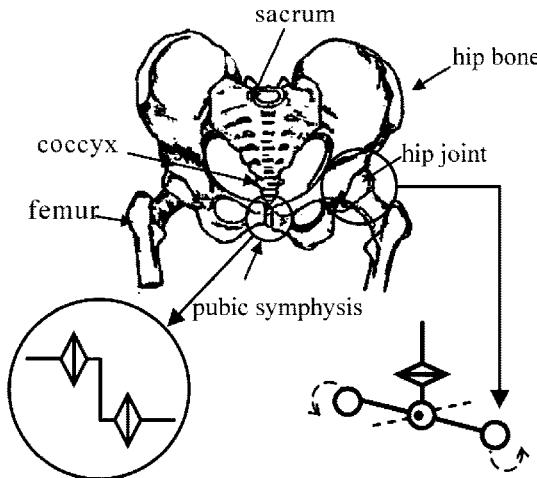


Fig. 4. Pelvis of a human

sliding on each other. Therefore, we considered that the hip joint is able to make two dimensional circular motions as shown in Fig.4.

On the other hand, human can move its trunk independently from the hip motion. The Japanese Association of Rehabilitation Medicine (JARM) and the Japanese Orthopaedic Association (JOA) have established basic roles of representation and measurement method for range of motion (ROM) (Klopsteg et al. 1963). The general idea of ROM does not always mean joints or articulation. These ROM measurements had carried out in a sitting position or with instruments to fix the pelvis in order to avoid the pelvis movement.

It is essential for a human motion simulator to have the ability to move its trunk. For example, trunk motions are used for rising a sitting position, walking with a limp, walking with movements are useful, not only for keeping the whole body balance, in other words, compensate motion for ZMP on contact ground, but also for absorption mechanism of positional error in the case that the robot grasps or leans against something on the ground. When the robot leans against a rail or use a walker or a walking assist machine, the system composed to the robot and the instrument becomes a statically indeterminate structure. Such a system will need some redundant DOF and a robust control method. It is considered that a human usually use its trunk motions unconsciously in these cases.

2.1.2 DOF Configuration

Figure 5 presents the DOF configuration of the new humanoid robot having 41 DOFs in Total. In this study, the initial pose of the robot is defined as standing straight, and rotational direction of each joint is defined by using inertial coordinate system fixed on the ground as shown in Fig. 5 (Ogura et al. 2004).

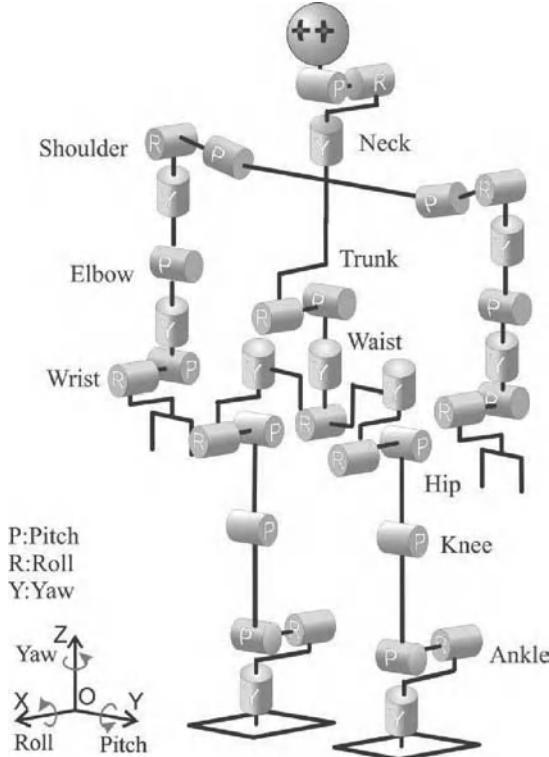


Fig. 5. DOF configuration of WABIAN-2 having 41 DOFs in total

WABIAN-2 has 7-DOF legs and arms like a human. Although many researchers have studied 7-DOF arms, there are few studies on mechanisms and control method of 7-DOF legs. The ankles of almost all conventional biped humanoid robots consist of the pitch and the roll axes. If the ankle is composed of pitch, roll and yaw joint, the biped robot can select a stable position and reduce the impact and/or contact forces produced between the landing foot and the ground using a proper control algorithm. Moreover, this leg system has an advantage in generating diverse walking patterns by using the leg redundancy. Biped robots which have only 6-DOF legs allow a unique knee orientation when position and orientation of those foot and waist are set. On the other hand, a biped robot which has 7-DOF legs can rotate knee orientation independently from foot trajectory. Therefore, this leg system will be useful when avoiding obstacles; for example, climbing a ladder up and down, riding on something, working in a narrow place and so on.

In 2-DOF waist system, the roll axis and yaw axis should be perpendicular to each other, and crossing the middle point between the two hip joints. This will result in minimizing the displacement of the trunk by waist motion and simplifying the kinematics calculation. In addition, the roll joint should be laid on the lower limb side and the yaw joint on the trunk side. This makes the yaw joint able to be used

as yaw rotation for both the hips and the trunk. This DOF configuration of waist and trunk gives substantial 3-DOF trunk motions.

2.2 Mechanisms

2.2.1 Overview

The whole mechanical design was done by using a 3D CAD software, SolidWorks 2003. The frameworks of WABIAN-2 are mainly made of duralumin in order to realize antithetical concepts; light weight, high stiffness and wide movable range. Each actuator system of joint consists of a DC motor, a Harmonic drive gear, a lug belt and two pulleys. This double speed reduction mechanism allows high reduction ratio, and also a joint axis to be set apart from a motor axis. Therefore, we could design a human-like joint mechanism without a big projection. In this paper, we mainly focus on the development of the waist, trunk and arms.

Specifications of each joint such as maximum torque and rotating speed are designed based on results of software simulations. Those results were computed by using Newton-Euler's Method and estimated mass distribution. The several types of the simulations were carried out for the determination of the joint specification. The details are described as follow.

2.2.2 Waist and Trunk

Figure 6 and 7 show the 2-DOF waist and 2-DOF trunk system. 2-DOF waist combination of a roll and a yaw joint is attached on the middle between the hip joints. 2-DOF trunk having a pitch and a roll joint is assembled over the waist.

In the design of the trunk some simulations have been conducted. The simulation tested the maximum torque for the trunk roll and pitch joint. During each walking step the robot moves its trunk in a way that can keep it balance. There are two type

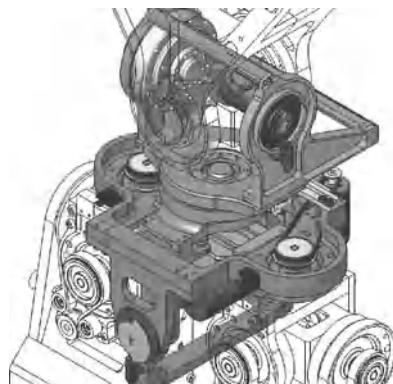


Fig. 6. Waist mechanism

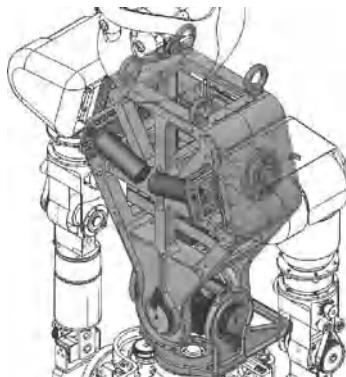


Fig. 7. Trunk mechanism

of simulation. First, static modeling which determined the maximum angle. Second, dynamic modeling which calculated the angle during the time used to complete one walking cycle. The maximum angle that can be determined by dynamic modeling is half the maximum movable angle determined by static modeling. Figure 8 shows a static model of the trunk in form of link and mass block. There are two models, one from side plane (Pitch axis) and other from front plane (Roll axis).

2.2.3 Arms

The arm of WABIAN-2 has 7-DOF. Figure 9 shows the 3D-CAD. The arms were designed in such a way that can support the robot balance while it is walking. It includes three actuators for fingers that can bend like human's hand fingers. Moreover, the arms were designed to hold the robot weight while it leans on a walking assist machine.

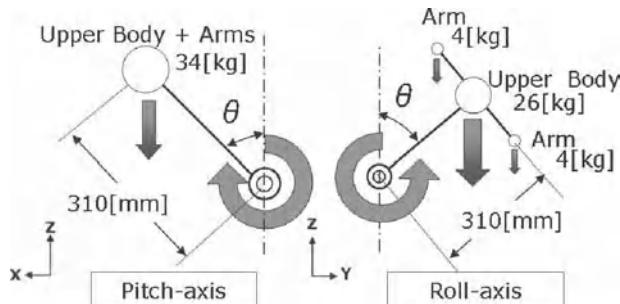


Fig. 8. Static Mechanics Model for Trunk Design

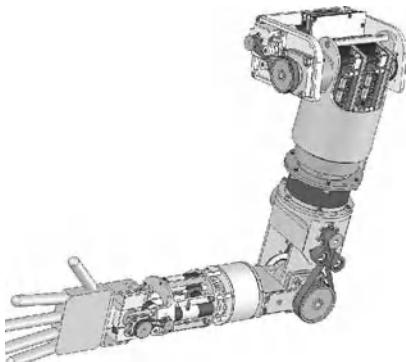


Fig. 9. Left Arm Mechanism

Since the robot could lean on a walking assist machine, most of the robot weight will become on both the elbow. In order to determine the suitable angles of the arms some software simulation were conducted. Figure 10 shows software simulations for determination of specification of upper limb mechanisms. In this simulation, the robot leans on a walking assist machine using its forearms. The elbow angle is 15deg from a posture bend at right angle, when the arm supports a half weight of the robot (30kg, the two arms support a whole weight of the robot (60kg)).

3 Emotion Expression Humanoid Robot WE-4RII

Humans take a certain posture in their communication. For example, when they are happy or cheerful, they take a posture in which the activity is high such as moving arms upward or opening the arms. When they are angry, they square the

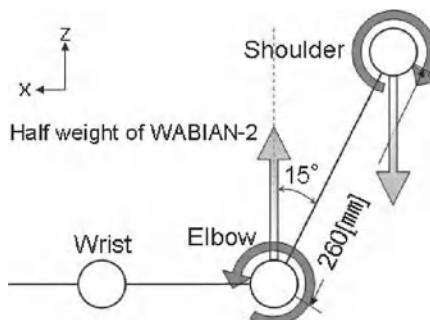


Fig. 10.

shoulders. When they are tired or sadness, they shrug the shoulder or close the arms (Hama et al. 2001). Therefore, the emotion and mental state are closely related to the human posture and behavior. And, human obtain many information from partner's posture in their communication. In this situation, the human arms play an important role.

We can control the usual 6-DOFs robot arms' tip position as accurately as human's arm. But, all their joint angles are fixed according to the inverse kinematics. By the way, humans have 7-DOFs arms consisting of 3-DOFs shoulder, 1-DOF elbow and 3-DOFs wrist. However, we considered that there is a center of rotation in the base of shoulder, and the shoulder joint itself moves up and down and moves back and forth so that humans square and shrug their shoulders. We considered that these motions played a very important role in the emotional expressions. Therefore, we tried to develop more emotional expressive humanoid robot arms than the usual 6-DOFs robot arms.

3.1 9-DOFs Emotion Expression Humanoid Arm

Figure 11 shows the 9-DOFs Emotion Expression Humanoid Arm developed in 2003 (Miwa et al. 2002), (Miwa et al. 2003), (Miwa et al. 2004). It has 2-DOFs at the base shoulder, 3-DOFs at the shoulder, 1-DOF at the elbow and 3-DOFs at the wrist. The robot arm can move each joint as widely as human for the more human-like emotional expression. Moreover, we designed the new robot to have the same dimension as the averaged male for the natural appearance using a 3D CAD software, SolidWorks 2003, like the WABIAN-2 design.

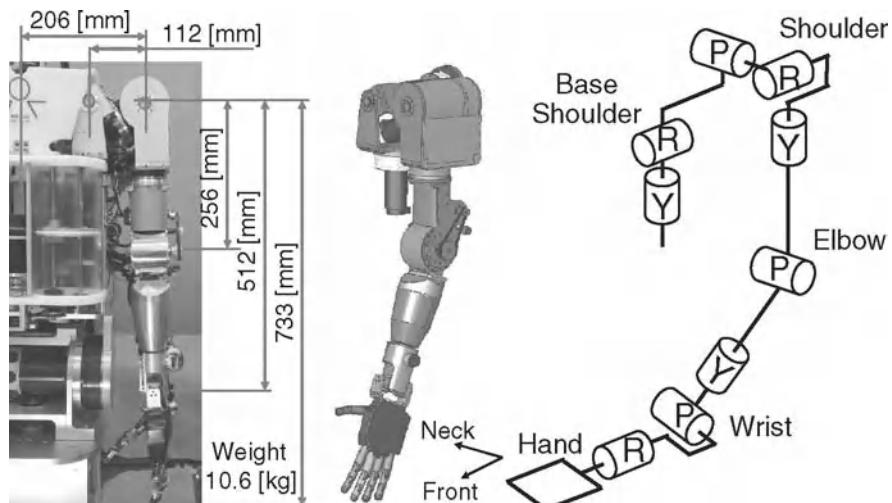


Fig. 11. 9-DOFs Arm Mechanism of WE-4RII and Its DOF Configuration

3.1.1 Base Shoulder

The base shoulder consists of the yaw axis and roll axis. They cross at the right angle. The yaw axis is driven by a direct driven mechanism with DC motors and harmonic drive systems. On the other hand, the roll axis is driven by a direct driven mechanism with an AC motor and a harmonic drive system because the roll axis needs the higher torque than the other axis to lift up the robot arm.

3.1.2 Shoulder

The shoulder has the pitch, roll and yaw axes. All axes are driven by a direct driven mechanism with DC motors and harmonic drive systems. In the case that the three shoulder axes cross at the identified position at the right angle, the posture where a robot horizontally stretches its arm is the singular point. So, we can't solve the inverse kinematics geometrically. However, the posture which the arm is lengthened just beside can be taken in everyday action. Therefore, we leaned the pitch axis 30 [deg] from the horizontal plane as shown in Figure 12 in order to reduce to move the arm to the singular point problem in a common use range. However, this mechanism couldn't avoid singular point problem completely. So, we avoid moving the tip of the arm to the singular point by software.

3.1.3 Elbow

The elbow has 1-DOF. In order to reduce the sense of incongruity on appearance realizing the same movable range with human, we adopted a belt driven mechanism, in which an output axis of a motor connects with a harmonic drive system by a timing belt.

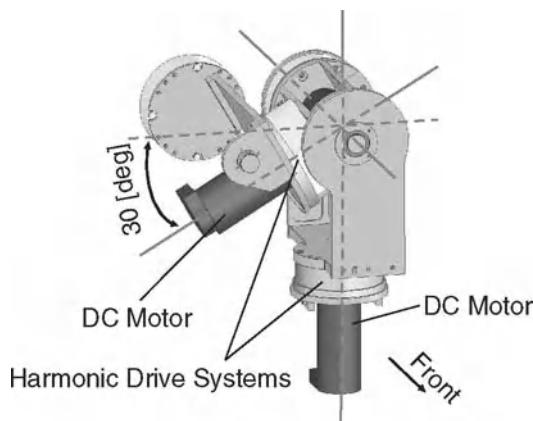


Fig. 12. Shoulder Mechanism of WE-4RII

3.1.4 Wrist

The wrist has the pitch, roll and yaw axes. They cross at the identified point at the right angle. The pitch axis is driven by a belt driven mechanism and the yaw and roll axes are driven by direct driven mechanism with DC motors paired with planetary gears.

3.1.5 Hand

The hands called RCH-1 (RoboCasa Hand No. 1) were designed in an international collaboration at RoboCasa which was established in 2003 between Waseda University in Japan and Scuola Superiore Sant' Anna in Italy (Zecca et al. 2004). RCH-1 is an under actuated hand having 6 DOFs of Motions while having 16 degrees of kinematical degrees.

3.2 Integration to Humanoid Robot WE-4RII

We developed the whole Emotion Expression Humanoid Robot WE-4RII shown in Figure 2 by integrating the 9-DOFs Emotion Expression Humanoid Arms and the 6-DOFs Humanlike Hands RCH-1 into the Human-like Head Robot WE-4. WE-4RII is 0.97 [m] tall and weigh 59.3 [kg]. And, it has 59-DOFs in total shown in Table 1. By adding the arms and the hands, WE-4RII could express its emotion

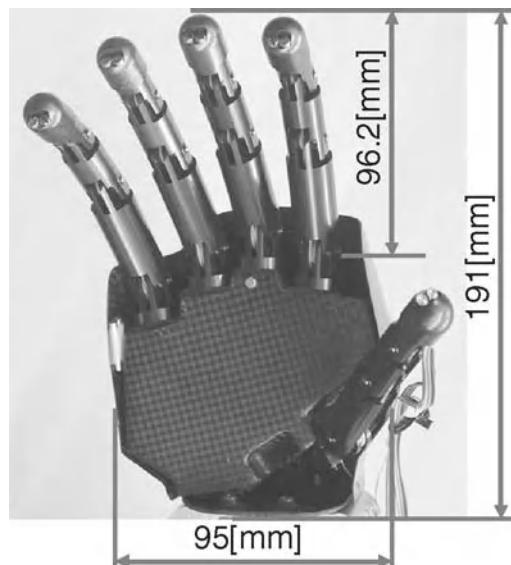


Fig. 13. RCH-1 Hand

with not only the facial expression but also the upper-half body including the waist, arms, hands and neck. Moreover, the motion velocity is as important as the posture in emotional expression. Therefore, we controlled both the posture and the motion velocity for the effective emotional expression. Figure 14 shows the emotional expression exhibited by WE-4RII.

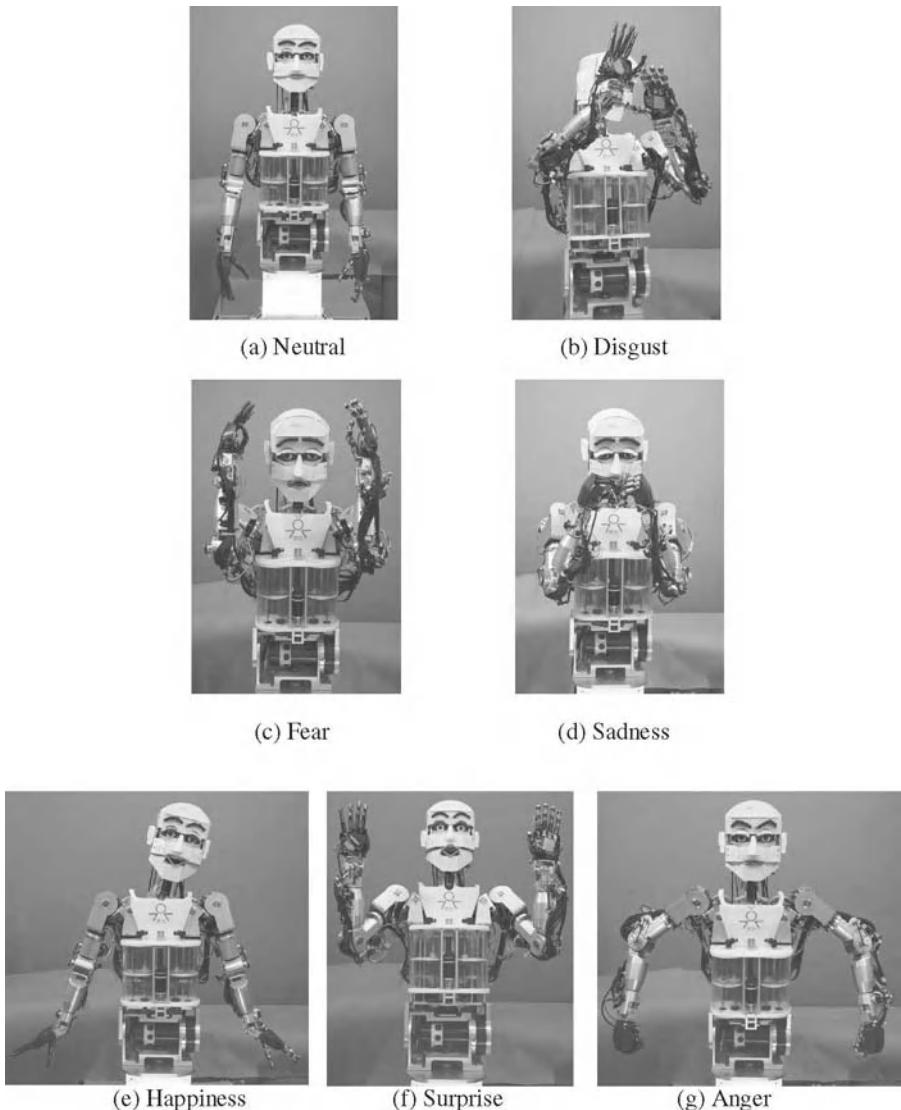


Fig. 14. Presented seven basic emotions by WE-4RII

Table 1. DOFs of WE-4RII

Part	DOF
Neck	4
Eyes	3
Eyelids	6
Eyebrows	8
Lips	4
Jaw	1
Lungs	1
Waist	2
Arms	18
Hands	12
Total	59

Weight: 59.3 [kg]

4 Conclusions and Future Work

This paper describes how we designed the two humanoid robots WABIAN-2 and WE-4RII. WABIAN-2 has 7-DOF legs, a 2-DOF waist, a 2-DOF trunk, and 7-DOF arms. In the development of the robot, new design principle for a robot which can use walking assist machine is proposed. In the near future, we shall propose a hardware simulator system capable of being applied to the evaluation of welfare machines or robots. In order to demonstrate the validity of the proposal, we are presently preparing an experiment in which a biped humanoid robot uses a walking assist machine. The measurements of the current or force/torque sensors will present a quantitative clarification of the manner in which the machine assists humanoid walking. We also designed the 9-DOFs Emotion Expression Humanoid Arms as well as the 6-DOFs RCH-1s, and integrated them into the Emotion Expression Humanoid Robot WE-4R. We also have developed an emotion expression control method for WE-4RII and that was presented in IROS 2004. In the future, we shall increase the emotional expression patterns and robot behaviors. And, we also shall introduce the behavior model which autonomously determines and outputs the most suitable behavior or emotional patterns according to the situation which is one of the essential functionalities of an intelligent robot to interact with humans.

Acknowledgment

A part of this research was conducted at the Humanoid Robotics Institute (HRI), Waseda University. This work is partially supported by NEDO (the New Energy and Industrial Technology Development Organization), Japan. The authors would like to express their gratitude to TMSUK Inc., Okino Industries Ltd., Osada Electric Co., Ltd., Sony Corporation, Sanyo Electric Co., Ltd. and ZMP Inc. for extending financial support to HRI. This research was supported in part by a Grant-in-Aid for the WABOT-HOUSE Project by the Gifu Prefecture and the JSPS Research Fellowships for Young Scientists. The authors also would like to thank Italian Ministry of Foreign Affairs, General Directorate for Cultural Promotion and Cooperation, for its support to the establishment of the RoboCasa laboratory and for the realization of the two artificial hands. We would also like to express our best gratitude to SolidWorks Japan K.K. for their support.

References

- Sakagami Y et al. (2002) The intelligent ASIMO: System overview and integration. In: Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, pp 2478-2483
- Nishiwaki K et al. (2000) Design and Development of Research Platform for Perception-Action Integration in Humanoid Robot: H6. In: Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, pp 1559-1564
- Nishiwaki K et al. (2002) Online Generation of Humanoid Walking Motion based on a Fast Generation Method of Motion Pattern that Follows Desired ZMP. In: Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, pp 2684-2689
- Löffler K et al. (2003) Sensor and Control Design of a Dynamically Stable Biped Robot. In: Proc. IEEE Int. Conference on Robotics and Automation, pp 484-490
- Kaneko K et al. (2002) Design of Prototype Humanoid Robotics Platform for HRP. In: Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, pp 2431-2436
- Fujiwara K et al. (2003) The First Human-size Humanoid that can Fall Over Safely and Stand-up Again. In: Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, pp 1920-1926
- Lim H, Ishii A, Takanishi A (1999) Motion Pattern Generation for Emotion Expression. In: Proc. Int. Symp. Humanoid Robots, Tokyo, Japan, Oct. 1999, pp 36-41
- Lim H, Kaneshima Y, Takanishi A (2002) Online Walking Pattern Generation for Biped Humanoid with Trunk. In: Proc. IEEE Int. Conference on Robotics and Automation, Washington, DC., USA, May 2002, pp 3111-3116
- Ogura Y et al. (2004) Development of a Human-like Walking Robot Having Two 7-DOF Legs and a 2-DOF Waist. In: Proc. IEEE Int. Conference on Robotics and Automation, pp 134-139
- Ogura Y et al. (2004) A Novel Method of Biped Walking Pattern Generation with Predetermined Knee joint Motion. In: Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 2831-2836
- Klopsteg P et al. (1963) Human Limbs and Their Substitutes, New York Hafner
- Ogura Y et al. (2004) A Novel Method of Biped Walking Pattern Generation with Predetermined Knee joint Motion. In: Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 2831-2836
- Hama H et al. (2001) Kanjo Shinrigaku heno Syoutai (in Japanese). Saiensu: pp 162-170

- Miwa H et al. (2002) Development of a New Human-like Head Robot WE-4. In: Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 2443-2448
- Miwa H et al. (2003) Introduction of the Need Model for Humanoid Robots to Generate Active Behavior. In: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1400-1406
- Miwa H et al. (2004) Design and Control of 9-DOFs Emotion Expression Humanoid Arm. In: Proceedings of the 2004 IEEE International Conference on Robotics & Automation, pp 128-133
- Zecca M et al. (2004) On the development of the Emotion Expression Humanoid Robot WE-4RII with RCH-1. HUMANOIDS 2004

Underwater Robotics

That Which Does Not Stabilize, Will Only Make Us Stronger

H. Kazerooni and R. Steger

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

Abstract. The Berkeley Lower Extremity Exoskeleton (BLEEX) is a load-carrying and energetically autonomous human exoskeleton that, in this first generation prototype, carries up to a 34 kg (75 lb) payload for the pilot and allows the pilot to walk at up to 1.3 m/s (2.9 mph). This article focuses on the human-in-the-loop control scheme and the novel ring-based networked control architecture (ExoNET) that together enable BLEEX to support payload while safely moving in concert with the human pilot. The BLEEX sensitivity amplification control algorithm proposed 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 tradeoffs between not having sensors to measure human variables, the need for dynamic model accuracy, and robustness to parameter uncertainty are described. ExoNET provides the physical network on which the BLEEX control algorithm runs. The ExoNET control network guarantees strict determinism, optimized data transfer for small data sizes, and flexibility in configuration. Its features and application on BLEEX are described.

1 Introduction

The goal of the exoskeleton 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 generation 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 his/her agility, physical effectiveness increases significantly with the aid of this class of lower extremity exoskeletons.

BLEEX has numerous potential applications; 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. 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.

The effectiveness of the lower extremity exoskeleton is a direct result of the control system's ability to leverage the human intellect to provide balance, navigation, and path-planning while ensuring that the exoskeleton actuators provide most of the strength necessary for supporting payload and walking. In operation, the exoskeleton becomes transparent to the pilot and there is no need to train or learn any type of interface to use the robot. The control algorithm ensures that the exoskeleton always moves in concert with the pilot with minimal interaction force between the two and was first presented in [1].

The control scheme needs no direct measurements from the pilot or the human-machine interface (e.g. no force sensors between the two); instead, the controller estimates, based on measurements from the exoskeleton structure only, how to move so that the pilot feels very little force. This control scheme 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 em-

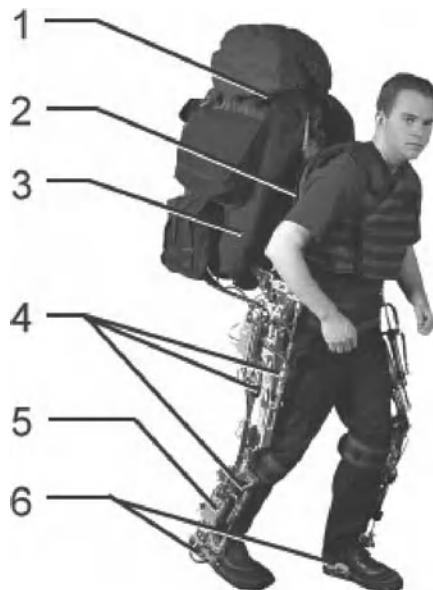


Fig. 1. Berkeley Lower Extremity Exoskeleton (BLEEX) first generation prototype and pilot. 1: Hydraulic power supply and payload occupy the upper portion of the backpack; 2: Rigid BLEEX spine connected to the flexible pilot vest; 3: Central control computer occupies the lower portion of the backpack; 4: Some of the hydraulic actuators (ankle, knee and hip); 5: Two of the control network's Remote I/O Modules (RIOM); 6: Rigid connection of the BLEEX feet to the pilot's boots. More photographs can be found at <http://bleex.me.berkeley.edu>

ployed for upper extremity exoskeletons [2], and haptic systems [3,4] 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 ex-

ternal forces and torques would respond to other external forces not initiated by its pilot. 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. step 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.

Realization of this control scheme requires a high-performance physical control architecture. This paper presents the ring-based protocol and distributed networked control hardware called the ExoNet. Traditional centralized control architectures where a supervisory controller directly interfaces in a point-to-point fashion with all sensors and actuators in the system have been successfully implemented in the past. They are generally feasible when a controller interfaces with small number of sensors and actuators and requires short wiring to them. Larger sophisticated multi-degree-of-freedom systems frequently require the control network to be compact, easily reconfigurable, expandable, and maintainable. Hence, we utilized a networked control system (NCS) as an alternative to the conventional centralized control system because of its advantages in flexibility, volume of wiring and capacity of distribution [5].

2 Previous Work

In our research work at U.C. Berkeley, we have divided 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 separation, 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 inte-

grate them. See [6,7] for research work on upper extremity exoskeletons at Berkeley.

The concept of a powered human-assistive exoskeleton has been explored in various reach projects since the 1950's, though recent advances in controls and computation have generated renewed interest (see [8-11]). A recent notable project is the "RoboKnee," which is a powered knee brace that functions in parallel to the wearer's knee but does not transfer loads to the ground [12]. This device transfers the weight of the backpack payload onto the human skeleton (including shanks, ankles, and feet). "HAL", a walking aid system for individuals with gait disorders, is another current exoskeleton-like device in that it adds to the force generated by the human muscles but relies on the human skeleton to transfer loads [13]. BLEEX draws on this history of exoskeleton development but is unique in that it mechanically functions as a true load bearing exoskeleton, is energetically autonomous, and utilizes a unique control system that does not require any direct measurements on the human.

Networked control systems (NCSs), such as the one developed for BLEEX, have broad applications beyond just exoskeleton control and have been adopted in fields related to industrial automation, building automation, office and home automation, intelligent vehicles, aircrafts, and space-crafts [14-18]. Several network types had been developed based upon the applications, such as a process field bus (PROFIBUS) [19], manufacturing automation protocol (MAP) [20], and fiber distributed data interface (FDDI) [21]. EtherNet, ControlNet, and DeviceNet are some other common NCSs that are compared as control networks in different situations and schemes in [5].

3 Sensitivity Amplification Controller

3.1 A Simple One Degree-of-Freedom (DOF) Example

The control of the exoskeleton is explained here through the 1 DOF example shown in Fig. 2. This figure schematically depicts a human leg attached and interacting with a 1 DOF exoskeleton leg in a swing configuration (no interaction with the ground). For simplicity, the exoskeleton leg is shown as a rigid link pivoting about a joint and powered by a single actuator. The exoskeleton leg in this example has an actuator that produces a torque, r , about pivot point A .

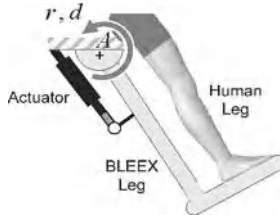


Fig. 2. One DOF conceptual model of an exoskeleton leg interacting with the pilot's leg.

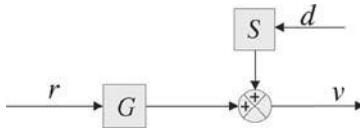


Fig. 3. The exoskeleton's angular velocity shown as a function of the input to the actuators, r , and the torques imposed by the pilot onto the exoskeleton, d .

Although the pilot is attached securely to the exoskeleton at the foot, other parts of the pilot leg, such as the shanks and thighs, can contact the exoskeleton and impose forces and torques on the exoskeleton leg. The location of the contacts and the direction of the contact forces (and sometimes contact torques) vary and are therefore considered unknown values in this analysis. In fact, one of the primary objectives in designing BLEEX was to ensure a pilot's unrestricted interaction with BLEEX. The equivalent torque on the exoskeleton leg, resulting from the pilot's applied forces and torques, is represented by d . Fig. 3 presents the system dynamics in block diagram form.

In the case where multiple actuators produce controlled torques on the system, r is the vector of torques imposed on the exoskeleton by the actuators. G is the transfer function from the actuator input, r , to the exoskeleton angular velocity, v (actuator dynamics are included in G). The form of G and the type of internal feedback for the actuator is immaterial for the discussion here. The Laplace operator has been omitted in all equations for the sake of compactness.

The sensitivity transfer function S , represents how the equivalent human torque affects the exoskeleton angular velocity. S maps the equivalent pilot torque, d , onto the exoskeleton velocity, v . If the actuator already has some sort of primary stabilizing controller, the magnitude of S will be small and the exoskeleton will only have a small response to the imposed forces and torques from the pilot or any other source. For example, a high gain velocity controller in the actuator results in small S , and consequently a small exoskeleton response to forces and torques. Also, non-

backdrivable actuators (e.g. large transmission ratios or servo-valves with overlapping spools) result in a small S which leads to a correspondingly small response to pilot forces and torques.

The resulting torque from pilot on the exoskeleton, d , is not an exogenous input; it is a function of the pilot dynamics and variables such as position and velocity of the pilot and the exoskeleton legs. These dynamics change from person to person, and within a person as a function of time and posture. These dynamics will be added to the analysis in later paragraphs, but they are unrelated to the purpose of current discussion.

Our goal is to develop an exoskeleton with a large sensitivity to forces and torques from the operator using measurements only from the exoskeleton (i.e. no sensors on the pilot or the exoskeleton interface with the pilot).

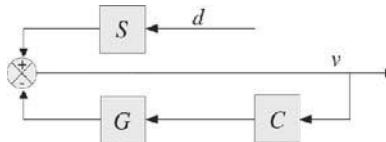


Fig. 4. Negative feedback control loop added to block diagram of Fig. 2 where C is the controller and it operates only on exoskeleton variables.

Creating a negative feedback loop from the exoskeleton variables only, as shown in Fig. 4, the closed-loop sensitivity transfer function is presented in Eq. (1).

$$S_{NEW} = \frac{v}{d} = \frac{S}{1 + GC} \quad (1)$$

Observation of Eq. (1) reveals that $S_{NEW} \leq S$, and therefore any negative feedback from the exoskeleton leads to an even smaller sensitivity transfer function. With respect to Eq. (1), our goal is to design a controller for a given S and G such that the closed loop response from d to v (the new sensitivity function as given by Eq. (1)) is greater than the open loop sensitivity transfer function (i.e. S) within some bounded frequency range. This design specification is given by inequality (2)

$$|S_{NEW}| > |S| \quad \forall \omega \in (0, \omega_0) \quad (2)$$

or alternatively

$$|1 + GC| < 1 \quad \forall \omega \in (0, \omega_0) \quad (3)$$

where ω_0 is the exoskeleton maneuvering bandwidth.

Exoskeleton control requires a totally opposite goal from classical and modern control theory: *maximize the sensitivity of the closed loop system to forces and torques*. In classical servo problems, negative feedback loops with large gains result in small sensitivity within a bandwidth, which means that they reject forces and torques (usually called disturbances). However, the above analysis states that the exoskeleton controller needs a large sensitivity to forces and torques.

The exoskeleton controller uses the inverse of the exoskeleton dynamics as a positive feedback such that the loop gain for the exoskeleton approaches unity from below (slightly less than 1), which can be written as:

$$S_{NEW} = \frac{v}{d} = \frac{S}{1 - GC} \quad (4)$$

where C is chosen as

$$C = (1 - \alpha^{-1})G^{-1} \quad (5)$$

and α is the amplification number greater than unity.

If $\alpha = 10$, then $C = 0.9G^{-1}$, and the new sensitivity transfer function is $S_{NEW} = 10S$ (ten times the force amplification). Equation (5) simply states that a positive feedback controller needs to be chosen as the inverse dynamics of the system dynamics scaled down by $(1 - \alpha^{-1})$. Note that Eq. (5) prescribes the controller in the absence of unmodeled high-frequency exoskeleton dynamics. In practice, C also includes a unity gain low pass filter to attenuate the unmodeled high-frequency exoskeleton dynamics that may not be captured in the model, G^{-1} .

The success of this control method is dependant on the accuracy of system model (i.e. G^{-1}) which governs how much torque is needed at each joint to compensate for the payload and dynamics of the exoskeleton. Models errors which cause the exoskeleton to apply too little actuation torque mean that the pilot would feel a portion of the payload. Errors which cause over-actuation however, could lead to instability. This straightforward control solution comes with an expensive price: robustness to parameter variations. To get the above method working, one needs to know the dynamics of the system very well. The next section discusses this tradeoff.

3.2 Robustness to Parameter Variations

The variation in the new positive feedback sensitivity transfer function (4) is given by Eq. (6).

$$\frac{\Delta S_{NEW}}{S_{NEW}} = \frac{\Delta S}{S} + \frac{GC}{1 - GC} \frac{\Delta G}{G} \quad (6)$$

If GC is close to unity (when the force amplification number, α , is large) any parameter variation on modeling will be amplified as well. For example if the parameter uncertainty in the system is about 10%, i.e.: $\left| \frac{\Delta G}{G} \right| = 0.10$ and $\left| \frac{\Delta S}{S} \right| = 0$, then Eq. (6) results in

$$\left| \frac{\Delta S_{NEW}}{S_{NEW}} \right| = \left| \frac{GC}{1 - GC} \right| 0.10 \quad (7)$$

Now assume C is chosen such that $C = 0.9G^{-1}$. Substituting into Eq. (7) results in

$$\left| \frac{\Delta S_{NEW}}{S_{NEW}} \right| = 0.90. \quad (8)$$

Equation (8) indicates that any parameter variation directly affects the system behavior. In the above example, a 10% error in model parameters results in nine times the variation in the sensitivity function. This is why model accuracy is crucial to exoskeleton control. One can see this problem as a tradeoff: the design approach described above requires no sensor (e.g. force or electromyogram) in the interface between the pilot and the exoskeleton; one can push and pull against the exoskeleton in any direction and at any location without measuring any variables on the interface. However, the control method requires a very good model of the system. At this time, our experiments with BLEEX have shown that this control scheme—which does not stabilize BLEEX—forces the exoskeleton to follow wide-bandwidth human maneuvers while carrying heavy loads.

3.3 Pilot Dynamics

In our control scheme, as will be shown, there is no need to include the internal components of the pilot limb model; the detailed dynamics of nerve conduction, muscle contraction, and central nervous system processing are implicitly accounted for in constructing the dynamic model of the pilot limbs. For more detail on in-depth modeling and analysis of the internal

components of the pilot limb as applied to haptic systems and human power amplifiers, see [22,23] and our preliminary results in [24].

The pilot force on the exoskeleton, d , is a function of both the pilot dynamics, H , and the kinematics of the pilot limb (e.g., velocity, position or a combination thereof).

$$d = -H(v) \quad (9)$$

The specific form of H is not known other than that it results in the human muscle force on the exoskeleton. In general, H is determined primarily by the physical properties of the human dynamics. Here we assume H is a nonlinear operator representing the pilot impedance as a function of the pilot kinematics.

Figure 5 represents the closed loop system behavior when pilot dynamics is added to the block diagram of Fig. 4. Examining Fig. 5 reveals that Eq. (4), representing the new exoskeleton sensitivity transfer function from d to v , is not affected by the feedback loop containing H .

Figure 5 shows an important characteristic for human exoskeleton control: *two distinct feedback loops in the system*. The upper feedback loop represents how forces and torques from the pilot affect the exoskeleton and is internal to the human. The lower loop shows how the controlled feedback loop affects the exoskeleton. While the lower feedback loop is positive (potentially destabilizing), the upper human feedback loop stabilizes the overall system of pilot and exoskeleton taken as a whole.

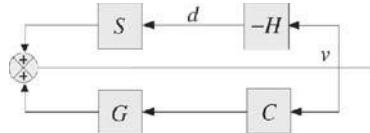


Fig. 5. The two feedback loops in this diagram represent the overall motion of the human and exoskeleton. The upper feedback loop shows how the pilot moves the exoskeleton through applied forces. The lower positive feedback loop shows how the controller drives the exoskeleton.

3.4 The Effect of Pilot Dynamics on Closed Loop Stability

How does the pilot dynamic behavior affect the exoskeleton behavior? In order to get an understanding of the system behavior in the presence of pilot dynamics we use our 1 DOF system and assume H is a linear transfer

function. The stability of the system shown in Fig. 5 is decided by the closed-loop characteristic equation:

$$1 + SH - GC = 0 \quad (10)$$

In the absence of feedback controller, C , the pilot carries the entire load (payload plus the weight of the exoskeleton torso). The stability in this case is decided by the characteristic equation:

$$1 + SH = 0 \quad (11)$$

Characteristic equation (11) is always stable since it represents pilot coupled to a passive exoskeleton (i.e. no controller means $GC = 0$). When a feedback loop with C is added, the closed loop characteristic equation changes from Eq. (11) to Eq. (10), and using the Small Gain Theorem, one can show that the closed loop stability is guaranteed as long as inequality (12) is satisfied:

$$|GC| < |1 + SH| \quad \forall \omega \in (0, \infty) \quad (12)$$

According to Eq. (5), C is chosen such that $|GC| < 1$ and therefore in the absence of uncertainties, Eq. (12) is guaranteed as long as $1 \leq |1 + SH|$. Unlike control methods utilized in the control of the upper extremity exoskeletons [6,25,26], the human dynamics in the control method described here has little potential to destabilize the system. Even though the feedback loop containing C is positive, the feedback loop containing H stabilizes the overall system of pilot and exoskeleton. The condition in Eq. (12) could be violated if $|GC| \geq 1$, which would result from model parameter uncertainties. In summary, the controller discussed here is stable when worn by the pilot as long as parameter uncertainties are kept to a minimum.

4 Controller Implementation on BLEEX

4.1 Mechanical System

The above discussion motivated the design philosophy using a one DOF system. BLEEX, as shown in Fig. 1, is a system with many degrees of freedom and therefore implementation of BLEEX control deserves further attention. Each BLEEX leg has three DOFs at the hip, one DOF at the knee, and three DOFs at the ankle, of which only four are powered DOFs:

hip, knee, and ankle joints in the sagittal plane and the hip abduction-adduction joints. See [24] for details of the BLEEX mechanical design.

The pilot and BLEEX have rigid mechanical connections at the torso and the feet; everywhere else, the pilot and BLEEX have compliant or periodic contact. The connection at the torso is made using an adjustable compliant vest that distributes the forces between BLEEX and the pilot, thereby preventing abrasion. The vest includes a rigid plate (with hole pattern) on the back that bolts to the rigid metal spine of the BLEEX torso.

The pilot's shoes or boots attach to the BLEEX feet using a modified quick-release binding mechanism similar to snowboard bindings. The binding cleat on the modified pilot boot does not interfere with normal wear when the pilot is unclipped from BLEEX. The BLEEX foot is composed of a rigid heel section with the binding and a compliant, but load bearing, toe section that begins mid foot and extends to the toe. The BLEEX foot has a compressible rubber sole with a tread pattern that provides both shock-absorption and traction while walking. The rubber sole of the BLEEX foot contains multiple embedded pressure sensors (coarse on/off information only), that are used to detect the trajectory of the BLEEX ground reaction force starting from "heel-strike" to "toe-off" in the walking gait cycle. This information is used in the BLEEX controller to identify the BLEEX foot configuration relative to the ground.

BLEEX is powered via a compact portable hybrid output power supply contained in the backpack. Several different portable BLEEX power supplies have been designed by our group for different applications and environments. Each provides hydraulic flow and pressure for the actuators and generates electric power for the sensors, network, and control computer. Details of the design, testing, and performance of the BLEEX power supplies can be found in [27].

4.2 Dynamic Modeling

Although biomechanical studies of walking frequently identify seven or more distinct phases of the human walking gait cycle [28], for simplicity in control we consider BLEEX to have three distinct phases (shown in Fig. 6) which manifest to three different dynamic models:

1. *Single support*: one leg is in the stance configuration while another leg is in swing.
2. *Double support*: both legs are in stance configuration and situated flat on the ground.

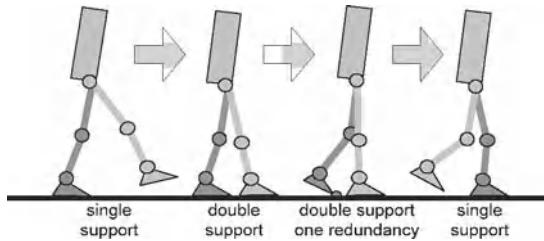


Fig. 6. Three phases of the BLEEX walking cycle.

3. *Double support with one redundancy*: both legs are in stance configuration, but one leg is situated flat on the ground while the other one is not.

Using the information from the sensors in the foot sole, the controller determines in which phase BLEEX is operating and which of the three dynamic models apply.

In our initial control design process, we decoupled the control of the abduction-adduction DOF at the hip from the control of joints in the sagittal plane. This is valid because we noticed through measurements that the abduction-adduction movements during normal walking (less than 0.9 m/s or 2 mph) are rather slow [29]. In comparison with the movements in the sagittal plane, the abduction-adduction movements can be considered quasi-static maneuvers with little dynamical affects on the rest of system. For the sake of brevity, the following sections describe the control method in the sagittal plane for a given set of abduction-adduction angles.

4.3 Single Support

In the single support phase, BLEEX is modeled as the seven DOF serial link mechanism in the sagittal plane shown in Fig. 7. The dynamics of BLEEX can be written in the general form as:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + P(\theta) = T + d \quad (13)$$

where $\theta = [\theta_1 \ \theta_2 \dots \theta_7]^T$ and $T = [0 \ T_1 \ T_2 \dots \ T_6]^T$.

M is a 7×7 inertia matrix and is a function of θ , $C(\theta, \dot{\theta})$ is a 7×7 centripetal and Coriolis matrix and is a function of θ and $\dot{\theta}$, and P is a 7×1 vector of gravitational torques and is a function of θ only. T is the 7×1 actuator torque vector with its first element set to zero since there is no actuator associated with joint angle θ_1 (i.e. angle between the BLEEX

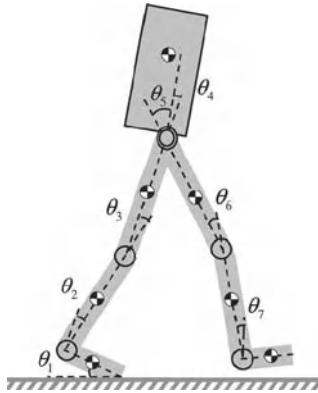


Fig. 7. Sagittal plane representation of BLEEX in the single stance phase.

foot and the ground). d is the effective 7×1 torque vector imposed by the pilot on BLEEX at various locations. According to Eq. (5), we choose the controller to be the inverse of the BLEEX dynamics scaled by $(1 - \alpha^{-1})$, where α is the amplification number.

$$T = \hat{P}(\theta) + (1 - \alpha^{-1})[\hat{M}(\theta)\ddot{\theta} + \hat{C}(\theta, \dot{\theta})\dot{\theta}] \quad (14)$$

$\hat{C}(\theta, \dot{\theta})$, $\hat{P}(\theta)$ and $\hat{M}(\theta)$ are the estimates of the Coriolis matrix, gravity vector, and the inertia matrix respectively for the system shown in Fig. 7. Note that Eq. (14) results in a 7×1 actuator torque. Since there is no actuator between the BLEEX foot and the ground, the torque prescribed by the first element of T must be provided by the pilot. Substituting T from Eq. (14) into Eq. (13) yields,

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + P(\theta) = \hat{P}(\theta) + (1 - \alpha^{-1})[\hat{M}(\theta)\ddot{\theta} + \hat{C}(\theta, \dot{\theta})\dot{\theta}] + d \quad (15)$$

In the limit when $M(\theta) = \hat{M}(\theta)$, $C(\theta, \dot{\theta}) = \hat{C}(\theta, \dot{\theta})$, $P(\theta) = \hat{P}(\theta)$, and α is sufficiently large, d will approach zero, meaning the pilot can walk as if BLEEX did not exist. However, it can be seen from (15) that the force felt by the pilot is a function of α and the accuracy of the estimates $\hat{C}(\theta, \dot{\theta})$, $\hat{P}(\theta)$, and $\hat{M}(\theta)$. In general, the more accurately the system is modeled, the less the human force, d , will be. In the presence of variations in abduction-adduction angles, only $P(\theta)$ in Eqs. (13, 14) needs to be modified.

4.4 Double Support

In the double support phase, both BLEEX feet are flat on the ground. The exoskeleton is modeled as two planar 3 DOF serial link mechanisms that are connected to each other along their uppermost link as shown in Fig. 8-a. The dynamics for these serial links are represented by Eqs. (16, 17).

$$M_L(m_{TL}, \theta_L) \ddot{\theta}_L + C_L(m_{TL}, \dot{\theta}_L, \theta_L) \dot{\theta}_L + P_L(m_{TL}, \theta_L) = T_L + d_L \quad (16)$$

$$M_R(m_{TR}, \theta_R) \ddot{\theta}_R + C_R(m_{TR}, \dot{\theta}_R, \theta_R) \dot{\theta}_R + P_R(m_{TR}, \theta_R) = T_R + d_R \quad (17)$$

where: $\theta_L = [\theta_{L1} \ \theta_{L2} \ \theta_{L3}]^T$ and $\theta_R = [\theta_{R1} \ \theta_{R2} \ \theta_{R3}]^T$. m_{TR} and m_{TL} are effective torso masses supported by each leg and m_T is the total torso mass such that:

$$m_T = m_{TR} + m_{TL} \quad (18)$$

The contributions of m_T on each leg (i.e., m_{TL} and m_{TR}) are chosen as functions of the location of the torso center of mass relative to the locations of the ankles such that:

$$\frac{m_{TR}}{m_{TL}} = \frac{x_{TL}}{x_{TR}} \quad (19)$$

where x_{TL} is the horizontal distance between the torso center of mass and the left ankle, and x_{TR} is the horizontal distance between the torso center of mass and the right ankle. For example, if the center of mass of the torso is located directly above the right leg, then $m_{TL} = 0$ and $m_{TR} = m_T$. Similar to the single stance phase, the controllers are chosen such that

$$T_L = \hat{P}_L(m_{TL}, \theta_L) + (1 - \alpha^{-1}) [\hat{M}_L(m_{TL}, \theta_L) \ddot{\theta}_L + \hat{C}_L(m_{TL}, \dot{\theta}_L, \theta_L) \dot{\theta}_L] \quad (20)$$

$$T_R = \hat{P}_R(m_{TR}, \theta_R) + (1 - \alpha^{-1}) [\hat{M}_R(m_{TR}, \theta_R) \ddot{\theta}_R + \hat{C}_R(m_{TR}, \dot{\theta}_R, \theta_R) \dot{\theta}_R] \quad (21)$$

Needless to say, Eq. (19) is valid only for quasi-static conditions where the accelerations and velocities are small. This is in fact the case, since in the double support phase, both legs are on the ground and BLEEX's angular acceleration and velocities are quite small. This allows us to simplify Eqs. (20, 21) during slow walking by removing all terms except the estimates of the gravitational vectors.

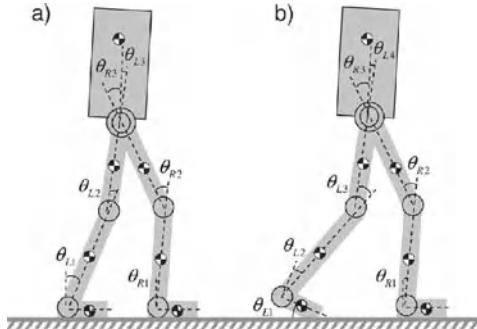


Fig. 8. Sagittal plane representation of BLEEX in a) the double support phase and b) the double support phase with one redundancy.

4.5 Double Support with One Redundancy

Double support with one redundancy is modeled as a 3 DOF serial link mechanism for the stance leg with the foot flat on the ground and a 4 DOF serial link mechanism for the stance leg that is not completely on the ground (Fig. 8-b). Each serial link supports a portion of the torso weight. The dynamics for these serial links are similar to Eqs. (16, 17), with the exception that the redundant leg equation represents four DOFs as opposed to three. For the specific instant shown in Fig. 8-b, the left leg has 4 DOF and the right leg has 3 DOF.

Similar to the double support case, the effective torso mass supported by each leg is computed by Eq. (19). Controllers for this case can be chosen in the same manner as Eqs. (20, 21). Note that the actuator torque vector associated with the leg that has 4 DOF (e.g. T_L for the case shown in Fig. 8-b) is a 4×1 vector. As in the single support phase, the torque prescribed by the first element of T must be provided by the pilot because there is no actuator between the BLEEX foot and the ground. As the pilot walks, BLEEX transitions through the various phases shown in Fig. 6. The foot sole pressure sensors detect which leg has four degrees of freedom and which leg has three degrees of freedom and the controller then chooses the appropriate algorithm for each leg.

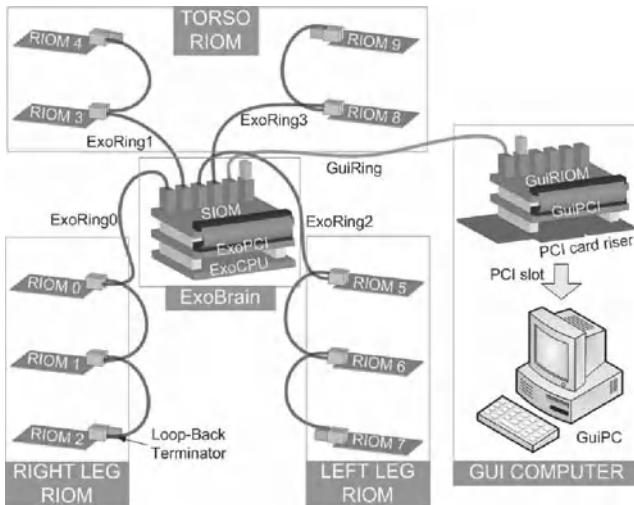


Fig. 9. Overall view of ExoNET networked control system and external GUI debug terminal.

5 Networked Control System

5.1 Network Structure

Fig. 9 provides global picture of the ExoNET networked control system that hosts the control algorithm. ExoNET was designed for BLEEX to enable the central controller to interact with distributed sensors, reduce the bulk, complexity, and difficulty of wiring, and achieve high-speed real-time control.

ExoNET consists of four ring networks (ExoRing0~3). Each ring has a series of sensor and actuator data aggregation network nodes we call Remote I/O Modules, or RIOMs. Also, an additional network (GuiRing) provides a graphical user interface (GUI) for debugging and can be hot-swapped into the system while the network is running.

These five networks are all served by a central module (ExoBrain) composed of a single board computer (ExoCPU), a PCI Interface Module (ExoPCI) and a Supervisor I/O Module (SIOM). The SIOM is a custom built board that has three independent transceiver channels. Channel 1 contains the two leg network rings: ExoRing0 and ExoRing2. Channel 2 contains the two torso network rings: ExoRing1 and ExoRing3. The third transceiver channel is coupled to GUI network (GuiRing).

The GUI network allows for real-time monitoring and administration of the control system for debugging purposes. The GUI network includes the GUI RIOM (GuiRIOM), the GUI PCI interface module (GuiPCI) and the GUI computer (GuiPC) inside the GUI computer case. The GuiRIOM and the GuiPCI stack together on a PCI riser card that is plugged into one of PCI slots of the GuiPC.

Fig. 10 shows the ExoNET ring topology of an ExoRing where N RIOMs (Slave 1,2,...N) are serially connected to one SIOM (Master). Each serial link consists of three twisted pairs of wires. While the first and second pairs are used for receiving and transmitting data, the third pair is used for carrying power to RIOMs. The direction of data flow in this network is from the transmit port (TX) of the master node to the receive port (RX) of the Slave 1 node, and from the TX of the Slave 1 node to the RX of the Slave 2 node, and so forth. This path continues to the RX of the Slave N node. A loop-back terminator completes the ring by plugging into the TX of the Slave-N node so that data leaving from the TX of the Slave-N node will arrive at the RX of the SIOM master node after passing through each slave node internal loop line (labeled LP in Fig. 10).

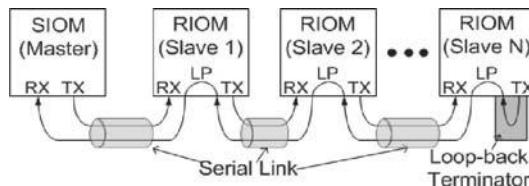


Fig. 10. Ring Topology

This ring topology provides flexibility and expandability for the network: a user can easily add or remove slave nodes and then plug a loop-back terminator at the last slave node to complete the network. This topology eliminates the requirement of a single circulating serial cable connecting from the last slave node to the master node. It is particularly useful in cabling a network where all nodes are physically oriented on a line (as on the BLEEX legs). In this case, only one serial link cable between any two consecutive nodes and a loop-back terminator on the last node are required to form a complete ring network.

5.2 Remote I/O Modules (RIOMS) – A Practical Solution to Managing Wiring Complexity

The ExoNET RIOM modules (Fig. 11) act as smart sensor and control data aggregation nodes. BLEEX is a complex multi-degree of freedom device

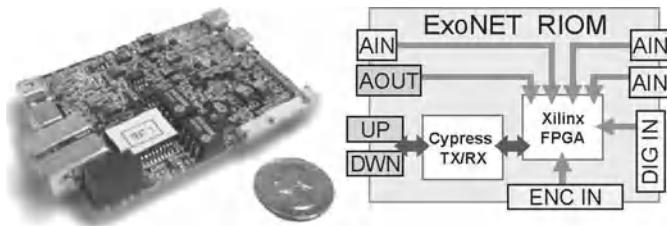


Fig. 11. ExoNET RIOM photo and schematic. Each RIOM provides for 3 analog inputs (AIN), 1 analog output (AOUT), 6 digital inputs (DIG IN), 1 quadrature encoder input (ENC IN), and 2 network communication ports (UP and DWN). Two integrated circuits handle processing (Xilinx Inc.) and network communication (Cypress Semiconductor Inc.) respectively.

with a large number of sensor inputs and control outputs. A common approach in robotic design would be to route all sensor and actuator signal wires directly to a central control computer. For BLEEX, this would have meant a total of over 210 wires. By distributing network nodes around the robot, we only require one wire per limb (ring network) to route all sensor and actuation information. Each node takes care of interfacing with the various types of sensors (serial, parallel, analog, digital, etc...) located physically close by it and assembles the data into digital packets that can be transmitted via the network to-from the central computer. In addition, the RIOM can send out analog control signals for actuation (e.g. controlling a hydraulic servo-valve). The distribution and location of RIOMs is generally chosen to achieve a minimum volume of wiring and a reasonable and convenient allocation of sensors and actuators to each RIOM.

Each RIOM in the first generation BLEEX provides five sensor inputs (three 16-bit analog inputs, one quadrature encoder input, and one 6-bit digital input) and one 16-bit analog actuation output. Details of the electronics hardware design can be found in [30]. Each link includes a pair of linear accelerometers that are connected to two of the analog input channels. When spaced out along a limb and read in differentially, these provide angular rate data for the controller. Each link is also responsible for recording data from an angular encoder at its proximal joint through the encoder channel. The remaining digital input channels are used on the foot links for capturing footswitch data that corresponds to the ground contact location. The analog output channel is used to send the control signal to the hydraulic servo-valve on an actuator.

5.3 Network Protocol

ExoNET network communication consists of passing “messages” around the ring formed by the nodes on the network (SIOM and RIOMs). A message is a series of data packets that is preceded by a chosen start delimiter code (“S” in Fig. 12) and proceeded by a chosen end delimiter code (“E” in Fig. 12). Each data packet in a message includes bits that indicate the source of the packet (e.g. the ID# of a RIOM), the type of data (e.g. error, sensor, actuation command,...), the actual data value, and error checking bits based on a cyclic redundancy check (CRC). Fig 12 shows two types of data packets in the message being passed between nodes: actuator commands sent from the master to the RIOMs (C1, C2), and sensor data destined for the master that was collected by the RIOMs (T11, T12, T21,...).

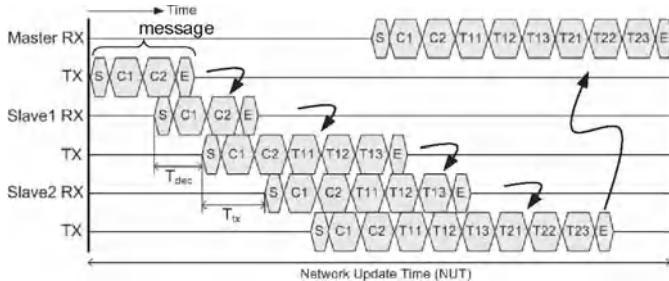


Fig. 12. Network communication overview. In a communication cycle, messages (strings of data packets beginning with start delimiter, S, and ending with end delimiter, E) pass from a master node (SIOM) to each slave node (RIOM) sequentially and return to the master node to complete the ring network communication.

Each communication cycle in the network protocol (Fig. 12) involves passing a message sequentially from the master node (e.g. a SIOM) to each slave node in the ring network (e.g. the RIOMs) and then returning the message back to the master node. As the message travels around the ring, each RIOM reads its assigned actuator command data packet (by looking for its RIOM ID#), and then appends its collected sensor data to the message. When the message returns to the master, completing the ring, it has grown to include the sensor data from each node in the network. Because the communication cycles occur at a fixed rate set by the control scheme, this protocol allows for deterministic control and provides built in network error detection because every message returning to the master must contain information from each node on the ring.

Testing ExoNET on BLEEX has shown that the network update time (NUT) for a ten RIOM network, passing 140 bytes of data, takes less than

20 μ s. The controller for BLEEX updates at 2 kHz (500 μ s sample time), which leaves 480 μ s to perform the control algorithm calculations on the ExoCPU. For a more detailed discussion of the BLEEX control network and performance analysis, see [30,31].

6 Conclusion

The Berkeley Lower Extremity Exoskeleton (BLEEX) is not a typical servo-mechanism. While providing disturbance rejection along some axes preventing motion in response to gravitational forces, BLEEX actually encourages motion along other axes in response to pilot interface forces. This characteristic requires large sensitivity to pilot forces which invalidates certain assumptions of the standard control design methodologies, and thus requires a new design approach. The controller described here uses the inverse dynamics of the exoskeleton as a positive feedback controller so that the loop gain for the exoskeleton approaches unity (slightly less than 1). Our current experiments with BLEEX have shown that this control scheme has two superior characteristics: 1) it allows for wide bandwidth maneuvers; 2) it is unaffected by changing human dynamics. The trade off is that it requires a relatively accurate model of the system. The ExoNET control network that hosts the control algorithm is also presented. Video clips which demonstrate BLEEX, the control network, and the effectiveness of the control scheme can be found at <http://bleex.me.berkeley.edu/bleex>.

References

1. H. Kazerooni, J.L. Racine, H. Huang, and R. Steger, "On the Control of the Berkeley Lower Extremity Exoskeleton (BLEEX)," IEEE Int. Conf. Robotics and Automation, Barcelona, Spain, Apr. 2005.
2. H. Kazerooni, and S. Mahoney, "Dynamics and Control of Robotic Systems Worn By Humans," ASME Journal of Dynamic Systems, Measurements, and Control, vol. 113, no. 3, Sept. 1991.
3. H. Kazerooni, "Human-Robot Interaction via the Transfer of Power and Information Signals," IEEE Transactions on Systems and Cybernetics, vol. 20, no. 2, Apr. 1990, pp. 450-463.
4. H. Kazerooni, and J. Guo, "Human Extenders," ASME Journal of Dynamic Systems, Measurements, and Control, vol. 115, no. 2B, June 1993, pp 281-289.

5. F.-L. Lian, J.R. Moyne, and D.M. Tibury, "Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet", IEEE Control Systems Magazine, vol. 21, no. 1, Feb. 2001, pp. 66-83.
6. H. Kazerooni, and M. Her, "The Dynamics and Control of a Haptic Interface Device," IEEE Transactions on Robotics and Automation, vol. 10, no. 4, Aug. 1994, pp 453-464.
7. H. Kazerooni, and T. Snyder, "A Case Study on Dynamics of Haptic Devices: Human Induced Instability in Powered Hand Controllers," AIAA J. of Guidance, Control, and Dynamics, vol. 18, no. 1, 1995.
8. B.J. Makinson, and General Electric Co., "Research and Development for Machine Augmentation of Human Strength and Endurance, Hardiman I Project," General Electric Report S-71-1056, Schenectady, NY, 1971.
9. M. Vukobratovic, D. Hristic, and Z. Stojiljkovic, "Development of Active Anthropomorphic Exoskeletons," Medical and Biological Engineering, Jan. 1974, pp. 66-80.
10. K. Yamamoto, K. Hyodo, M. Ishii, and T. Matsuo, "Development of Power Assisting Suit for Assisting Nurse Labor," JSME International Journal Series C., vol. 45, No. 3, Sep. 2002.
11. P. Neuhaus, and H. Kazerooni, "Industrial-Strength Human-Assisted Walking Robots", IEEE Robotics and Automation Magazine, vol. 8, no. 4., Dec. 2001, pp. 18-25.
12. J. Pratt, B. Krupp, C. Morse, and S. Collins, "The RoboKnee: An Exoskeleton for Enhancing Strength During Walking", IEEE Conf. on Robotics and Aut., New Orleans, Apr. 2004.
13. H. Kawamoto, and Y. Sankai, "Power Assist System HAL-3 for gait Disorder Person", ICCHP, Austria, July 2002.
14. S. Biegacki and D. VanGompel, "The application of DeviceNet in process control", ISA Trans, vol. 35, no.2, 1996, pp. 169-176.
15. L.D. Gibson, "Autonomous control with peer-to-peer I/O networks", Sensors, vol. 12, no. 9. Sept. 1995, pp. 83-90.
16. A. Ray, "Introduction to networking for integrated control systems", IEEE Contr. Syst. Mag., vol. 9, Jan. 1989, pp. 76-79.
17. G. Schickhuber and O. McCarthy, "Distributed fieldbus and control network systems", Computing Contr. Eng., vol. 8, no. 1, Feb. 1997, pp. 21-32.
18. D. Song, T. Divoux, and F. Lepage, "Design of the distributed architecture of a machine-tool using FIP fieldbus", in Proc. IEEE Int. Conf. Application Specific Systems, Architectures, Processors, Los Alamitos, 1996, pp. 250-260.
19. G. Cena, C. Demartini, and A. Valenzano, "On the performances of two popular fieldbuses," in Proc. IEEE Int. Workshop Factory Commun. Syst., Barcelona, Spain, Oct. 1997, pp. 177-186.
20. J.D. Wheelis, "Process control communications: Token Bus, CSMA/CD, or Token Ring?" ISA Trans., vol. 32, no. 2, July 1993, pp. 193-198.
21. S. Saad-Bouzefrane and F. Cottet, "A performance analysis of distributed hard real-time applications," in Proc. IEEE Int. Workshop Factory Commun.Syst., Barcelona, Spain, Oct. 1997, pp. 167-176.

22. D.R. Wilkie, "The relation between force and velocity in human muscle", *J. Physiology*, vol. K110, 1950, pp. 248-280.
23. J.M. Winters, and L. Stark, "Analysis of fundamental human movement patterns through the use on in-depth antagonistic muscle models", *IEEE Trans. on Biomed. Engr.* vol. BME32, no. 10, 1985.
24. A. Chu, H. Kazerooni, and A. Zoss, "On the Biomimetic Design of the Berkeley Lower Extremity Exoskeleton (BLEEX)", *IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain, April 2005.
25. H. Kazerooni, "The extender technology at the University of California, Berkeley," *Journal of the Society of Instrument and Control Engineers in Japan*, vol. 34, 1995, pp. 291-298.
26. H. Kazerooni, "The Human Power Amplifier Technology at the University of California, Berkeley", *Journal of Robotics and Autonomous Systems*, vol. 19, 1996, pp. 179-187.
27. K. Amundsen, J. Raade, N. Harding, and H. Kazerooni, "Hybrid Hydraulic-Electric Power Unit for Field and Service Robots," *IEEE IROS*, Edmunton, Canada, Aug. 2005.
28. J. Rose, and J.G. Gamble, *Human Walking*, Second Edition, Williams & Wilkins, Baltimore, 1994, p. 26.
29. A. Zoss and H. Kazerooni, "On the Mechanical Design of the Berkeley Lower Extremity Exoskeleton," *IEEE IROS*, Edmunton Canada, Aug. 2005.
30. S. Kim, G. Anwar, and H. Kazerooni, "High-speed Communication Network for Controls with Application on the Exoskeleton", *American Control Conference*, Boston, June 2004.
31. S. Kim, and H. Kazerooni, "High-speed Ring-Based Distributed Networked Control System for Real-Time Multivariable Applications," *2004 IMECE & RD&D Expo*, Nov. 2004.

Session Overview

Underwater Robotics

Louis L. Whitcomb¹ and Hugh Durrant-Whyte²

¹ Johns Hopkins University, llw@jhu.edu

² Australian Centre for Field Robotics, hugh@acfr.usyd.edu.au

It is an auspicious time for this first-ever ISRR special session on the topic of underwater robotics. Underwater robots are now performing high-resolution acoustic, optical, and physical oceanographic surveys in the deep ocean that previously were considered impractical or infeasible. For example: in 2001 the Argo II underwater robotic vehicle, [1], was employed to discover the first off-axis hydrothermal vent field located 15 km from the Mid-Atlantic Ridge at 30° North Latitude [5]. The dynamics of this important hydrothermal vent site have since been mapped, sampled, and probed extensively with human-occupied submersibles, tethered remotely controlled underwater robots, and untethered autonomous underwater robots [6, 4, 7].

The technical obstacles arising in underwater robotic missions differ from those in land, air, and space missions in several fundamental respects: First, the rapid attenuation of acoustic and electromagnetic radiation in seawater severely restricts the range (and field of view) of high resolution acoustic and optical sensors. In consequence, high-resolution underwater survey sensors must be submerged to the immediate vicinity of a survey site — in sharp contrast to airborne and space-based survey sensors systems. Moreover, radio navigation techniques commonly employed in land, air, and space operations do not function undersea. Second, the high ambient pressure of the underwater environment poses formidable design challenges both for (inhabited) submarines and (uninhabited) robots. At present, only a handful of the world's submarines are capable of diving beyond 1000 meters in depth. Only one present-day operational research submarines can dive to 6500 meters; none can dive to the ocean's deepest depths of 11,000 meters. In contrast, numerous underwater robots operate to 6500 meters, and at least one vehicle presently under construction will be capable of 11,000 meters operation [2, 3]. Finally, in the case of untethered vehicles, underwater missions are limited not only by on-board energy storage capacity, but also by the severely limited bandwidth and delay inherent in underwater acoustic communication, the intelligence of on board control system, and payload capacity.

The three papers in this session represent accomplishments in the engineering science problems arising in the problem domain of underwater robotics. Moreover, they are exemplars of engineering science which is motivated by and directly advance the natural sciences by enabling new methods of oceanographic research.

Plotnik and Rock report a computer vision tracking system to enable a remotely controlled underwater vehicle to track autonomously a class of gelatinous animals (e.g. free-swimming Coelenterates or “jellyfish”) in the water column. The problem addressed is that, due to the irregular natural motion of these animals in the water column and the limited field of view of underwater camera systems, it has proven difficult or impossible for an underwater vehicle to observe these creatures for any significant length of time, thus limiting scientific observations thereof. The paper reports a model based approach which seeks to classify statistically the observed motion of the animals into the distinct phases of motion which characterize their natural swimming behavior. This system, which is evaluated on field data obtained with an actual oceanographic robotic vehicle, holds promise of significantly enhancing our ability to observe these animals and, in consequence, enable advances in mid-water Pelagic Biology.

Yoerger, Jakuba, Bradley, and Bingham report the algorithms developed and refined with the Autonomous Benthic Explorer (ABE) autonomous underwater vehicle over a decade of field work performing autonomous scientific surveys in the deep sea. At the time of this paper’s writing, ABE had successfully performed a total of over 150 science dives, traveling survey paths totaling over 2,500 Km and over 1,300 hours of bottom-time at an average of over 2,000 meters depth. The paper articulates the need for precisely navigated co-registered AUV surveys in order to combine datasets obtained with a variety of disparate scientific sensors, vehicles, and deployments. The paper reports robust and accurate methods for autonomous navigation of underwater vehicles with long baseline acoustic navigation, bottom following and obstacle avoidance, and automated nested survey methodologies for locating hydrothermal vents on the mid-ocean ridges. These methods have resulted directly in numerous scientific discoveries, for example [7].

Singh, Roman, Pizarro, and Eustice report advances in high resolution acoustic and optical imaging from underwater vehicles. The authors report advances in methodologies to exploit consistency and redundancy of local sensor measurements of the environment to construct large scale high-resolution optical and acoustic maps that are a self-consistent quantitative representation of the environment. Their approach extends techniques from simultaneous localization and mapping (SLAM), photogrammetry, and computer vision to address directly the structure-from-motion problem as it arises in large scale underwater surveys with sensors possessing limited range. The authors report an overview of their research in large-scale structure from motion, self consistent bathymetric mapping, and visually aided navigation. The utility of these methods is demonstrated on several large scale deep-ocean data sets including

a survey of the shipwreck *RMS Titanic* and the Trans-Atlantic Geotraverse (TAG) Hydrothermal Vent site at 26°N 44°W on the Mid-Atlantic Ridge.

References

1. R. Bachmayer, S. Humphris, D. J. Fornari, C. L. Van Dover, J. C. Howland, A. B. Bowen, R. L. Elder, T. Crook, D. E. Gleason, W. J. Sellers, and S. Lerner. Oceanographic research using remotely operated underwater robotic vehicles: Exploration of hydrothermal vent sites on the Mid-Atlantic ridge at 37 deg North 32 deg West. *Marine Technology Society Journal*, 32(3):37 – 47, 1998.
2. A. D. Bowen, D. R. Yoerger, L. L. Whitcomb, and D. J. Fornari. Exploring the deepest depths: Preliminary design of a novel light-tethered hybrid ROV for global science in extreme environments. *Marine Technology Society Journal*, 38(2):92 – 101, 2004.
3. R. Cooke. Back to the bottom. *Nature*, 437(7059):612 – 613, 2005.
4. G. L. Fruh-Green, D. S. Kelley, S. M. Bernasconi, J. A. Karson, K. A. Ludwig, D. A. Butterfield, C. Boschi, and G. Proskurowski. 30,000 years of hydrothermal activity at the Lost City vent field. *Science*, 301(5632):495 – 498, 2003.
5. D. Kelley, J. Karson, D. Blackman, G. Fruh-Green, D. Butterfield, M. Lilley, E. Olson, M. Schrenk, K. Roe, G. Lebon, and P. Rivizzigno. An off-axis hydrothermal vent field near the Mid-Atlantic Ridge at 30 deg N. *Nature*, 412(6843):145 – 149, 2001.
6. D. S. Kelley, J. A. Baross, and J. R. Delaney. Volcanoes, fluids, and life at mid-ocean ridge spreading centers. *Annual Review of Earth and Planetary Sciences*, 30:385 – 491, 2002.
7. D. S. Kelley, J. A. Karson, G. L. Fruh-Green, D. R. Yoerger, T. M. Shank, D. A. Butterfield, J. M. Hayes, M. O. Schrenk, E. J. Olson, G. Proskurowski, M. Jakuba, A. Bradley, B. Larson, K. Ludwig, D. Glickson, K. Buckman, A. S. Bradley, W. J. Brazelton, K. Roe, M. J. Elend, A. Delacour, S. M. Bernasconi, M. D. Lilley, J. A. Baross, R. E. Summons, and S. P. Sylva. A serpentinite-hosted ecosystem: The Lost City hydrothermal field. *Science*, 307(5714):1428 – 1434, 2005.

Improved Estimation of Target Velocity Using Multiple Model Estimation and a Dynamic Bayesian Network for a Robotic Tracker of Ocean Animals

Aaron Plotnik¹ and Stephen Rock²

¹ Department of Aeronautics and Astronautics
Stanford University
aplotnik@stanford.edu

² Department of Aeronautics and Astronautics
Stanford University
and
Monterey Bay Aquarium Research Institute
Moss Landing, CA
rock@stanford.edu

Summary. A vision-based automatic tracking system for ocean animals in the midwater has been demonstrated in Monterey Bay, CA. Currently, the input to this system is a measurement of relative position of a target with respect to the tracking vehicle, from which relative velocities are estimated by differentiation. In this paper, the estimation of target velocities is extended to use knowledge of the modal nature of the motions of the tracked target and to incorporate the discrete output of an online classifier that categorizes the visually observable body motions of the animal. First, by using a multiple model estimator, a more expressive hybrid dynamical model is imposed on the target. Then, the estimator is augmented to input the discrete classification from the secondary vision algorithm by recasting the process and sensor models as a dynamic Bayesian network (DBN). By leveraging the information in the body motion classifications, the estimator is able to detect mode changes before the resulting changes in velocity are apparent and a significant improvement in velocity estimation is realized. This, in turn, generates the potential for improved closed-loop tracking performance.

1 Introduction

A vision-based automatic tracking system for ocean animals in the midwater has been developed and demonstrated under a program of joint research between the Stanford University Aerospace Robotics Lab and the Monterey Bay Aquarium Research Institute (MBARI) [1–4]. In field tests using MBARI’s ROVs

Ventana and *Tiburon* in the Monterey Bay, this system has demonstrated fully autonomous closed-loop control to track animals for periods up to 1.5 hours. This tracking system has been designed for both ROV and AUV deployments. A snapshot of the system tracking a small squid in Monterey Bay is shown in Fig. 1.

The current tracking system's control laws use velocities derived by differentiating the relative position of the target with respect to the vehicle as measured by the stereo vision sensing system. This system has been very effective in tracking many target specimens but performance can be poor when tracking very mobile and/or small targets. The performance of human pilots when doing this task, however, does not degrade nearly as much when tracking such targets. One difference between the logic currently embedded in this system and the way human pilots operate is that human pilots exploit their *a priori* knowledge of the strongly modal motion behaviors of the tracked animal and the visible body deformations associated with those motions. They do not rely solely on lead information determined through differentiation of relative position.

To improve the robustness of the tracker and expand its applicability to smaller and more mobile animals, this paper presents methods for incorporating into an estimator knowledge of the modal motions of the tracking target. Some information about the motion modes can be inferred from measured water-relative velocities. However, visual information of the type that the pilot uses is potentially available to the system, quantified as a classification of the visible body deformations exhibited by the animal [5, 6]. This non-traditional knowledge-based lead information exploits the detection of mode

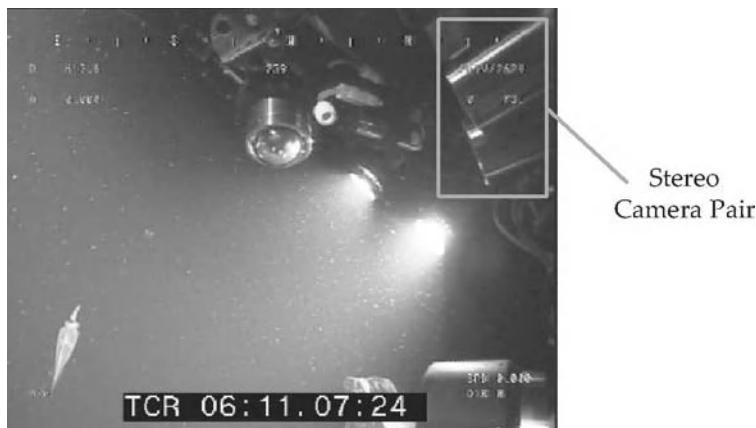


Fig. 1. A side view of robotic tracking of a squid (lower left) using the ROV *Tiburon*. The stereo camera pair used for relative position sensing is indicated at top right.

switches as an early indicator of acceleration or deceleration, and hence provides improved velocity estimation (e.g., faster convergence).

In [7], an estimator was presented that derives water-relative velocities of the tracking vehicle and target by merging measurements of water-relative velocities with the stereo vision measurements using a Sigma Point Kalman Filter (SPKF) [8]. In this paper, the estimator is augmented with knowledge of the modal nature of the tracking specimen. Improved velocity tracking is demonstrated using a multiple model bootstrap filter³ [9]. The estimator is then further augmented with discrete measurements from the online vision algorithm that classifies the type of body deformation patterns exhibited by the specimen [5]. This discrete information is fused by recasting the process and sensor models as a dynamic Bayesian network (DBN) [10] and by characterizing the discrete classification of body deformation patterns with a probability distribution conditioned on the propulsive motion mode of the specimen.

Section 2 of this paper describes the motion modes exhibited by typical midwater ocean animals. In Section 3, a set of process models for the target dynamics within a multiple model bootstrap estimator are proposed to represent the propulsive modes of a typical tracking target. Section 4 details the DBN framework that is used to incorporate the body deformation classifications from the secondary vision algorithm into the bootstrap filter. Finally, Section 5 presents the resulting performance improvements attained by utilizing multiple models and the unique lead information in these visual cues in the new DBN-based filter.

2 A Mode Model of Motion Behavior

In this section, the motion behaviors exhibited by gelatinous animals are enumerated, and a body motion mode model derived from those behaviors is established.⁴

Gelatinous animals generally effect propulsive forces by deforming some part of their bodies (or the entire body) in a pulsing motion. Pumping water in and out of the bell portion of their bodies generates a thrust force on the surrounding water. Many species exhibit periodic bell pulsing behaviors in order to propel themselves through the water. Others exhibit these pulsing actions in non-periodic patterns. Fig. 2 shows some image sequences with examples of these motions. For medusa jellyfish, an expression for dynamic

³ The method referred to by this paper as “bootstrap” filtering is a Sequential Monte Carlo method of filtering that is often called other names including particle filtering, condensation, and Monte Carlo filtering.

⁴ While the automatic tracking system is most commonly used to track gelatinous animals, other types of animal such as small squid and some (low swimming speed) fish species have been tracked. The visible cues related to their motions are also discernible by the vision algorithms of [5]. However, (1) does not apply to those classes of animal.

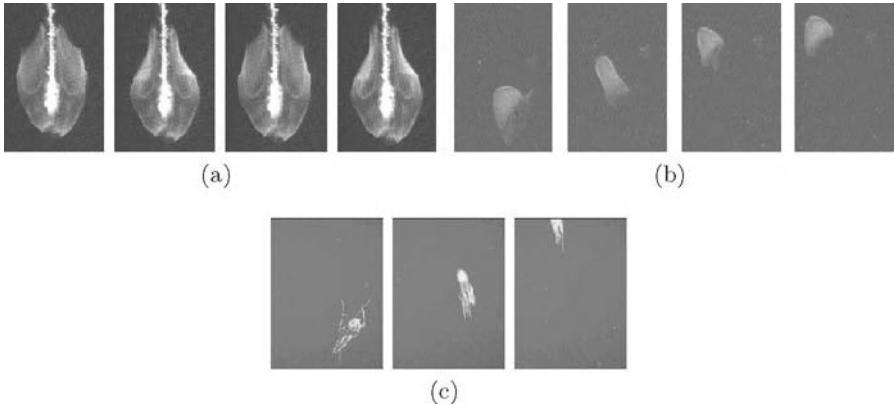


Fig. 2. Examples of body motion behaviors: (a) the head of a *Praya* siphonophore exhibiting repetitive pulsing motions with period of about 0.9 sec. time-lapse at 0.43 sec intervals and stabilized, (b) a *Benthocodon* jellyfish making a single pulse swimming motion, time-lapse at 0.27 sec intervals, (c) a *Colobonema* jellyfish makes a sudden swimming motion while being tracked by the automatic system, evading the tracking system, time-lapse at 1 sec intervals. (Images courtesy of MBARI.)

thrust given by Equation (1) shows thrust, T , related to water density, ρ , and generated by the time rate of change in the volume of the bell, V , and the velar area, A_v , as proposed by Daniel [11].

$$T = -\frac{\rho}{A_v} \frac{\delta V}{\delta t} \text{abs} \left(\frac{\delta V}{\delta t} \right) \quad (1)$$

Fig. 3(a) shows a finite automaton that expresses an observer's perception of body motion mode, including the criteria used to detect switching events. Algorithms for applying this finite automaton to a stream of video images of an object in real time were presented in [5] and [6]. The output of this vision algorithm is a classification of the motion type exhibited by the body of the observed animal at time t :

$$\sigma(t) \in \{\text{Resting}, \text{Moving}, \text{RepetitivePulsing}\} \quad (2)$$

For species whose active swimming modes are accomplished by moving or contracting significant portions of their bodies, the visible indication of those motions will lead the actual changes in velocity as the animal accelerates. In Section 5, this information is shown to assist the estimation of an animal's velocity by allowing the estimator to anticipate accelerations and decelerations before they manifest as significant changes in measured velocity.

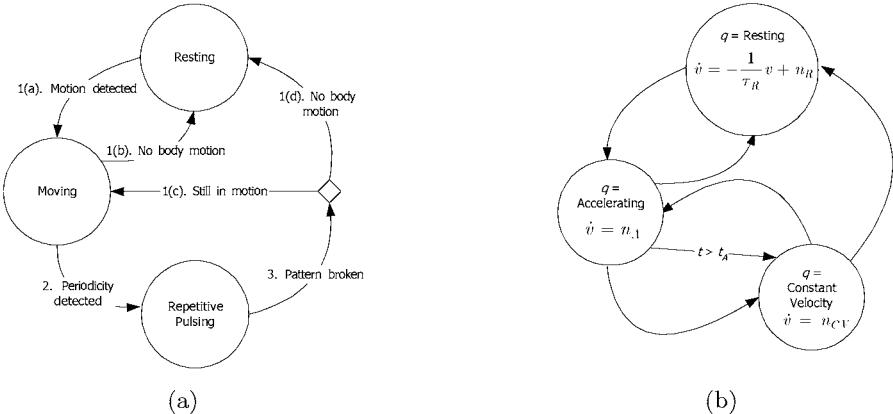


Fig. 3. (a) Gelatinous animal body motion from point of view of observer. (b) The hybrid dynamical system representation of the motion of the tracking target used by the multiple model bootstrap filter. The relationship between the body motions of (a) and the water-relative dynamic modes of (b) is assumed to be one-to-one. Thus, (1) when the body is not actively moving, no thrust is produced, (2) when the body begins moving, the animal accelerates, and (3) periodic motion of the body results in approximately constant velocity.

3 Multiple Model Estimation in the Water Frame

In [7], estimation of water-relative velocities was accomplished by merging relative bearing measurements from the stereo camera pair with water-relative vehicle velocities measured by a Doppler Velocity Log (DVL). By applying the additional knowledge of the modal nature of the active motions by the target, velocity estimation can be improved.

The use of multiple models in target tracking estimation is a popular method to estimate target motion without direct knowledge of the accelerating inputs on the target, for instance [12]. Approaches such as the Interacting Multiple Model (IMM) estimator [12] or the multiple model bootstrap filter [9] have the effect of adapting the bandwidth of the estimator based on the most probable models of the model set. The model set is typically chosen to have appropriate bandwidth properties for different tracking situations and/or to incorporate specific knowledge of the modes present in a hybrid dynamical system [13]. For a multiple model estimator tracking an unknown maneuvering target, during non-maneuvering periods, a low bandwidth estimator (one with low process noise assumed) is preferable to mitigate the effects of noisy sensors. However, during maneuvers by a target, a higher bandwidth estimator is preferable to allow the state estimate to adapt quickly to the changing conditions. Multiple model estimators for target tracking are designed to adapt

by favoring the most likely models based on the evidence provided by the measurements.

To apply these techniques to the underwater tracking system, a three-mode hybrid dynamical model is used, as shown in Fig. 3(b). Because detailed dynamic parameters are not known for a given specimen, only very generic kinematic models are used to model the motion. The $q = Resting$ dynamics include a small white noise acceleration term and a damping term representing the tendency for water-relative velocity to stay low in this mode. $q = Accelerating$ is represented as simply driven by large variance white noise acceleration. The $q = ConstantVelocity$ mode is associated with constant velocity dynamics with a different, more moderate white noise acceleration term. Switching between modes is assumed to be a Markov process for all unmarked transitions. To help capture the case of repetitive pulsing motions that accelerate from rest to a steady-state velocity, a timed transition is added from the *Accelerating* and *ConstantVelocity* modes. This timed transition also encodes the delay by the motion classification algorithm when classifying a motion as periodic, which typically requires the observation of 1 or 2 periods of motion. The full process and sensor models for the vehicle and the relative bearing (vision) and water relative velocity (DVL) sensors are presented in [7], and are not dependent on the discrete mode q of the target.

4 Incorporation of Visual Classifications of Body Motion via a Dynamic Bayesian Network Model

In this section, the modeling of a stochastic dynamic system as a dynamic Bayesian network is briefly defined and related to the familiar Kalman filter. Then the model of the tracking system's target is recast as a DBN, including an additional observation model to represent the motion classifier.

4.1 Bayesian and Dynamic Bayesian Networks

A Bayesian network (BN) is a graphical representation of a model of the probabilistic relationships and conditional independence of a set of variables [14]. The network in Fig. 4 is an example. An arrow connecting two nodes in the network (an edge) indicates that a conditional dependence exists for the child variable (the node pointed to) upon the parent variable. Based on the conditional dependencies defined by the network and the distributions associated with the variables and their parents, full joint distributions may be computed for any particular assignments to the variables in the network. That is, if each of N variables in the network are X_1, \dots, X_N , then the joint probability that they are assigned to x_1, \dots, x_N , respectively, is given by

$$p(x_1, \dots, x_N) = \prod_{i=1}^N p(x_i | parents(X_i)) \quad (3)$$

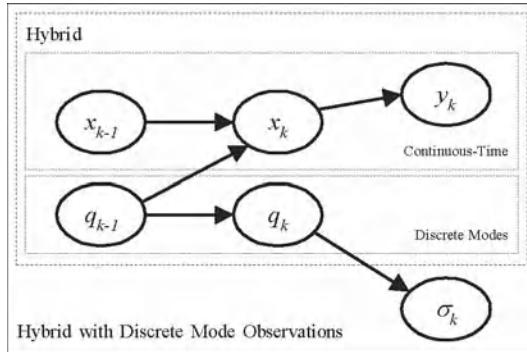


Fig. 4. A dynamic Bayesian network (DBN) representation of the process and sensor models of the multiple model estimator, including the output σ_k from the online body motion classifier, Equation (2).

A dynamic Bayesian network (DBN) is a Bayesian network describing the relationships between dynamic state variables that evolve over time. In this case, the network illustrates the dependencies between these variables at given instances in time. The Kalman filter (KF) [15] for linear Gaussian state-space systems is one example. The continuous-time portion of the DBN of Fig. 4 could represent the relationships between the states x_{k-1} and x_k and the observations y_k as modeled by a discrete-time KF (with no control inputs). Then the conditional probabilities represented by the edges in that DBN are the conditional Gaussian distributions of x_k given x_{k-1} defined by the state transition matrix and the process noise covariance, and of y_k given x_k based on the observation matrix and the observation noise covariance.

4.2 Estimation of the Target's Velocity with a DBN-Based Bootstrap Filter

In [16], bootstrap filtering was applied to a set of hybrid dynamical systems by representing the systems being monitored as dynamic Bayesian networks. The conditional probability densities (CPDs) associated with the DBNs were of varying forms including Gaussian, simple conditional probability tables (CPTs) and softmax densities [17]. A similar approach to the problem of estimating the velocities of a tracked ocean animal is employed here, combining dissimilar probabilistic models for the evolution and observation of both continuous states (the velocities) and discrete variables (the propulsive mode of the animal, q , and the discrete observation variable, σ).

Fig. 4 illustrates the state evolution and observation models of the target, cast in the form of a DBN. Note the inclusion of the discrete motion classifier, whose output, σ_k , is assumed to be dependent only upon q_k . To use this DBN in a bootstrap filter, some CPD (that can be numerically sampled) relating the

observation σ_k to the value of q_k is necessary. Based on the error rates expected when running the visual classifier [5], a simple CPT can be constructed and utilized to approximate this relationship.

The bootstrap filter, with the discrete classifier observation model added is evaluated as in [9] with two modifications based on the models used here. The first change is to the mode transition model that updates each sample for mode changes from step $k - 1$ to k . In [9], the finite automaton is assumed to be entirely Markovian. However, the mode model as specified by Fig. 3(b), requires time-in-mode for a sample to be tracked. This is accomplished by augmenting the continuous state vector with this variable and integrating it throughout the sample's life, resetting it to zero upon any mode transition. For each sample in the filter, during the time update step, this time is checked before applying the Markovian mode update for any superseding switching criteria.

The second change from [9] is more significant, the modification of the measurement update step of the filter. In systems with only continuous variables being observed, the measurement update consists of calculating the probability of y_k , the continuous-time observation, given the continuous state of each sample, $x_k^{(i)}$. Here, the measurement update requires the computation of a joint probability of (y_k, σ_k) given (x_k, q_k) of the sample. Because the values of the (x_k, q_k) pair (and their parents x_{k-1} and q_{k-1}) have been instantiated to specific values, the probabilities of y_k and σ_k given x_k and q_k become independent. Therefore the joint probability is given by the product of the two separately conditioned probabilities:

$$p(y_k, \sigma_k | x_k, q_k) = p(y_k | x_k)p(\sigma_k | q_k) \quad (4)$$

5 Results

5.1 Test Data Description

To test the performance of the multiple model and DBN-based estimators, test data was generated using the baseline tracking control software to track a simulated moving target. This target was tracked through a sequence of propulsive modes, q , in the following order: resting, acceleration, steady swimming, resting. Noisy, distorted pixel measurements for each camera of the stereo pair as well as compass and angular rate readings were generated at the 10 Hz update rate of the tracking system. DVL velocities with additive zero-mean Gaussian noise at a standard deviation of 3 cm/s were generated at 5 Hz (the maximum update rate of the DVL deployed at MBARI), with angular rates of the vehicle coupled into the measurements based on the location of the DVL on the ROV *Ventana*.

The vector of values for σ , the noisy and imperfect mode classification, is generated based upon the expected error rates of the body motion classifier

vision algorithm as applied to the “true” mode, q , of the target. This CPT, S , is given by the following:

q	$p(\sigma = Resting)$	$p(\sigma = Moving)$	$p(\sigma = RepPulse)$
<i>Rest</i>	0.90	0.08	0.02
<i>Accel</i>	0.15	0.80	0.05
<i>ConstVel</i>	0.03	0.09	0.88

5.2 Estimator Detailed Design

This data set was used to compare the performance of three estimators: (i) the UKF estimator of [7], (ii) a multiple model bootstrap filter as described in this paper but operating only on continuous state observations, and (iii) the multiple model bootstrap filter from (ii) supplemented by the classifier data, σ . For all models, the sensor noise standard deviations assumed were 2 pixels for each camera measurement (in a 160x120 image) and 3 cm/s on the DVL water velocities. Vehicle disturbance process noise standard deviation was set to 0.5 volts on all axes (on a scale with limits at +/- 5 volts for thruster command levels).

Several parameters specify the target models of Fig. 3(b) for the multiple model estimators. Target process noise terms were specified with standard deviations of 2 cm/s² (*Resting*, *ConstantVelocity*) and 10 cm/s² (*Accelerating*). The exponential decay (damping) term of the *Resting* mode dynamics was specified with a time constant, τ_R , of 2 sec. The time-in-mode limit for the *Accelerating* mode, t_A was set to 5 seconds.

For the UKF, which uses a single model design, the target model is a constant velocity model, with a single value for white noise acceleration standard deviation of 4 cm/s². This choice lies between the settings within the multiple model estimators for quiescent modes and the maneuvering mode, and is the result of the compromise required such that a single mode estimator will track adequately through more than one type of motion behavior.

The Markov mode switching probabilities are given by H below, where h_{ij} represents the probability of switching to mode i from mode j . The modes are indexed from 1 to 3 in the order presented in Equation 2.

$$H = \begin{bmatrix} 0.85 & 0.33 & 0.10 \\ 0.15 & 0.34 & 0.05 \\ 0.0 & 0.33 & 0.85 \end{bmatrix}$$

The direct use of the CPT probabilities from S in Equation 4 were found to make the estimator too sensitive to errors in the classifier output, σ , forcing the estimator to be too trusting in σ over evidence in the continuous measurements and the priors. To blunt this effect and achieve a better balance between the discrete classifier outputs and the prior belief states, a more uncertain version of S was used in the DBN-based estimator, given as S_{DBN} :

q	$p(\sigma = Rest)$	$p(\sigma = Moving)$	$p(\sigma = RepPulse)$
<i>Rest</i>	0.75	0.18	0.07
<i>Accel</i>	0.20	0.67	0.13
<i>ConstVel</i>	0.10	0.18	0.72

5.3 Results Summary

A marked improvement in overall tracking quality is demonstrated by the augmented multiple model estimator (iii) over the other two estimators. The test trajectory was partitioned into phases by the motion of the target, and the estimator errors for each algorithm are tabulated in Table 1. The modal trajectory of the target consists of a sequence of $\{Resting, Accelerating, ConstantVelocity, Resting\}$, and for the purposes of judging the performance of the estimators, the transition from *ConstantVelocity* to *Resting* is broken down into two phases (deceleration and resting, where velocity is nearly zero).

The estimator that utilizes the output of the body motion mode classifier (iii) outperforms both of the other estimators in all motion phases. The performance improvement is particularly notable in the deceleration portion of the trajectory, where this estimator is able to anticipate the deceleration based on visual cue of the ceasing of body motions by the target (as recognized by the body motion classifier).

The mode probabilities calculated for each multiple model estimator are shown in Fig. 5, with (a) showing results from the estimator of (ii) which uses continuous measurements only, and (b) showing the results from estimator (iii). These results demonstrate that the uncertainties of the measurements and of the vehicle and target dynamics are too high to discern modal information successfully without the extra information from the online classifier. The noisiness in the mode probabilities in (b) are primarily in response to errors in σ . This response is momentary, countered by the evidence in the priors and continuous measurements, keeping overall state tracking errors due to classifier errors small. Velocity tracking results for the target's velocity in the vertical direction and the 2-norm of the overall velocity error are presented in Fig. 6 (a) and (b), respectively.

Table 1. Average of 2-norm of error in target velocity estimates (cm/s), by target motion phase.

Estimator	<i>Rest1</i>	<i>Accel</i>	<i>ConstVel</i>	<i>Decel</i>	<i>Rest2</i>	Overall
(i) UKF	1.1	2.4	1.4	3.1	1.7	1.8
(ii) MM	0.7	4.1	4.5	1.2	1.3	2.5
(iii) MM-DBN	0.4	2.3	1.2	1.1	0.9	1.0

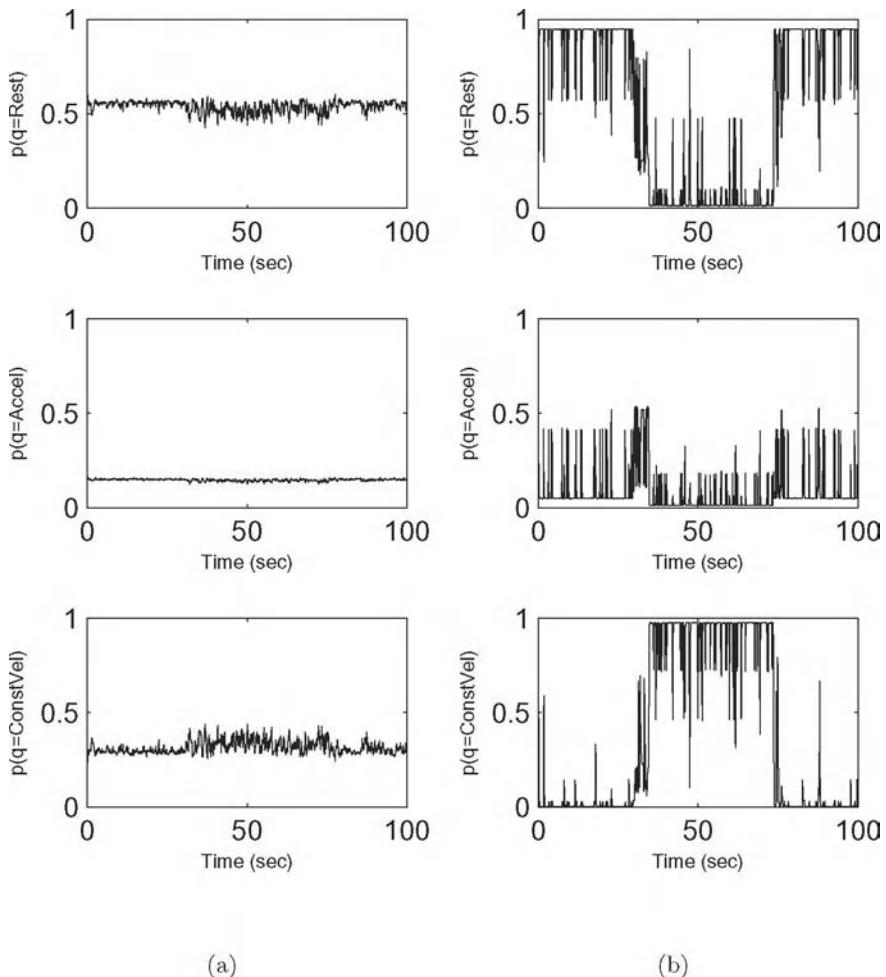


Fig. 5. Mode probability estimates calculated by (a) multiple model bootstrap, and (b) bootstrap utilizing DBN model and discrete mode observations.

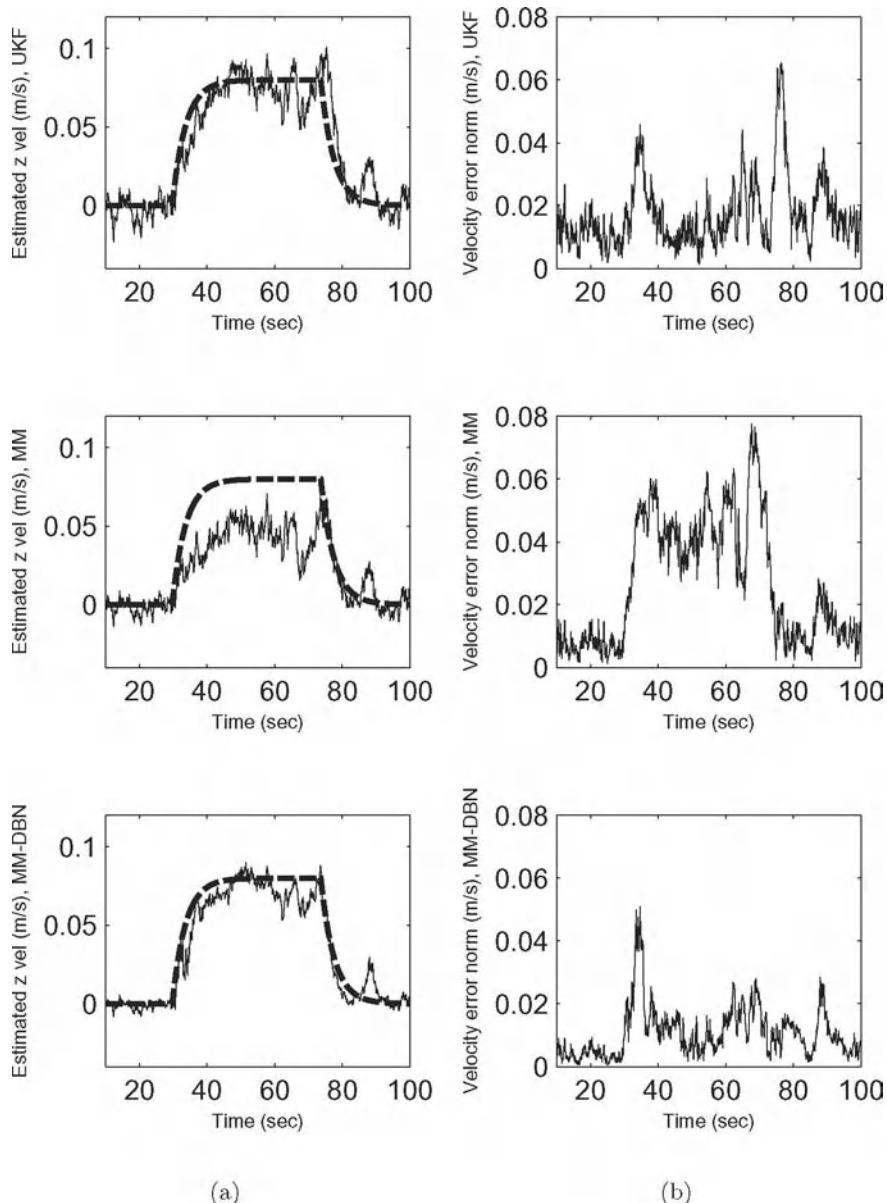


Fig. 6. Velocity tracking results comparing performance with UKF, multiple model bootstrap (denoted MM), and bootstrap utilizing DBN model and discrete mode observations (denoted MM-DBN). (a) Tracking of velocity in vertical direction (m/s) [true velocity is in bold, dashed], (b) 2-norm of tracking error for target velocity (m/s).

6 Conclusion

It has been shown that a significant advantage can be imparted to a multiple model estimator of target velocity by incorporating the extra information from a vision-based body motion mode classifier. This information allows the estimator to discern the modes of the tracked specimen in spite of large uncertainties present in the measurements, the dynamics of the vehicle and target, and the disturbances encountered by the tracking vehicle in the underwater environment. This approach allows the on-line estimator to interpret the scene in a way that is modeled after the manner in which human pilots do. With these improved estimates of the tracking target's velocities, the performance of the tracking control system can be expected to improve, especially when tracking actively maneuvering targets.

References

1. J. Rife and S. M. Rock, "Field experiments in the control of a jellyfish tracking ROV," in *Proceedings of the IEEE OCEANS Conference*, pp. 2031–2038, 2002.
2. J. Rife, *Automated Robotic Tracking of Gelatinous Animals in the Deep Ocean*. PhD thesis, Stanford University, Stanford, California, December 2003.
3. J. Rife and S. M. Rock, "Design and validation of a robotic control law for observation of deep-ocean jellyfish," *IEEE Transactions on Robotics*, submitted.
4. A. M. Plotnik and S. M. Rock, "Relative position sensing and automatic control for observation in the midwater by an underwater vehicle," in *Proceedings of the Unmanned Untethered Submersible Technology Conference (UUST)*, (Durham, NH), AUSI, Aug 2005.
5. A. M. Plotnik and S. M. Rock, "Improving performance of a jelly-tracking underwater vehicle using recognition of animal motion modes," in *Proceedings of the Unmanned Untethered Submersible Technology Conference (UUST)*, (Durham, NH), AUSI, Aug 2003.
6. A. M. Plotnik and S. M. Rock, "Quantification of cyclic motion of marine animals from computer vision," in *Proceedings of the IEEE OCEANS Conference*, pp. 1575–1581, 2002.
7. A. M. Plotnik and S. M. Rock, "A multi-sensor approach to automatic tracking of midwater targets by an ROV," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, (San Francisco, CA), Aug 2005.
8. R. van der Merwe, E. Wan, and S. Julier, "Sigma-point Kalman filters for non-linear estimation and sensor-fusion: Applications to integrated navigation," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, (Providence, RI), Aug 2004.
9. S. McGinnity and G. W. Irwin, *Sequential Monte Carlo Methods in Practice*, ch. 23. Springer, 2001.
10. T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," *Computational Intelligence*, vol. 5, no. 3, pp. 142–150, 1989.
11. T. Daniel, "Mechanics and energetics of medusan jet propulsion," *Canadian Journal of Zoology*, vol. 61, pp. 1406–1420, 1983.

12. Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley, 2001.
13. X. Koutsoukos, J. Kurien, and F. Zhao, “Estimation of distributed hybrid systems using particle filtering methods,” in *Hybrid Systems: Computation and Control*, 2003.
14. J. Pearl and T. Verma, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
15. R. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
16. D. Koller and U. Lerner, *Sequential Monte Carlo Methods in Practice*, ch. 21. Springer, 2001.
17. D. Koller, U. Lerner, and D. Angelov, “A general algorithm for approximate inference and its application to hybrid bayes nets,” in *Proceedings of the Fifteenth Annual Conference on Uncertainty in AI (UAI)*, pp. 324–333, 1999.

Techniques for Deep Sea Near Bottom Survey Using an Autonomous Underwater Vehicle

Dana R. Yoerger¹, Michael Jakuba¹, Albert M. Bradley¹,
and Brian Bingham²

¹ Woods Hole Oceanographic Institution dyoerger@whoi.edu,
mjakuba@alum.mit.edu, abradley@whoi.edu

² Franklin W. Olin College of Engineering bbing@olin.edu

Summary. This paper reports the development and at-sea deployment of a set of algorithms that have enabled our autonomous underwater vehicle, ABE, to conduct near-bottom surveys in the deep sea. Algorithms for long baseline acoustic positioning, terrain-following, and automated nested surveys are reported.

1 Introduction

This paper reports navigation algorithms that enable an underwater vehicle to accomplish fully autonomous scientific surveys in the deep sea. These algorithms allow the vehicle to determine its position, to bottom-follow (maintain a constant height above seafloor terrain) and avoid obstacles, and to autonomously focus on the highest value parts of a survey.

Scientific exploration of the deep sea has traditionally been performed using inhabited submersibles, towed vehicles, and tethered remotely operated vehicles (ROVs). Autonomous underwater vehicles (AUVs) have begun to replace these vehicles for mapping and survey missions. Autonomous vehicles complement the capabilities of these existing systems, offering superior mapping capabilities, improved logistics, and improved utilization of the surface support vessel. AUVs are particularly well suited to systematic preplanned surveys using sonars, in situ chemical sensors, and cameras in the rugged deep sea terrain that is the focus of many scientific expeditions. Inhabited submersibles and ROVs remain the only option for manipulation tasks such as sampling, deploying and recovering experiments on the seafloor, detailed inspection, and servicing subsea instruments; however, high resolution maps from AUVs can facilitate these tasks.

Figure 1 shows the Autonomous Benthic Explorer (ABE), a 6000 m autonomous underwater vehicle that our team has been developing and deploying for fine-scale quantitative survey and mapping of the seafloor. ABE can survey at constant depth or bottom-follow even in rugged terrain, and it can



Fig. 1. Operations with the Autonomous Benthic Explorer (ABE): (left) ABE being recovered; (right) world map depicting ABE's 155 science dives. To date, ABE has surveyed a distance of over 2500 km in over 1300 h of bottom-time. The average depth of these dives was over 2000 m

autonomously determine its position and drive tracklines with a precision on the order of several meters. ABE carries a variety of sensors, including scanning and multibeam sonars; a magnetometer; a digital still camera; two sets of pumped conductivity and temperature probes; an acoustic Doppler current profiler (ADCP); several chemical sensors for hydrothermal plume mapping; and occasional mission-specific instrumentation. ABE's shape and thruster placement allow it to maintain control over a wide range of speed, and to stop or back up if necessary to avoid obstacles.

ABE descends to the seafloor with the aid of a descent weight. ABE glides in a controlled spiral trajectory to ensure that it reaches the desired starting point without consuming significant battery energy. After reaching the seafloor and performing a series of checks, ABE releases its descent weight to become neutrally buoyant and begins its survey. Throughout the dive, including descent, ABE uses acoustic long-baseline (LBL) transponder navigation and, when in range of the bottom (< 300 m), bottom-lock acoustic Doppler measurements to determine its position and velocity.

A dive can consist of a mix of hydrothermal plume survey at constant depth, sonar and magnetics survey following the seafloor (at heights of 50–200 m), and digital photography (height of 5 m). ABE usually surveys until its batteries are depleted (between 15 and 30 hours depending on sensor payload and terrain). At the end of its dive, ABE releases its ascent weight to become positively buoyant and returns to the surface.

The remainder of this report is organized as follows: Sect. 2 summarizes scientific survey tasks that have motivated our AUV work, Sect. 3 reports an algorithm for acoustic positioning, Sect. 4 reports methods for terrain-following and obstacle avoidance, Sect. 5 reports a technique for automated nested survey, and Sect. 6 presents a brief summary and conclusion.

2 Precisely Navigated, Coregistered AUV Surveys

Proximity to the seafloor, precise navigation, robust control, and coregistered sensors permit an AUV to characterize the seafloor and the near-bottom environment with complementary sensing modalities on the meter-scale. This section summarizes scientific work in which ABE-derived bathymetric maps, magnetics maps, digital photos, and hydrographic maps have played critical enabling roles.

Meter-scale bathymetric and magnetic maps made using ABE have provided geologists and geophysicists with new perspectives on important seafloor processes. Combined magnetics and bathymetric maps show crustal magnetization, which permits the age and thickness of lava flows to be determined. Combined maps have also been used to identify volcanic features such as lava flow units [1], delimit their fronts, and estimate their thicknesses [2, 3]. Meter-scale bathymetric maps show tectonic features such as faults with great clarity, even enabling them to be resolved into multiple components [4]. In other cases, these maps have revealed the relationship between tectonic features and morphology, such as volcanic domes [3], and hydrothermal vents [1]. ABE bathymetric maps have proved to be of sufficient detail and precision for one collaborator to reconstruct the tectonic history of a rift valley by computationally removing faults [5]. The result revealed a dome-like structure from which the valley evolved. On a recent cruise to the Atlantis Massif, detailed renderings of faults and the hydrothermal structures provided critical clues as to the mechanisms controlling the hydro-geology at the newly discovered Lost City hydrothermal vent site [6]. Digital photographs of the seafloor from ABE have provided details of lava flow types and effusion rates [3], sediment cover, and the distribution of benthic organisms.

Water column data from ABE yields indications of hydrothermal plume activity and has been used to estimate heat flux from known hydrothermal vent sites, and to locate undiscovered sites on the seafloor. To estimate the heat flux from vent fields on the Juan de Fuca Ridge in the Northeast Pacific ($47^{\circ}54' \text{ N}$, $129^{\circ}10' \text{ W}$) [7], ABE measured temperature, salinity, and three-axis water velocity while repeatedly executing a tight grid pattern above the field [8]. Recently ABE located and preliminarily characterized several previously unmapped hydrothermal sites on the Eastern Lau Spreading Center (ELSC) south of Tonga ($21^{\circ}08' \text{ S}$, $175^{\circ}12' \text{ W}$) [9]; and on the Southern Mid Atlantic Ridge (SMAR) north of Ascension Island ($7^{\circ}57' \text{ S}$, $14^{\circ}22' \text{ W}$) [10]. In each case, we started with clues provided by towed systems that indicated a vent site within several kilometers. ABE then executed a three-dive sequence [9, 10] of grid patterns at increasing finer scales and increasingly close to the seafloor. To plan each dive, the scientific party carefully scrutinized the data from the previous dive along with any available ancillary data.

These vent prospecting missions capitalized on ABE's ability to conduct precisely navigated surveys at scales $O(\text{m-km})$, to operate over rugged terrain, and relied on nearly all of ABE's sensing modalities. Figure 2 shows

tracklines from sequence of dives designed to locate and survey a vent site on ELSC along with a sampling of the variety of data products acquired and used to plan each stage of the dive sequence. ABE mapped plume activity (temperature, optical backscatter, and reduction-oxidation potential (eH) [11]) to pinpoint the locations of plumes emanating from the field, built fine-scale bathymetric maps of the vent fields and surrounding environment, and finally photographed the vent structures and animal populations.

The remainder of this paper presents the underlying algorithms that enabled ABE to perform this work.

3 Real-Time Long Baseline Acoustic Navigation

Acoustic navigation from a set of seafloor transponders [13, 14] provides a critical capability for a seafloor mapping AUV. By georeferencing transponder location using GPS, the vehicle's position on the globe can be determined to within a few meters.

Long-baseline (LBL) positioning refers to the determination of position via interrogation of two or more fixed transponders separated by large distances (long baselines). The transponders all listen for a particular interrogation code and then reply immediately with their own unique codes. An AUV can determine its own position within a subsea LBL net as follows:

1. Before the vehicle is launched, an array of two or more acoustic transponders are moored near the seafloor. Their position and depth are determined by concurrently measuring acoustic travel times between each transponder and a number of different GPS-referenced vessel positions.
2. After launch, the vehicle repeatedly interrogates the transponders on a regular cycle and measures the time between the outgoing interrogation code and incoming replies from each transponder.
3. From these travel times, slant ranges to each transponder are computed using knowledge of the local sound speed profile, and are then projected onto the horizontal plane with knowledge of vehicle depth.
4. The horizontal position of the vehicle is then determined either through deterministic trilateration (two ranges received; relative position of baseline known), or through a nonlinear least-squares calculation (three or more ranges received).

Figure 3 shows acoustic travel time and range data from an ABE dive that illustrates many of the difficulties in autonomously processing LBL data. The upper panel of the figure shows many returns that do not correspond to the direct, round-trip path between transponder and vehicle. Some of these returns are distributed somewhat randomly, while others are systematic. The systematic incorrect returns correspond to paths that include either one or two reflections off the surface, so called bounce paths. These could be used for positioning if properly identified [15]. Returns from the previous cycle that

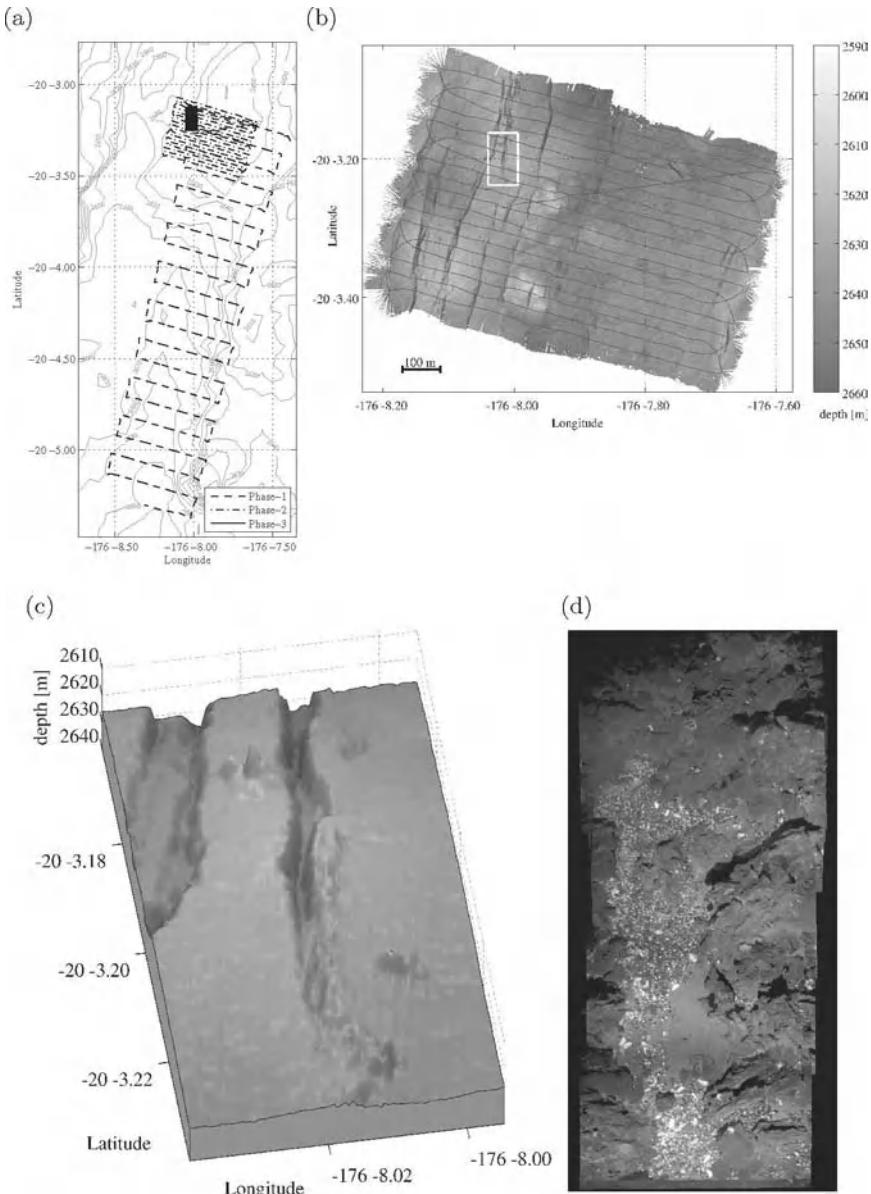


Fig. 2. Data products from a series of nested surveys whose purpose was to locate the source vent field of a hydrothermal plume near the ELSC: (a) vehicle tracklines from the four nested surveys; (b) multibeam bathymetry overlain with vehicle tracklines; (c) closeup of bathymetry showing spires of hydrothermal origin (no vertical exaggeration); (d) photomosaic created automatically [12] from six individual photographs. We relied primarily on water column hydrothermal tracer data (not shown) to drive the design of each survey stage.

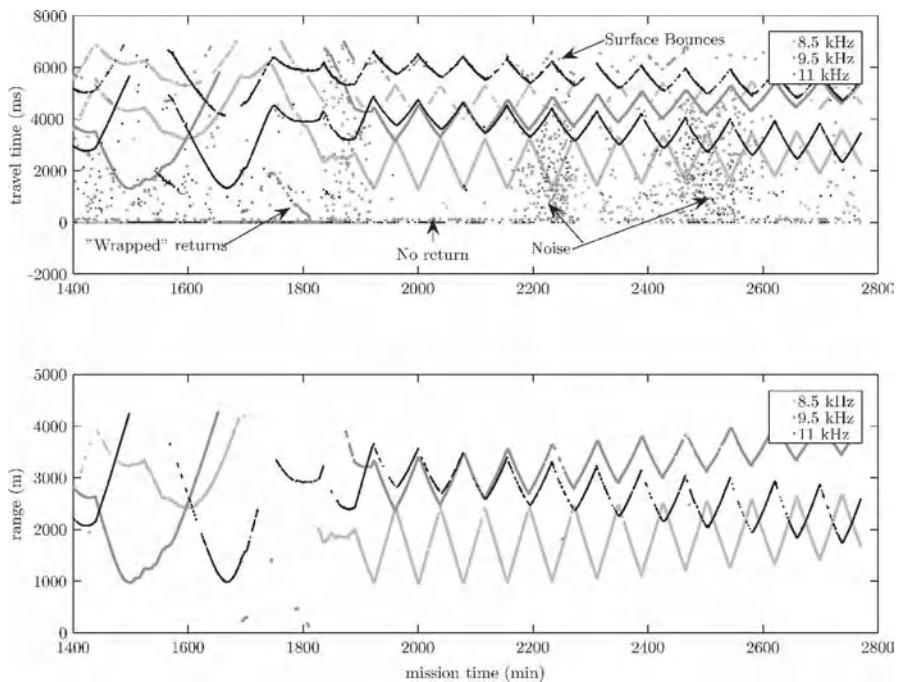


Fig. 3. Acoustic travel-time processing for long-baseline navigation: (**top**) raw acoustic travel times from three transponders, indicated by color, received by the vehicle during a survey operation; (**bottom**) filtered ranges computed by the vehicle in real-time. The labels indicate many of the pathologies of long baseline data: surface bounces, wrapped returns, missed returns, and nonstationary noise. The filtered results show that the median test and range gates in ABE's LBL navigation algorithm (Fig. 4) have largely rejected systematic, but incorrect returns from bounce paths and from previous cycles (wrap-around), and have also rejected most random, unsystematic returns

exceed the cycle period show up as short, consistent ranges (wrap-around). Reflections from terrain can provide consistent, incorrect returns as well. The distribution of unsystematic, random returns can change dramatically over the course of a dive, and we have even seen periods of uncorrelated returns due to active interrogation of the transponders by marine mammals. In summary, LBL range data is systematically corrupted by noise processes with non-Gaussian, nonstationary error distributions.

Because of these inherent difficulties, our LBL algorithm for autonomous AUV navigation (Fig. 4) emphasizes reliability over accuracy. The algorithm is self-starting and recovers gracefully from long periods of bad data. For instance, if the algorithm calculates a bad position due to unanticipated cir-

cumstances (e.g. a consistent reflection off a steep cliff), the computed position returns quickly to the correct position when the anomalous condition passes.

Much of the algorithm's reliability derives from the use of a median test for range consistency and from range gates that eliminate consistent bounce paths. These tests prevent most incorrect ranges from entering into fix computation. Both the capabilities and limitations of this portion of the algorithm are evident in the lower panel of Fig. 3. The indirect and wrap-around returns have been rejected as have nearly all the random returns. Even dense sections of noisy returns result in only a few bad ranges being accepted, and these single returns were rejected by the residual test when the fix was computed. The weakness of the median test is also evident: when the percentage of good ranges drops to near 50% or lower, the median test drops most ranges. Despite this weakness, we employ the median test for its predictability and ability to recover quickly from extended periods of bad data.

When the vessel is close enough to the vehicle, we observe the vehicle's position from the support vessel by monitoring the arrival time of the interrogation pulse from the vehicle and the corresponding replies from the transponders. This process is simplified if the time at which the interrogation is initiated at the vehicle is known. ABE's LBL cycle is controlled by a precise clock (accurate to 2 parts in 10^7 , i.e. a drift of about 1 ms/h). From the vessel, we periodically interrogate a separate transponder on the vehicle to measure the slant range to the vehicle. Comparing this range to that implied by the interrogation pulse allows the start time of the acoustic cycle to be determined and any clock drift to be tracked.

4 Bottom-Following and Obstacle Avoidance

To execute mapping missions successfully, ABE must drive commanded track-lines and avoid unexpected obstacles. On seafloor imaging dives, ABE must also follow the seafloor at a prescribed height to ensure proper sonar (height: 50–200 m) and camera (height: 5 m) performance. Bottom-following and obstacle avoidance become especially difficult during photographic surveys, as the vehicle must cope with steep-sided features such as hydrothermal spires and scarps that are many times the nominal survey height (Fig. 1(c)). Spires and volcanic collapse pits frequently have overhanging structures that present a substantial threat, as the vehicle has no up-looking sonars and could become trapped.

Our a priori knowledge of the seafloor bathymetry is rarely sufficient to permit preplanning, therefore ABE's terrain-following algorithm (Fig. 6, left panel) relies strictly on real-time height data from three different sonar devices (two down-looking, one forward looking). The algorithm uses the height data to command the set-point for the depth controller, which remains active since the depth sensor is more reliable than the bottom-finding sonars. The depth setpoint is varied to keep the height off bottom within a prescribed depth

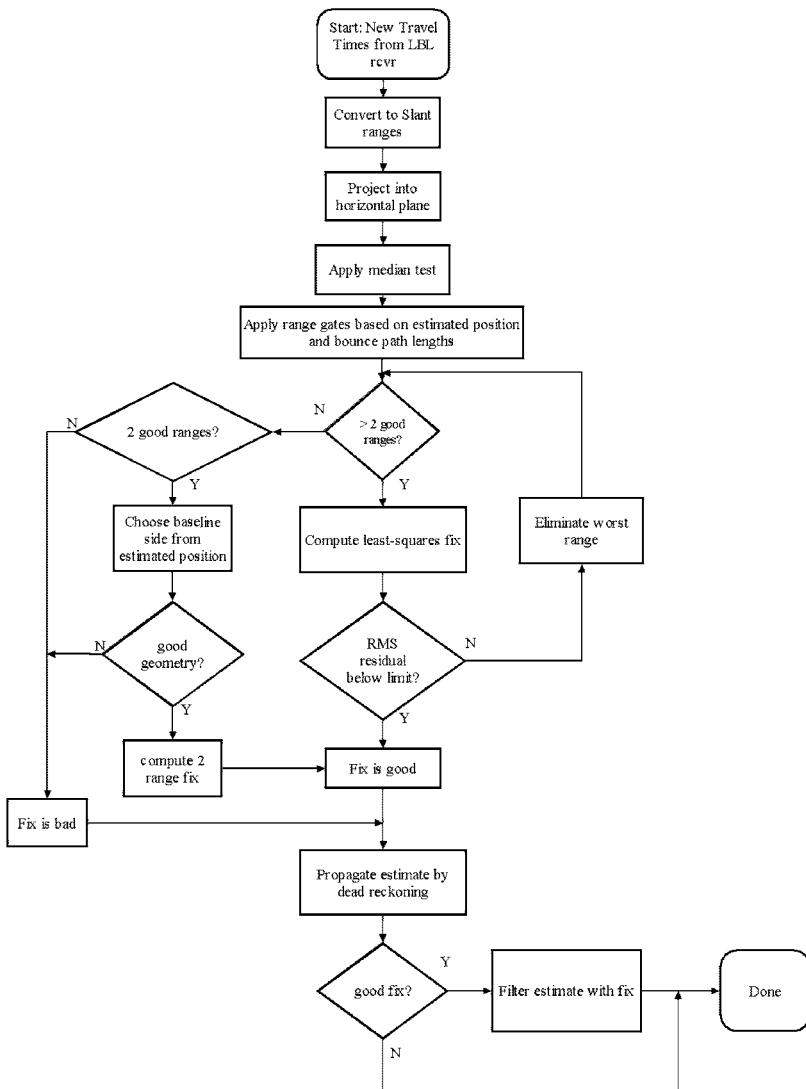


Fig. 4. Block diagram of our long baseline acoustic navigation processing algorithm. After converting travel times to ranges, a median test eliminates ranges that lack consistency over time. Consistent ranges may still be excluded before entering into the fix-computation portion of the algorithm if they fall outside of range gates derived from estimated position and expected bounce path lengths. Fixes are then computed with all available ranges; if three or more ranges are available, a least-squares solution is computed. Before a computed fix finally enters the vehicle's state estimate, it is checked for good transponder/vehicle geometry and sufficiently low residual error.

envelope. The envelope is established based on the desired depth, the desired height off bottom, and a prescribed thickness. The algorithm recomputes the envelope at each iteration based on the shortest of all reported ranges; however, it updates the depth setpoint only when the setpoint strays outside the envelope. When this happens, the algorithm recenters the commanded vehicle depth setpoint within the current envelope.

Tighter envelopes require the commanded vehicle position to follow the terrain more closely, but this requires increased vertical thruster activity and increases power consumption. When climbing or descending steadily, the depth set-point advances in steps. Additionally, forward thrust is reduced when the vehicle cannot change depth rapidly enough to stay in the envelope. Large deviations from the prescribed envelope, such as might be caused by the sudden proximal detection of a hydrothermal spire, cause the vehicle to reduce or even reverse forward thrust. The tight coupling between forward thrust and ABE's bottom-following and obstacle-avoidance performance takes advantage of ABE's ability to maintain control over a wide speed range. Figure 6 shows an example of bottom-following performance during a sonar survey at a nominal height of 40 m with a 10 m envelope.

5 Automated Nested Survey

This section describes our approach to data-driven nested survey for increasing the yield of high value data from unexplored hydrothermal vent fields. The critical component is a mapping algorithm (Fig. 7) that condenses hydrographic data acquired by our vehicle into a compact, two-tiered spatial representation of seafloor regions likely to contain active hydrothermal vents. We have implemented and field-tested the algorithm on near-bottom photo-surveys. After completing the preplanned portions of these dives, ABE used maps constructed by the algorithm to plan additional tracklines over actively venting hydrothermal structures. These autonomously-directed additional surveys yielded improved coverage over high-value targets at a fraction of the cost of additional dives. Additional dives would have been necessary had human interpretation of the data been required.

The first tier representation produced by the algorithm consists of multiple fixed-length sorted lists of hydrographic measurements classified as indicative of hydrothermally altered water, relative the rest of each data set. Prior to classification, the raw data are filtered in time to enhance signals associated with venting, to compensate for sensor dynamics, and to reduce noise. For convenience, we filter the data such that large positive values cause measurements to be selected. As each new measurement arrives, its filtered value is compared with the least positive member in the corresponding list. Favorable comparisons cause the new measurement to be stored along with the vehicle's current location and the original list member to be discarded. This component of the algorithm is executed iteratively, with list-lengths set to $O(0.1\%)$

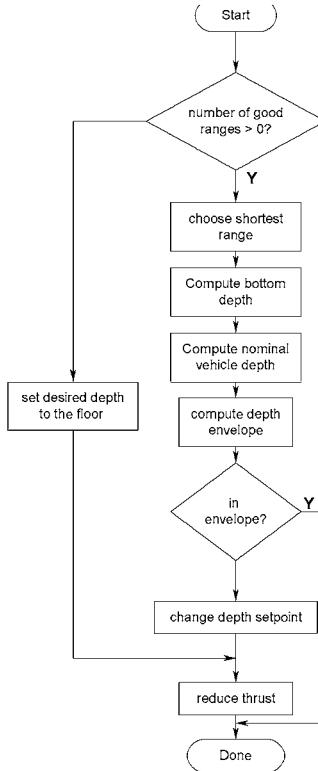


Fig. 5. ABE's bottom-following algorithm

of the total number of sensor measurements expected. The iterative implementation eliminates the need for batch processing of entire data sets, which often consist of $O(10^5)$ measurements per sensor.

The second tier consists of spatial groupings (clusters) of the selected measurements from the first tier. We use maximum inter-measurement distance as the criterion for cluster membership, set to twice the trackline spacing. Each stored measurement location is treated independently of its associated data type when deciding cluster membership. The algorithm then assigns a scalar value to each cluster that represents the relative value of revisiting the area circumscribed by that cluster (its revisit-merit). The revisit-merit of a cluster reflects both the ranking of its member measurements in their respective lists, and prior knowledge about the relative importance of each sensor in identifying hydrothermally altered water: first the filtered values stored in each list of selected measurements are normalized, then scaled according to data type, and finally summed to arrive at a scalar value for the cluster's revisit-merit. Since the first tier of the algorithm consists of

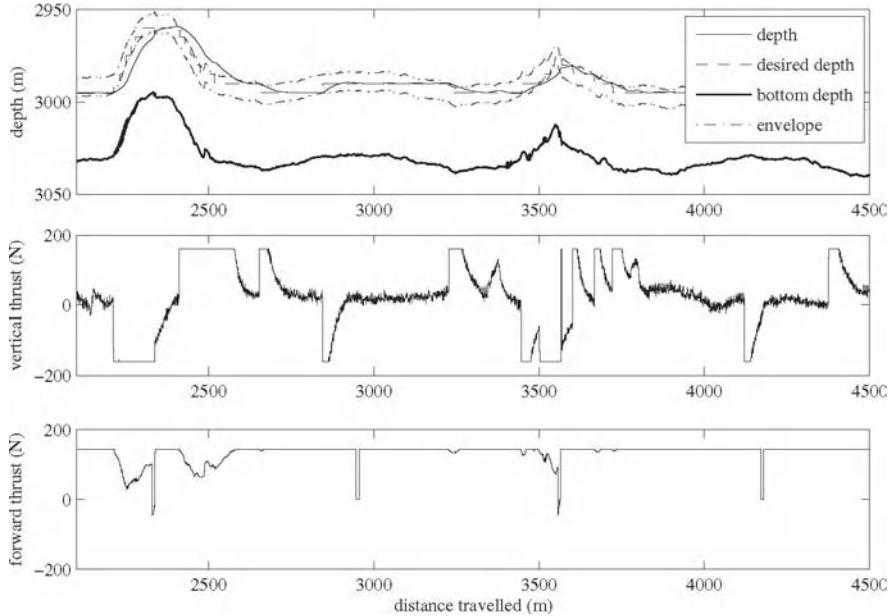


Fig. 6. Bottom following performance during a sonar survey at a height of 40 m: (**top**) vehicle depth, control envelope, and bottom trace as a function of distance traveled; (**middle**) vertical thrust; (**bottom**) forward thrust. These plots show that the bottom following algorithm (Fig. 5) makes infrequent adjustments to the commanded depth in mild terrain, driving with constant forward thrust and drawing only occasional bursts of vertical thrust

only a small fraction of the total data acquired, the computational costs of clustering are modest.

Thus far our approach to autonomous survey design based on these maps has been conservative. We have predefined the extent, spacing, and orientation of additional tracklines, leaving only absolute position to be determined in real-time as the centroid of the highest ranked cluster. Other procedures for autonomous survey design based on these maps might for instance define survey extent to cover member points such that some fraction total cluster value is included, or such that survey bounds completely encompass a cluster. These alternate strategies transfer much more of the onus of data-driven survey design to the vehicle and we anticipate they will improve performance by virtue of more tightly coupling data and trajectory generation.

Our strategy requires that an AUV survey an entire site to first build a map of high-revisitation-merit features before it makes any autonomous decisions. This approach to nested survey permits the vehicle to focus on the most interesting features, rather than relying on careful design of a threshold [16] to trigger pursuit of all potentially interesting features as they are encountered.

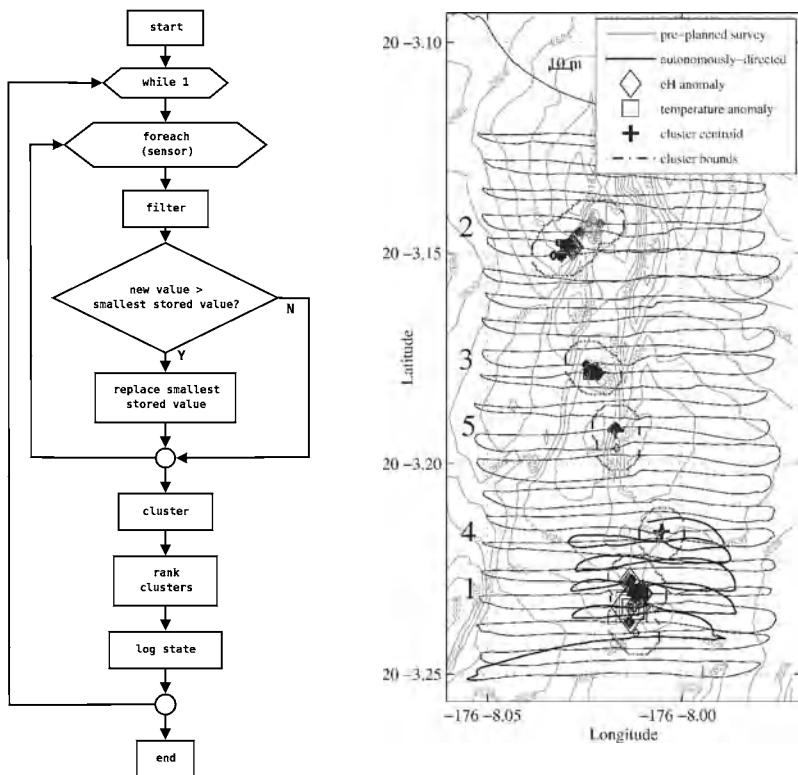


Fig. 7. Mapping in support of automated nested survey: (left) flowchart depicting the maintenance of sorted lists of selected measurements from multiple sensors, clustering by measurement location, and the ranking of the resulting clusters; (right) ranked clusters of selected hydrographic measurements. The map is shown just before the start of the autonomously directed portion of the dive. The size of each mark indicates the relative intensity of that sample

For the problem of detecting and locating hydrothermal vents, our approach may require the vehicle to cover more ground than a triggered approach would; however, the vehicle is less likely to spend time exploring false alarms or low-value targets.

To date we have applied our methodology to a total of four near-bottom photo-survey dives on two expeditions with ABE (ELSC and SMAR). Of these four dives, three returned with additional photographs of vent structures and associated fauna. A camera problem prevented the single unsuccessful survey from returning any photographs at all, but even after careful scrutiny of the intact data from the dive, the autonomously-directed portion of the unsuccessful dive became the basis of a subsequent preplanned dive. The right panel of Fig. 7 shows example results from a dive to a hydrothermal site on the ELSC. The output of the mapping algorithm is shown at the conclusion

of the preplanned portion of the dive along with the autonomously-directed portion of the dive. The latter portion comprised only 5% of total dive time, yet it yielded a 36% increase in the number of photographs with high scientific value.

Our goal on future expeditions is to apply the same map-based strategy to automate the sequence of successively higher resolution and lower altitude surveys currently employed to find hydrothermal vent sites on the seafloor with ABE (Fig. 1(a)).

6 Conclusion

We describe a set of algorithms that have been successfully used in the field with our autonomous underwater vehicle, ABE. We present our algorithm for processing acoustic long baseline returns that enables the vehicle to reliably determine its position despite the non-Gaussian noise properties of the data. We report on the algorithm that permits ABE to follow the seafloor even in very rugged terrain. Finally we report on an algorithm for automated, nested survey that permits ABE to automatically revisit sites of high interest.

We have used ABE, enabled by these algorithms, to build detailed sonar maps and photomosaics, and to locate and characterize deep sea hydrothermal vent sites. These mapping results have yielded scientifically significant data in their own right, and have enabled follow-up expeditions using remotely operated vehicles to efficiently perform detailed survey and sampling.

Acknowledgment

We thank our collaborators in the deep sea scientific community for envisioning innovative uses for ABE, and especially for recognizing its potential to provide new scientific insights unavailable or difficult to acquire through other means. In particular we gratefully acknowledge the continued support of Drs. Russell McDuff, John Delaney, Deborah Kelley, Timothy Shank, Christopher German, and Charles Langmuir. Work herein was supported by the National Science Foundation (OCE-0241913) and the National Oceanic and Atmospheric Administration (NA030AR4600115).

References

1. Fornari D, Tivey M, Schouten H, et al. (2004) Submarine lava flow emplacement processes at the East Pacific Rise 9°50' N: Implications for hydrothermal fluid circulation in the upper ocean crust. In: German C, Lin J, Parson LM (eds) The Thermal Structure of the Ocean Crust and Dynamics of Hydrothermal Circulation. AGU Geophysical Monograph 148(31)

2. Tivey MA, Johnson HP, Bradley AM, Yoerger DR (1998) Thickness of a submarine lava flow determined from near-bottom magnetic field mapping by autonomous underwater vehicle. *Geophys Res Lett* 25(6):805–808
3. Cormier M-H, Ryan WBF, Shah A, et al. (2003) Waxing and waning volcanism along the East Pacific Rise on a millennium timescale. *Geology* 31(7):633–636
4. Schouten H, Tivey M, Fornari D, et al. (2003) Central anomaly magnetization high: Constraints on the volcanic construction and architecture of young upper oceanic crust, EPR 9–10° N. Ridge 2000 Events 1:30–34
5. Carbotte SM, Ryan WBF, Jin W, et al. (2003) Magmatic subsidence of the EPR at 18°14' S revealed through fault restoration of ridge crest bathymetry. *Geochem Geophys Geosyst* 4(1):1008, doi:10.1029/2002GC000337
6. Kelly DS, Karson JA, Früh-Green GL, et al. (2005) A serpentinite-hosted ecosystem: The Lost City hydrothermal field. *Science* 307(5714):1428–1434
7. Stahr F, McDuff R, Yoerger D, Bradley A, Nakamura K (2000) Heat flux measurements at the Main Endeavour Vent Field, Juan de Fuca Ridge. In: *Eos Trans AGU*, Fall Meet Suppl 81(48), Abstract OS52I-03
8. Yoerger DR, Bradley AM, Stahr F, McDuff R (2001) Surveying deep-sea hydrothermal vent plumes with the Autonomous Benthic Explorer (ABE). In: *Proceedings of the 12th International Symposium on Unmanned Untethered Submersible Technology*
9. Langmuir CH, German C, Michael P, et al. (2004) Hydrothermal prospecting and petrological sampling in the Lau Basin: Background data for the Integrated Study Site. *Eos Trans AGU*, Fall Meet Suppl 85(47), Abstract B13A-0189
10. German CR, Connelly DP, Prien RG (2005) New techniques for hydrothermal plume investigation by AUV. *Geophys Res Abs, EGU General Assembly*, Abstract EGU05-A-04361
11. Nakamura K, Veirs S, Sarason CP, et al. (2000) Electrochemical signals in rising buoyant plumes and tidally oscillating plumes at the Main Endeavour Vent Field, Juan de Fuca Ridge. In: *Eos Trans AGU*, Fall Meet Suppl 81(48), Abstract OS52I-05
12. Pizarro O, Singh H (2003) Toward large area mosaicing for underwater scientific applications. *IEEE J Ocean Eng* 28(4):651–672
13. Hunt MM, Marquet WM, Moller DA, Peal KR, Smith WK (1974) An Acoustic Navigation System. Tech Rep WHOI-74-6, Woods Hole Oceanographic Institution, Woods Hole, Massachusetts
14. Milne PH (1983) Underwater Acoustic Positioning Systems. Gulf Publishing Company, Houston
15. Deffenbaugh M, Schmidt H, Bellingham JG (1996) Acoustic positioning in a fading multipath environment. In: *Proceedings IEEE/MTS Oceans'96*:596–600
16. Camilli R, Bingham B, Jakuba M, Singh H, Whelan J (2004) Integrating in situ chemical sampling with AUV control systems. In: *Proceedings Oceans'04 MTS/IEEE Techno-Ocean'04* 1:101–109

Advances in High Resolution Imaging from Underwater Vehicles

Hanumant Singh¹, Christopher Roman^{1,2}, Oscar Pizarro^{1,3}, and Ryan Eustice^{1,4}

¹Woods Hole Oceanographic Institution, USA hanu@whoi.edu

²University of Rhode Island, USA cnr@gso.uri.edu

³Australian Centre for Field Robotics o.pizarro@cas.edu.au

⁴University of Michigan, USA eustice@umich.edu

Summary. Large area mapping at high resolution underwater continues to be constrained by the mismatch between available navigation as compared to sensor accuracy. In this paper we present advances that exploit consistency and redundancy within local sensor measurements to build high resolution optical and acoustic maps that are a consistent representation of the environment.

We present our work in the context of real world data acquired using Autonomous Underwater Vehicles (AUVs) and Remotely Operated Vehicles (ROVs) working in diverse applications including shallow water coral reef surveys with the Seabed AUV, a forensic survey of the RMS Titanic in the North Atlantic at a depth of 4100 meters using the Hercules ROV and a survey of the TAG hydrothermal vent area in the mid-Atlantic at a depth of 2600m using the Jason II ROV.

Specifically we focus on the related problems of Structure from Motion and Visually Based Navigation from underwater optical imagery assuming pose instrumented calibrated cameras. We present general wide baseline solutions for these problems based on the extension of techniques from the SLAM, photogrammetric and the computer vision communities. We also examine how such techniques can be extended for the very different sensing modality and scale associated with multi-beam bathymetric mapping. For both the optical and acoustic mapping cases we also show how the consistency in mapping can be used not only for better mapping but also to refine navigation estimates.

1 Introduction

A number of oceanographic applications require large area site surveys from underwater imaging platforms. Such surveys are typically required to study hydrothermal vents and spreading ridges in geology [1], ancient shipwrecks and settlements in archaeology [2], forensic studies of modern shipwrecks and airplane accidents [3,4], and surveys of benthic ecosystems and species in biology [5]. Scientific users in these disciplines often rely on multi-scalar, multi-sensor measurements to best characterize the environment.

At finer scales, for resolutions down to millimeters, optical imaging of the seafloor offers scientists a high level of detail and ease of interpretation. However, light underwater suffers from significant attenuation and backscatter, limiting the practical coverage of a single image to a few square meters. To cover larger areas of interest, hundreds or thousands of images may be required. The rapid attenuation of the visible spectrum in water implies that a composite view of a large area (or photomosaic) can only be obtained by exploiting the redundancy in multiple overlapping images distributed over the scene. Although there has been considerable effort in this regard for land-based applications, the constraints on imaging underwater are far different. Mosaicing assumes that images come from an ideal camera (with compensated lens distortion) and that the scene is planar. Under these assumptions the camera motion will not induce parallax and therefore no 3D effects are involved and the transformation between views can then be correctly described by a 2D homography. These assumptions often do not hold in underwater applications since light attenuation and backscatter rule out the traditional land-based approach of acquiring distant, nearly orthographic imagery. Underwater mosaics of scenes exhibiting significant 3D structure usually contain significant distortions. In contrast to mosaicing, the information from multiple underwater views can be used to extract structure and motion estimates using ideas from structure from motion (SFM) and photogrammetry.

For coarser resolutions ($O(10\text{cm})$), but covering far greater ($O(10\text{m}-100\text{m})$) swaths, acoustic sensing centered at several hundred kilohertz is the modality of choice. Multi-beam sensors mounted on underwater platforms can provide high resolution three dimensional scans of the environment that can be transformed into bathymetric maps.

Unfortunately for both optical and acoustic sensors, the fundamental limitation in converting high resolution sensor measurements into quantitative maps is the mismatch between sensor accuracy and navigation as illustrated schematically in Figure 1. Due to the rapid attenuation of the electromagnetic spectrum, GPS signals are not available underwater. Instead underwater imaging platforms typically rely on a combination of acoustic transponders and inertial navigation systems. Acoustic transponders [6], like sonar systems, must trade off range for resolution. Although transponders have been built to work as high as 300kHz providing centimeter level accuracy over an area of 100 square meters, the typical large area surveys utilize lower frequency (8-13 kHz) long-baseline transponders that provide meter level accuracy across several kilometers. The deployment of such systems is nontrivial and usually requires significant time and effort as each individual transponder must be deployed and its position independently calibrated.

Inertial navigation sensors such as Doppler velocity logs used in combination with fiber optic or ring laser gyros can provide navigation estimates underwater [7] that grow as a function of time (distance traveled). However, such systems inherently provide an estimate whose error characteristic grows without bound over time (distance). Although expensive, from a cost, power and size standpoint, these systems are far easier to use as they are integral to the underwater vehicle and as such do not require any extra effort for deployment and use.

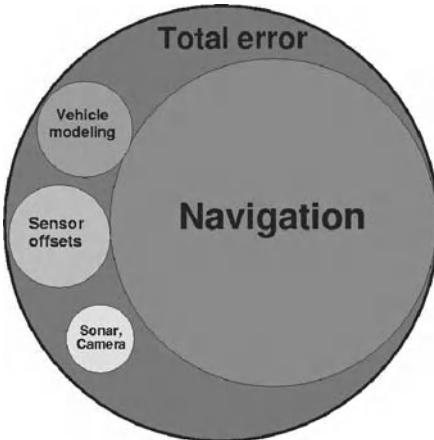


Fig. 1. A schematic of error sources for high resolution optical and acoustic deep water mapping. Vehicle based mapping is navigationally limited in comparison to other potential error sources.

2 Structure from Motion Underwater – The Two-View Case

As outlined above, the fundamental problem of obtaining a large area perspective of an underwater scene is constrained by the attenuation and backscatter of light, the highly unstructured nature of underwater terrain, and issues associated with moving lighting on underwater robotic vehicles.

Our methodology for Structure from Motion takes a local to global approach inspired by mosaicing and other land-based applications of SFM [8-12] but differs in that it takes advantage of navigation and attitude information [13]. Local sequences are derived independently [11] and then registered in a global frame for bundle adjustment [10]. Our approach seems more suitable than pure sequential methods [9,11] because in an un-

derwater survey each 3D feature appears only in a few images making the global solution look more like a series of weakly correlated local solutions.

We relate images using a feature-based approach under wide-baseline imaging conditions with changing illumination and unknown scene structure. A modified Harris corner detector yields interest points by selecting local maxima of the smaller eigenvalue of the second moment matrix. We extract features by determining a neighborhood around each interest point that is invariant to affine geometric transformations. In essence, we sample the neighborhood along lines radiating from the interest point. For each line we select the extrema of an affine invariant function (maximum difference in intensities between the interest point and points along the ray). The set of these maximal points defines the boundary of a region that can be extracted under affine geometric transformations. This region is approximated with an elliptical neighborhood which is then mapped onto the unit circle. These circular patches are normalized for affine photometric invariance. Features are then represented compactly using moment-based descriptors. We chose to use Zernike [13] moments as descriptors for their compactness and highly discriminative nature.

The core of the algorithm for SFM is based on robust estimation of the essential matrix [8]. Similarity of descriptor vectors is used to propose correspondences between features. The navigation-based estimates of inter-image motion and vehicle altitude are used to limit possible correspondences by propagating pose and altitude uncertainties through the two view point-transfer equation [13] as shown in Figure 2. A modified version of RANSAC determines the correspondences which are consistent with that essential matrix and the essential matrix consistent with the inliers as illustrated in Figure 3. In cases of multiple valid solutions we select the one closest (in the Mahalanobis distance sense) to the navigation-based prior. The inliers and the essential matrix estimate are used to produce a maximum a posteriori estimate of relative pose with the navigation-based estimates as a prior. The solution includes the triangulated 3D features.

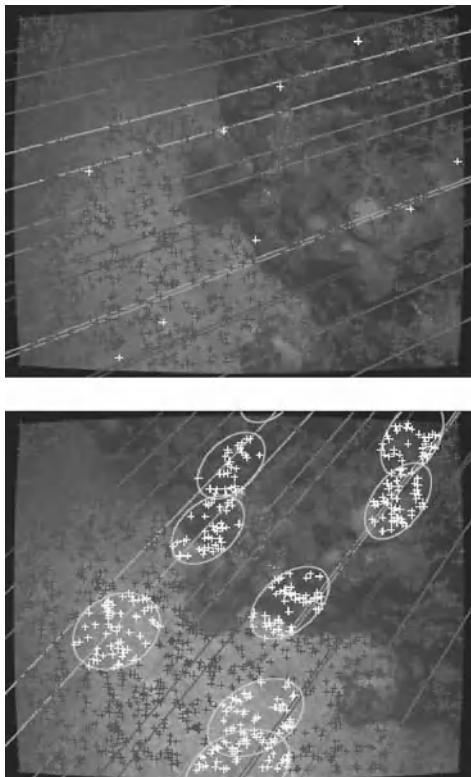


Fig. 2. Prior pose restricted correspondence search on a pair of underwater coral reef images. A sampling of interest points are shown in the top image along with their color coded sensor instantiated epipolar lines. The bottom image shows the corresponding color coded constrained search regions for the interest points in the top image; the sensor instantiated epipolar lines; and the candidate interests points which fall within these regions.

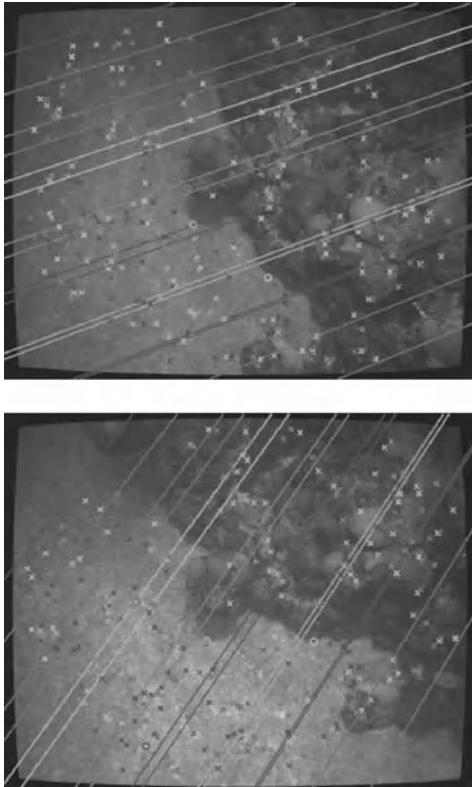


Fig. 3. Epipolar geometry and correspondences. The given image pair illustrates the maximum likelihood refined image-based epipolar geometry. RANSAC determined 106 consistent inliers designated ‘x’, from the putative set of 116 matches. The rejected outliers are designated ‘o’.

3 Large Area Structure from Motion

The temporal sequence of images is processed into a set of 3D submaps with estimates of coordinate transformations between temporally adjacent submaps (Figure 4). This can be viewed as a graph where each node is the origin of a submap and the edges in the graph are the coordinate transformations between submaps. Our algorithm attempts to establish additional spatial relationships between submaps corresponding to overlap from parallel tracklines or loop closures.

While the sparse set of 3D points contained in the submaps do not consistently offer discriminating structure, the very fact that they exist as 3D

points implies that their appearance in multiple views is characteristic enough to effectively establish correspondences and be reconstructed by the SFM algorithm. We therefore extend the feature description and similarity based matching between images to matching submaps by relying on the appearance of 3D points to propose corresponding features between submaps. The average of the descriptors of the 2D neighborhoods on all views is used as the appearance of the 3D point. The underlying assumption is that a similarity measure which was effective to match 3D points along track will also be effective when matching across submaps. Corresponding 3D points are proposed based on appearance and a robust registration using RANSAC with Horn's algorithm [20] is used to determine which points are in correspondence and the transformation parameters.

The search of additional links continues until no links are left to check or an upper limit is reached (we use eight times the number of submaps for sparse, locally-connected graphs). The submaps are then placed in a global frame by minimizing the discrepancies between composed global estimates and the transformations between submaps. Additional cost terms consider the navigation prior.

Once submaps are in a global frame, camera poses within submaps can also be placed in the global frame. These camera poses are then used to triangulate the location of 3D features. Sparse bundle adjustment then refines both camera poses and 3D feature locations.

Figure 5 illustrates this process. The results are from a survey performed in the Johns Hopkins University (JHU) Hydrodynamics Test Facility using the JHU ROV. As shown in the figure the results were highly consistent with ground-truth obtained by draining the test tank and laser scanning the scene geometry. We have also obtained similar results from a survey using the Seabed AUV at a coral reef off of Bermuda [5].

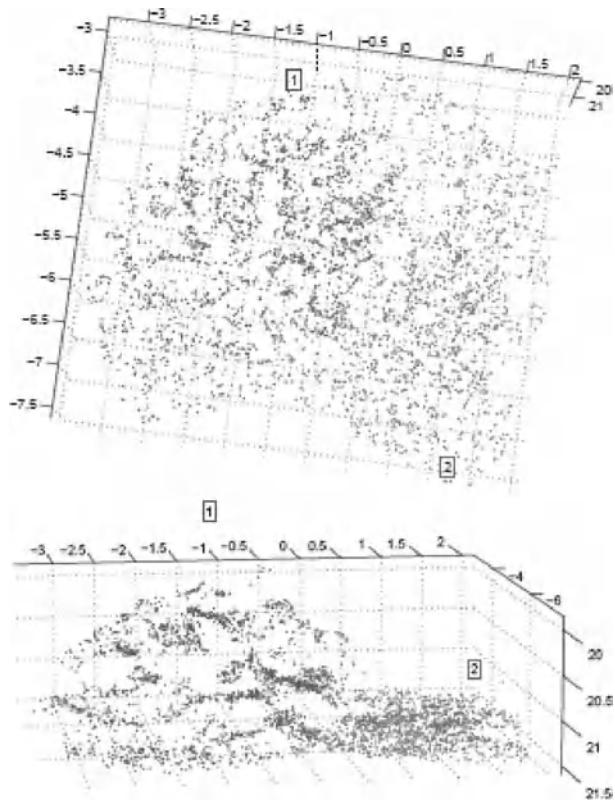


Fig. 4. Two views of the registered submaps derived from images sequences that correspond to two neighboring sections along the images shown in figures 2 and 3. The blue and green dots correspond to features from the neighboring tracklines that have been successfully co-registered.

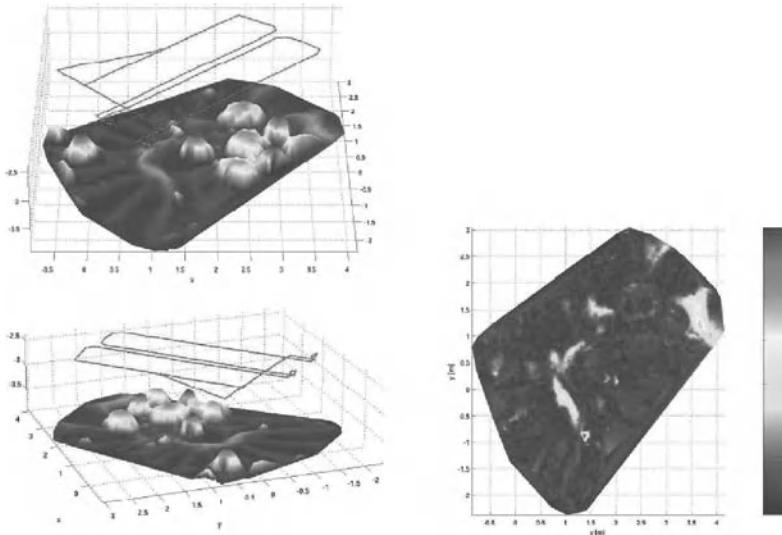


Fig. 5. (Left) Two views of the reconstruction of poses and structure for the JHU tank dataset. The camera poses are connected by a red line. A Delaunay triangulation interpolates a surface between 3D feature points. The structure is color-coded according to height. Units are in meters. (Right) Distance map from SFM 3D points to the ground-truth laser scan after ICP registration. Areas of large discrepancies tend to correspond to the carpet being buoyant for the visual survey. An outlier in the reconstruction produced the large error visible at approximately $x=1.4$ m, $y=0.8$ m.

4 Self Consistent Bathymetric Mapping

Another application of our techniques arises from the case of multi-beam mapping [16] where the areas of interest encompass several square kilometers that are typically mapped with a sonar with ten centimeter sensor accuracy but where the navigation from the standard combine of long baseline transponders and inertial navigation is only good to a meter. To avoid this navigation limitation we break the total mapping problem down into small pieces, each of which contains internal errors typical of the mapping sonar rather than the navigation [19]. This is accomplished by assembling small bathymetry sub-maps using only the short term dead reckoning information provided by the vehicle navigation sensors. Algorithmically this is accomplished using a delayed state Extended Kalman Filter (EKF) and a simple constant velocity dynamic model of the vehicle motion. This simple model is sufficient given the slow dynamics typical of underwater sur-

vey vehicles. The current estimate of the filter state vector contains the position and velocity information required for a 6 degree of freedom (DOF) state estimate. The delayed portion of the state vector is used to archive historical poses of the vehicle which serve as local 6 DOF origins for the small sub-maps.

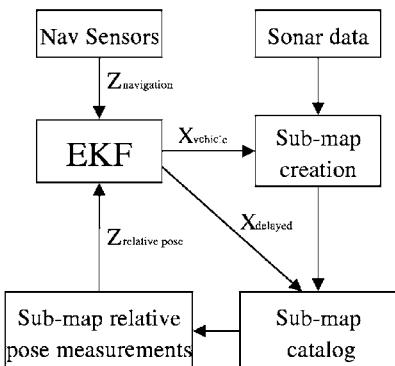


Fig. 6. The delayed state EKF block diagram. The sub-mapping algorithm utilizes vehicle navigation data to create small bathymetric sub-maps. The sub-map origins are held in a delayed state vector and used to create relative pose measurements that reduce navigation error.

After accounting for issues specific to acoustic sensors such as possible errors associated with weak returns, beam patterns effects resulting in the acoustic pulse not striking the bottom, and other false detections, we can approximate the sonar as a three dimensional line scanner. These line scans are assembled into sub-maps using the range data and the vehicle position estimates extracted from the state vector at the time each sonar ping is taken (Figure 6).

The individual beam ranges are projected and kept as points (soundings) in 3D dot clouds referenced to the delayed state local origins (Figure 7). Sub-maps are sized by collecting pings in this manner over short time scales during which the vehicle position error associated with the inertial navigation is considered small. A sub-map is broken, and a new one started, when the one of several condition are met. A map will be broken when the terrain contained in the map has sufficient 3D structure, the map has become too large or has poor outline shape, or when the estimate of vehicle position uncertainty relative to the map origin becomes greater than a threshold consistent with mapping sonar accuracy. The position error based end condition is designed to keep the navigation errors from corrupting the mapping data.

The mapping error growth associated with navigation errors across the entire survey is reduced by registering the terrain sub-maps to one another and generating relative position measurements between previously visited vehicle states. The sub-maps are compared and aligned using a 2 DOF translation based on correlation followed by a 6 DOF alignment with a modified Iterative Closest Point (ICP) [17,18] approach. Improved registration results are obtained by selecting matching points bases on the quality of the individual sonar returns [19,21,22]. The end result of the algorithm is a constraint network, between the sub-map origins (Figure 8). This network enforces consistency based on the sub-map alignments and helps to significantly reduce the inconsistency that would be present if navigation was used alone to produce the terrain map.

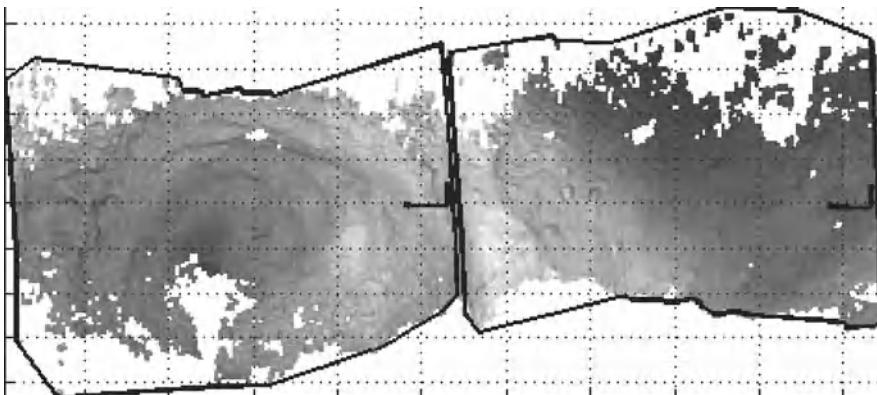


Fig. 7. Two sample sub-maps showing their outlines and local reference frames. Note the fine scale features that can be resolved individually within each sub-map. Normal smoothed navigation tends to blur these real and often significant features. cf Figure 9.

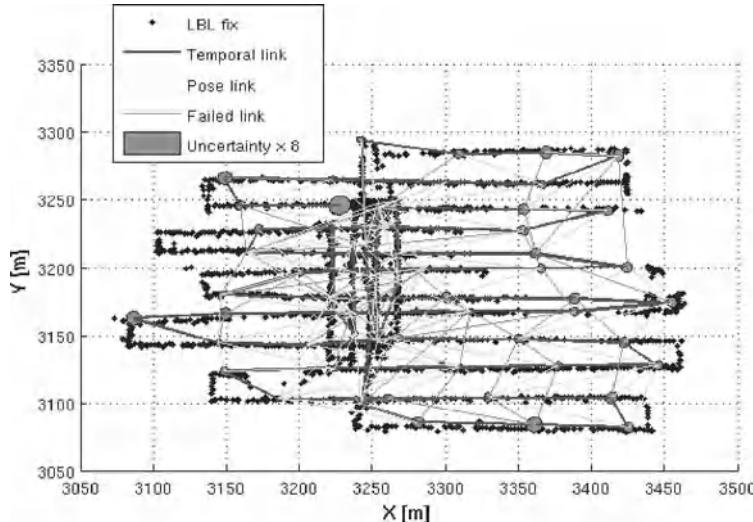


Fig. 8. Sub-mapping pose network. This pose network was established by the sub-mapping algorithm. Nodes indicate the location of the sub-map origins. Blue links indicate consecutive poses in time. Green links indicate where relative pose measurements were made. Magenta links indicate links that were tried but not established. The uncertainty ellipses have been scaled in size by 8 times for visibility. Note that the poses fall into alignment with the LBL fix locations even though this algorithm did not utilize LBL measurements. This survey consisted of 62 sub maps and 92 established links.

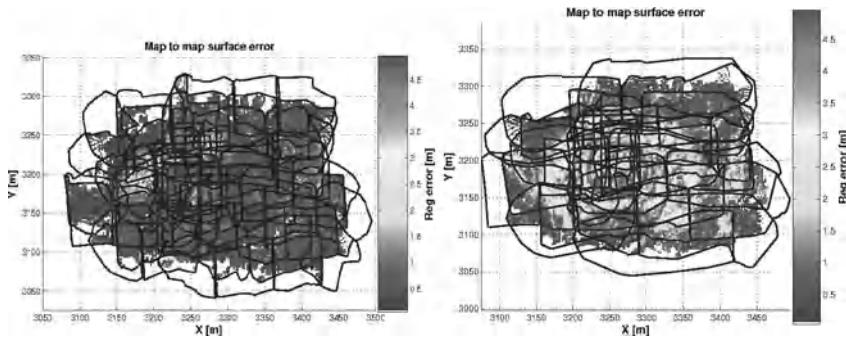


Fig. 9. Error in bathymetric mapping as measured by self consistency across multiple sub-maps. (Left) Map to map surface error for our algorithm versus (right) map to map surface error using a standard smoothing method.

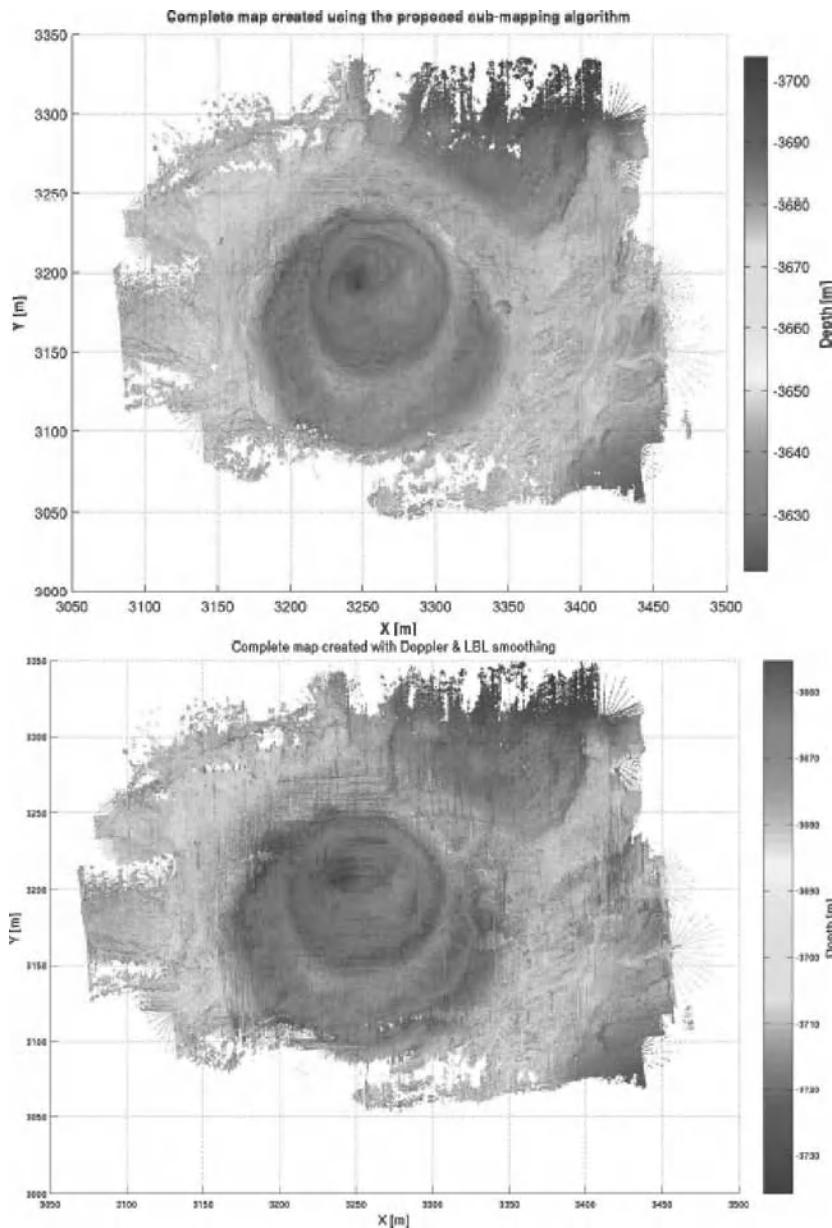


Fig. 10. A comparison between the terrain map created using sub-mapping, (top), and a version of the map created using a standard smoothing method (bottom). The sub-map created map shows significantly less registration error and sonar scan patterning. The sub-mapped version also brings out details that were lost in the smoothed map due to mis-registration.

Another application of our techniques arises from the case of multi-beam mapping [21] where the areas of interest encompass several square kilometers that are typically mapped with a sonar with ten centimeter sensor accuracy but where the navigation from the standard combination of long baseline transponders and inertial navigation is only good to a meter.

The results of our framework are illustrated in Figures 9 and 10 using data collected by the Jason ROV at the TAG Hydrothermal Vent Site located at a depth of 2600 meters on the mid-ocean ridge in the Atlantic Ocean. One can see that the resulting map is a far better representation of the environment. We have also used consistency within the submaps to derive corrected navigation estimates for the vehicle trajectory over the course of the survey.

5 Visually Augmented Navigation

We can further build upon the delayed-state EKF framework and two-view structure from motion case to formulate a vision-based simultaneous localization and mapping (SLAM) approach to providing high precision, accurate navigation measurements on an underwater robotic vehicle. Similar to the bathymetry-based EKF submapping strategy, our methodology is to employ pairwise-generated image measurements as spatial constraints in a graph over a collection of historical vehicle poses. However, because we are able to generate camera measurements at the pairwise level, we choose to instead maintain all pose samples that are associated with image acquisition (Figure 11). This differs from the aggregate submapping strategy used for bathymetry-based navigation and implies that the EKF's scalability becomes a severe issue (due to the quadratic complexity of maintaining the covariance) as the image-based navigation uses orders of magnitude more delayed-states.

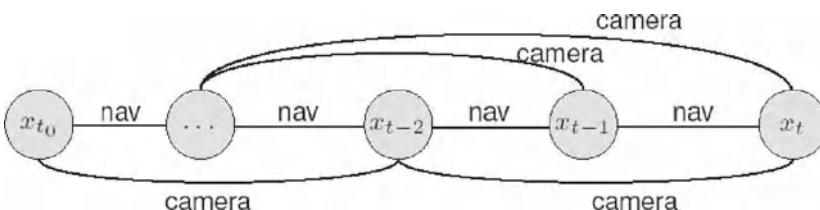


Fig. 11. The above diagram is a conceptual depiction of the delayed-state graph network and its constraints.

A well known and very attractive property of formulating SLAM in the information form is that the information matrix (as in Fisher information) has the direct interpretation as a Gaussian graphical model [23,24]. Sparsity in this model (i.e., missing edges) implies available conditional independencies in the joint-distribution, which can be exploited to realize efficient inference. While others have shown that the feature-based SLAM information matrix obeys a "close-to-sparse" structure when properly normalized [14,25] in our formulation of view-based SLAM [15], the information matrix is exactly sparse without having to make any sparse approximations. This implies that for a bounded graph structure, as is the case with typical underwater surveys, view-based SLAM systems comprise a sparse information parameterization without incurring any sparse approximation error.

Based upon this insight, we have implemented a view-based SLAM system for underwater applications built around fusing 6-DOF relative-pose camera measurements from monocular overlapping seafloor imagery with traditional underwater vehicle dead-reckon navigation sensors. Our state vector consists of samples from the robot's trajectory acquired at image acquisition and is maintained using an exactly sparse delayed-state filter (ESDF) [15]. We use our two-view image registration engine to provide non-Markov edge constraints in the corresponding pose network. These "spatial" edges constrain the pose graph and enforce global consistency from local constraints. This system was tested with data collected using the Hercules ROV operating at a depth of 3750 meters at the wreck of the RMS Titanic. The survey covered an area of about 3100 square meters on the seafloor with a accumulated survey path length over 3.4 kilometers. Results are shown in Figure 12.

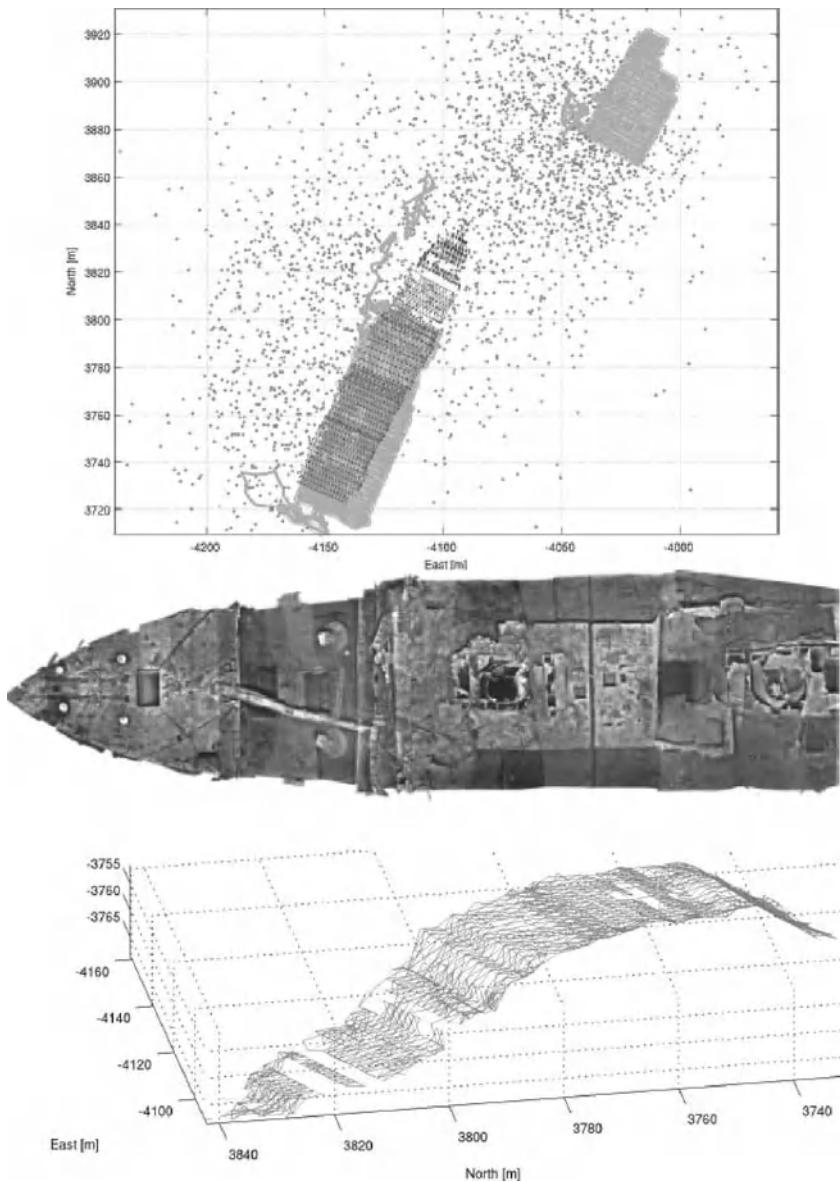


Fig. 12. Results from visually mapping the RMS Titanic. (Top) An XY plot comparing the raw dead-reckon navigation data (gray), ship-board ultra-short baseline tracking (brown), and reconstructed survey trajectory from a vision-based 6 DOF SLAM information filter (red). (Middle) A photomosaic of the RMS Titanic constructed from the digital still images and (Bottom) the complete 6 DOF visually based navigation results for the entire survey.

6 Conclusions

In this paper we have highlighted some of the fundamental issues associated with the lack of precise and accurate navigation and how they affect our ability to conduct high resolution mapping efforts in the deep sea. We have presented three different applications of systems level approaches for deep water mapping that exploit vehicle attitude and navigation information and enforce local and global consistency within sensor measurements to yield superior mapping results commensurate with sensor accuracy. While improving mapping fidelity these methods also provide us with independent mechanisms for ground-truthing, refining and bounding the coarse navigation estimates that are typical in the deep ocean.

These algorithms are applicable across the entire suite of imaging and robotic underwater vehicles – manned, towed, tethered and autonomous. Our work in these areas is continuing with an emphasis on implementing a number of these algorithms in real-time on board the vehicles to better help us exploit our precision mapping algorithms for real-time adaptive surveys.

Acknowledgements

The authors gratefully acknowledge our multitude of collaborators on the following research expeditions from which data for this paper was gathered – Dr. Rob Reves-Sohn (Bermuda Coral and TAG), Prof. Louis Whitcomb (JHU Hydrodynamics Facility), and Prof. Robert Ballard (Titanic). This work was funded in part by a grant from the National Science Foundation’s CenSSIS ERC under grant # EEC-9986821.

References

1. D.R. Yoerger, A.M. Bradley, M.-H. Cormier, W.B.F. Ryan, and B.B. Walden. Fine-scale seafloor survey in rugged deep-ocean terrain with an autonomous robot. In IEEE International Conference on Robotics and Automation, volume 2, pages 1767–1774, San Francisco, USA, 2000.
2. R.D. Ballard, L.E. Stager, D. Master, D.R. Yoerger, D.A. Mindell, L.L. Whitcomb, H. Singh, and D. Piechota. Iron age shipwrecks in deep water off Ashkelon, Israel. American Journal of Archaeology, 106(2):151–168, April 2002.
3. J. Howland. Digital Data Logging and Processing, Derbyshire Survey, 1997. Technical report, Woods Hole Oceanographic Institution, December 1999.

4. National Transportation Safety Board, Washington, DC. Aircraft Accident Brief: EgyptAir Flight 990, Boeing 767-366ER, SU-GAP, 60 Miles South of Nantucket, Massachusetts, October 31, 1999, 2002. Aircraft Accident Brief NTSB/AAB-02/01.
5. H. Singh, R. Eustice, C. Roman, O. Pizarro, R. Armstrong, F. Gilbes, and J. Torres. Imaging coral I: Imaging coral habitats with the SeaBED AUV. *Subsurface Sensing Technologies and Applications*, 5(1):25–42, January 2004.
6. P. H. Milne, Underwater Acoustic Positioning Systems. Gulf Publishing Company, Houston, 1983.
7. L. Whitcomb, D. Yoerger, and H. Singh, .Advances in Doppler-Based Navigation of Underwater Robotic Vehicles,. in Proceedings of the 1999 International Conference on Robotics and Automation, vol. 1, 1999, pp. 399.406.
8. R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2000.
9. M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Hand-held acquisition of 3d models with a video camera. In Second InternationalConference on 3-D Digital Imaging and Modeling, pages 14–23, Los Alamitos, CA, 1999. IEEE Computer Society Press.
10. B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, “Bundle Adjustment - a Modern Synthesis,” in *Vision Algorithms: Theory and Practice*, ser. LNCS, W. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer-Verlag, pp. 298–375. 2000
11. A.W. Fitzgibbon and A. Zisserman, Automatic Camera Recovery for Closed or Open Image Sequences, *Proceedings of the 5th European Conference on Computer Vision*, pages 311–326, Freiburg, Germany, June 1998.
12. C.C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry, Bethesda, MD, fourth edition, 1980.
13. O. Pizarro, R. Eustice, and H. Singh, “Relative Pose Estimation for Instrumented, Calibrated Imaging Platforms,” in *Proceedings of Digital Image Computing Techniques and Applications*, Sydney, Australia, pp. 601–612, December 2003.
14. S. Thrun, Y. Liu, Z. Koller, H. Ghahramani, H. Durrant-Whyte, and A.G. Ng, .Simultaneous mapping and localization with sparse extended information filters., *Internation Journal of Robotics Research*, vol. 23, pp. 693.716, 2004.
15. R. Eustice, H. Singh, and J. Leonard, .Exactly Sparse Delayed-State Filters,. in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, SPAIN, April 2005.
16. H. Singh, L. Whitcomb, D. Yoerger, and O. Pizarro, .Microbathymetric Mapping from Underwater Vehicles in the Deep Ocean,. *Computer Vision and Image Understanding*, vol. 79, no.1, pp.143.161, 2000.
17. P.J. Besl and N.D. McKay, A method for registration of 3-d shapes, *IEEE PAMI*, vol. 14, pp. 239-256, 1992.
18. S. Rusinkiewic and M. Levoy, Efficient variants of the icp algorithm, in *Proceedings IEEE 3DIM*, Canada, 2001, pp. 145.152.

19. C. Roman, Self Consistent Bathymetric Mapping from Robotic Vehicles in the Deep Ocean, Ph.D. Thesis, MIT/WHOI Joint Program in Oceanographic Engineering, May 2005.
20. B. Horn, .Closed-Form Solution of Absolute Orientation Using Unit Quaternions., Journal of the Optical Society of America, vol. 4, no. 4, pp. 629.642, April 1987.
21. C. Roman, H. Singh, Improved vehicle based multibeam bathymetry using sub-mapping and SLAM, IEEE Intelligent Robotics and Systems (IROS), 2005 to appear.
22. H. Medwin, H.C. Clay, Fundamentals of Acoustical Oceanography, Academic Press, London, 1998.
23. J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufman Publishers, San Mateo, CA. 1988.
24. Y. Weiss, W.T. Freeman, Correctness of belief propagation in Gaussian Graphical Models of Arbitrary Topology. Neural Computation, 13(10):2173-2200, 2001.
25. U. Frese, A Proof for the Approximate Sparsity of SLAM Information Matrices, Proceedings of IEEE International Conference on Robotics and Automation, pp 331-337, Barcelona, Spain, 2005.

Learning and Adaptive Behavior

Session Overview

Learning and Adaptive Behavior

Paolo Dario

Scuola Superiore Sant'Anna, Pisa, Italy

In the evolution of robotics, robots have been increasingly operating in a variety of environments, unstructured, and dynamically changing over time. It has been clear since the first progresses of advanced robotics how the capability of perceiving the environment and of behaving accordingly is critical for robots.

Learning and adaptive behavior are therefore basic capabilities for most categories of robots, being them applied in services, in assistance, in autonomous tasks.

Though different in shape and functions, most robots share a common mechatronic structure, which integrates mechanisms, actuators, sensors, electronics, control, and power supply. Such structure interacts on one side with the external world and with a human user, on the other.

Learning and adaptive behavior is first of all needed for (autonomous) interaction with dynamic environments.

Considering the closer and closer interaction that robots have with human beings, learning and adaptive behavior is becoming increasing important also in the interaction with the human user. Robots are going to become more similar to personal assistants than to appliance, and a fruitful interaction requires that the robot and the user know each other, learn each other's habits and preferences, and adapt to each other.

Finally, biological inspiration in robotics is leading to complex structures with many sensor and unconventional actuators. Still following a biological inspiration, learning often plays an important role in the development of sensory-motor coordination on such structures. Furthermore, leading-edge research is also investigating the development of mechatronic structures that can change over time and evolve, so that it is required that robots have the capability to adapt to the modifications of their own bodies.

The papers in the ISRR 2005 session on "Learning and Adaptive Behavior" presented new advances in different aspects of this area.

The first paper, "Semantic labeling of places", by Cyrill Stachniss, Oscar Martinez Mozos, Axel Rottmann, and Wolfram Burgard, presented a novel

approach to classify different places in the environment of a mobile robot, into semantic classes. This approach was successfully validated both in simulation and with real robots. The algorithm developed by the authors allows a mobile robot to identify typical environmental places, like rooms, hallways, doorways, and other, and to classify them into semantic classes. The robot thus gets the capability of interpreting its operational environment and also to share semantic knowledge with human users.

The second paper, by Yasuo Kuniyoshi, Shinsuke Suzuki, and Kyosuke Shiozumi, presented “Emergence, Exploration and Learning of Embodied Behavior”. In this case, learning is adopted to control a complex musculo-skeletal system. A novel model is proposed for dynamic emergence and adaptation of embodied behavior, based on a number of chaotic elements, each driving a muscle with local sensory feedback. The results obtained with a simulation confirm that modeling the musculo-skeletal structure as a couple chaotic system allows to obtain emergent ordered patterns that correspond to motor coordination patterns, able to re-organize in response to environmental changes.

The session was closed by the paper “Extracting places and activities from GPR traces”, by Lin Liao, Dieter Fox, and Henry Kautz. The authors presented a novel approach to extract, simultaneously, activities and places of a person, from GPS data. Learning is used to identify patterns of human behavior, which may results useful in a variety of human-robot interaction contexts.

Using AdaBoost for Place Labeling and Topological Map Building

Óscar Martínez Mozos, Cyrill Stachniss, Axel Rottmann, and Wolfram Burgard

University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany
omartine|stachnis|rottmann|burgard@informatik.uni-freiburg.de

Summary. Indoor environments can typically be divided into places with different functionalities like corridors, kitchens, offices, or seminar rooms. We believe that the ability to learn such semantic categories from sensor data or in maps enables a mobile robot to more efficiently accomplish a variety of tasks such as human-robot interaction, path-planning, exploration, or localization. In this work, we first propose an approach based on supervised learning to classify the pose of a mobile robot into semantic classes. Our method uses AdaBoost to boost simple features extracted from vision and laser range data into a strong classifier. We furthermore present two main applications of this approach. Firstly, we show how our approach can be utilized by a moving robot for robust online classification of the poses traversed along its path using a hidden Markov model. Secondly, we introduce a new approach to learn topological maps from geometric maps by applying our semantic classification procedure in combination with probabilistic labeling. Experimental results obtained in simulation and with real robots demonstrate the effectiveness of our approach in various environments.

1 Introduction

In the past, many researchers have considered the problem of building accurate maps of the environment from the data gathered with a mobile robot. The question of how to augment such maps by semantic information, however, is virtually unexplored. Whenever robots are designed to interact with their users, semantic information about places can be important. It can furthermore be used to intuitively segment an environment into different places and learn accurate topological models.

In this work, we address the problem of classifying places of the environment of a mobile robot using range finder and vision data as well as building topological maps based on that knowledge. Indoor environments, like the one depicted in Figure 1, can typically be divided into areas with different functionalities such as laboratories, office rooms, corridors, or kitchens. Whereas some of these places have special geometric structures and can therefore be distinguished merely based on laser range data, other places can only be identified according to the objects found there like, for example, coffee machines in kitchens. To detect such objects, we use vision data acquired by a camera system.

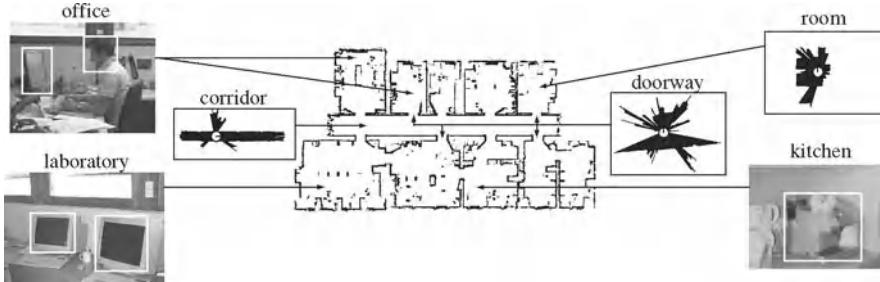


Fig. 1. An environment with offices, doorways, a corridor, a kitchen, and a laboratory. Additionally, the figure shows typical observations obtained by a mobile robot at different places.

In the approach described here, we apply the AdaBoost algorithm [7] to boost simple features, which on their own are insufficient for a reliable categorization of places, to a strong classifier for the semantic labeling of the poses of a robot in an indoor environment. Since the original version of AdaBoost provides only binary decisions, we determine the decision list with the best sequence of binary strong classifiers. We then use this semantic classifier in two main applications. Firstly, we show how to classify the different poses of a mobile robot along its trajectory by applying a hidden Markov model (HMM) which estimates the label of the current pose based on the current and the previous outputs of the semantic classifier. Secondly, we introduce a new approach to learn topological maps from occupancy grids. This is achieved by simulating the laser scans of a mobile robot at the corresponding locations and applying our semantic classification algorithm. We then apply a probabilistic relaxation algorithm to smooth the classification output, followed by a region extraction. Experimental results presented in this paper illustrate that our classification system yields recognition rates of more than 88% or 98% (depending on the number of classes to distinguish between). We also present experiments illustrating that the resulting classifier can even be used in environments from which no training data were available. This offers the opportunity to label places and to learn accurate topological maps from unknown environments.

In the past, several authors considered the problem of adding semantic information to places. Buschka and Saffiotti [4] describe a virtual sensor that is able to identify rooms from range data. Also Koenig and Simmons [11] apply a pre-programmed routine to detect doorways from range data. Althaus and Christensen [1] use line features to detect corridors and doorways. Some authors also apply learning techniques to localize the robot or to identify distinctive states in the environment. For example, Oore *et al.* [20] train a neural network to estimate the location of a mobile robot in its environment using the odometry information and ultrasound data.

Learning algorithms have additionally been used to identify objects. For example, Anguelov *et al.* [2, 3] apply the EM algorithm to cluster different types of objects from sequences of range data and to learn the state of doors. Limketkai *et al.* [16] use relational Markov networks to detect objects like doorways based on laser range data. Furthermore, they employ Markov Chain Monte Carlo to learn the parameters

of the models. Treptow *et al.* [29] utilize the AdaBoost algorithm to track a soccer ball without color information. Finally, Torralba and colleagues [28] use hidden Markov models for learning places from image data.

Compared to these approaches, our algorithm is able to combine arbitrary features extracted from different sensors to form a sequence of binary strong classifiers to label places. Our approach is also supervised, which has the advantage that the resulting labels correspond to user-defined classes.

On the other hand, different algorithms for creating topological maps have been proposed. Kuipers and Byun [14] extract distinctive points in the map. These points are defined as local maxima using a measure of distinctiveness between locations. Kortenkamp and Weymouth [12] fuse the information obtained with vision and ultrasound sensors to determine topologically relevant places. Shatkey and Kaelbling [26] apply a HMM learning approach to learn topological maps in which the nodes represent points in the plane. Thrun [27] uses the Voronoi diagram to find critical points, which minimize the clearance locally. These points are then used as nodes in a topological graph. Choset [5] encodes metric and topological information in a generalized Voronoi graph to solve the simultaneous localization and mapping problem. Additionally, Kuipers and Beeson [13] apply different learning algorithms to calculate topological maps of environments of a mobile robot.

In contrast to these previous approaches, the technique described in this paper applies a supervised learning method to identify complete regions in the map like corridors, rooms or doorways that have a direct relation with a human understanding of the environment. The knowledge about semantic labels of places is used to build accurate topological maps with a mobile robot.

The rest of the chapter is organized as follows. In Section 2, we describe the sequential AdaBoost classifier. In Section 3, we present the application of a hidden Markov model to the online place classification with a moving mobile robot. Section 4 contains our approach for topological map building. Finally, Section 5 presents experimental results obtained using our methods.

2 Semantic Place Labeling Using AdaBoost

One of the key problems to be solved is to define a classifier that allows us to categorize places in the environment according to a set of given categories. Rather than hand-coding such a classification system, our approach is to apply the AdaBoost algorithm to learn a strong classifier from a large set of simple features. In this section, we first present the AdaBoost algorithm and our approach to deal with multiple classes. We then describe the different features extracted from laser and vision data used in our current system.

2.1 The AdaBoost Algorithm

Boosting is a general method for creating an accurate strong classifier by combining a set of weak classifiers. The requirement for each weak classifier is that its accuracy is better than a random guessing. In this work we apply the boosting algorithm

AdaBoost in its generalized form presented by Schapire and Singer [25]. The input to this algorithm is a set of labeled training examples. The algorithm repeatedly selects a weak classifier $h_j(x)$ using a distribution D over the training examples. The selected weak classifier is expected to have a small classification error on the training data. The idea of the algorithm is to modify the distribution D by increasing the weights of the most difficult training examples in each round. The final strong classifier H is a weighted majority vote of the best T weak classifiers.

Throughout this work, we use the approach presented by Viola and Jones [30] in which the weak classifiers depend on single-valued features $f_j \in \mathbb{R}$. Two kinds of weak classifiers are created in our current system. In addition to the classifier defined by Viola and Jones, which has the form

$$h_j(x) = \begin{cases} +1 & \text{if } p_j f_j(x) < p_j \theta_j \\ -1 & \text{otherwise,} \end{cases} \quad (1)$$

where θ_j is a threshold and p_j is either -1 or $+1$ and thus represents the direction of the inequality, we designed a second type

$$h_j(x) = \begin{cases} p_j & \text{if } \theta_j^1 < f_j(x) < \theta_j^2 \\ -p_j & \text{otherwise,} \end{cases} \quad (2)$$

where θ_j^1 and θ_j^2 define an interval and p_j is either $+1$ or -1 indicating whether examples inside the interval are positive or negative. For both types of weak classifiers, the output is $+1$ or -1 indicating whether the classification is positive or negative. The AdaBoost algorithm determines for each weak classifier $h_j(x)$ the optimal parameters, such that the number of misclassified training examples is minimized. The final AdaBoost algorithm place categorization is shown in Algorithm 0.1.

The AdaBoost algorithm has been designed for binary classification problems. To classify places in the environment, we need the ability to handle multiple classes.

Algorithm 0.1 Generalized version of AdaBoost for place categorization.

Input: Set of N labeled examples $(x_1, y_1), \dots, (x_N, y_N)$, where $y_n = +1$ for positive examples and $y_n = -1$ for negative examples.

Initialize weights $D_1(n) = \frac{1}{2l}$ for $y_n = +1$ and $D_1(n) = \frac{1}{2m}$ for $y_n = -1$,
where l and m are the number of positive and negative examples respectively.

for $t = 1, \dots, T$ **do**

1. Normalize the weights $D_t(n)$ so that $\sum_{n=1}^N D_t(n) = 1$.
2. For each feature f_j train a weak classifier h_j using D_t .
3. For each classifier h_j calculate $r_j = \sum_n D_t(n) y_n h_j(x_n)$,
with $h_j(x_n) \in \{-1, +1\}$.
4. Choose the classifier h_j that maximizes $|r_j|$ and set $(h_t, r_t) = (h_j, r_j)$.
5. Update the weights $D_{t+1}(n) = D_t(n) \exp(-\alpha_t y_n h_t(x_n))$,
where $\alpha_t = \frac{1}{2} \ln(\frac{1+r_t}{1-r_t})$.

end for

Output: The final strong hypothesis $H(x) = \text{sign}(F(x))$, where $F(x) = \sum_{t=1}^T \alpha_t h_t(x)$.

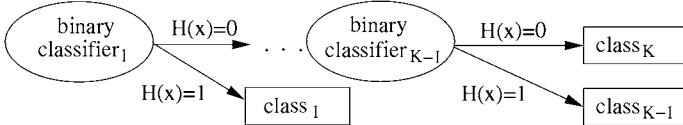


Fig. 2. A decision list classifier for K classes using binary classifiers.

To achieve this, we use a sequence of binary classifiers, where each element of such a sequence determines if an example belongs to one specific class. If the binary classifier returns a positive result, the example is assumed to be correctly classified. Otherwise, it is recursively passed to the next element in this list. Figure 2 illustrates the structure of such a decision list classifier.

In our current system, we typically consider a small number of classes which makes it feasible to evaluate all potential sequences and choose the best order of binary classifiers. Although this approach is exponential in the number of classes, the actual number of permutations considered is limited in our domain due to the small number of classes. In practice, we found out that the heuristic which sorts the classifiers in decreasing order according to their classification rate also yields good results and at the same time can be computed efficiently. Compared to the optimal order, the classifier generated by this heuristic for six different classes performed in average only 1.3% worse as shown by Rottmann *et al.* [23].

To evaluate the performance of the decision list, we compared it to the AdaBoost.M2 [7] algorithm, which is a multi-class variant of AdaBoost. In our experiments, the sequential AdaBoost classifier yields better results than the AdaBoost.M2 algorithm. A more detailed comparison between both algorithms can be found in the work by Martínez Mozos [17].

2.2 Features from Vision and Laser Data

In this section, we describe the features used to create the weak classifiers in the AdaBoost algorithm. Our robot is equipped with a 360 degree field of view laser sensor and a camera. Each laser observation consists of 360 beams. Each vision observation consists of eight images which form a panoramic view. Figure 1 shows typical laser range readings as well as fractions of panoramic images taken in an office environment. Accordingly, each training example for the AdaBoost algorithm consist of one laser observation, one vision observation, and its classification.

Our method for place classification is based on single-valued features extracted from laser and vision data. All features are invariant with respect to rotation to make the classification of a pose dependent only on the position of the robot and not on its orientation. Most of our laser features are standard geometrical features used for shape analysis [9, 24]. Typical examples considered by our system are illustrated in Figure 3. A detailed list of laser features is contained in our previous work [18]. In the system described here, we implemented several additional features which are listed in Table 1.

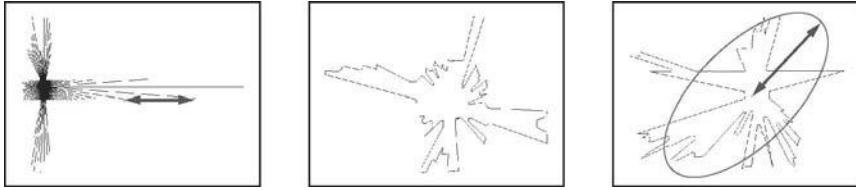


Fig. 3. Examples for features generated from laser data, namely the average distance between two consecutive beams, the perimeter of the area covered by a scan, and the mayor axis of the ellipse that approximates the polygon described by the scan.

Table 1. New Laser Features

1. Average and standard deviation of the fraction between the length of two consecutive beams.
2. Average and standard deviation of the fraction between the length of two consecutive beams divided by the maximum beam length.
3. Circularity. Let \mathbf{P} be the perimeter of the area covered by the beams and \mathbf{A} be the area covered by the beams. The circularity is defined as \mathbf{P}^2/\mathbf{A} .
4. Average and standard deviation of the distance from the centroid of \mathbf{A} to the shape boundary of \mathbf{A} , divided by the maximum distance to the shape boundary.
5. Number of gaps. Two consecutive beams form a gap if the fraction between the first and the second is smaller than a threshold.
6. Kurtosis. The kurtosis is defined as

$$\frac{\sum_{i=1}^N (\text{length}(\text{beam}_i) - \bar{l})^4}{N \cdot \sigma^4} - 3$$

where \bar{l} is the average beam length and σ the corresponding standard deviation.

In the case of vision, the selection of the features is motivated by the fact that typical objects appear with different probabilities at different places. For example, the probability of detecting a computer monitor is larger in an office than in a kitchen. For each type of object, a vision feature is defined as a function that takes as argument a panoramic vision observation and returns the number of detected objects of this type in it. This number represents the single-valued feature f_j within AdaBoost according to Eq. (1) and Eq. (2). In our case, we consider monitors, coffee machines, soap dispensers, office cupboards, frontal faces, face profiles, full human bodies, and upper human bodies. An example of such objects is shown in Figure 1. The individual objects are detected using classifiers also trained with AdaBoost and based on the set of Haar-like features proposed by Lienhart *et al.* [15].

In case the observations do not cover a 360 degree field of view, the property of the rotational invariance is lost. In such a situation, we expect that more training data will be necessary and that the classification will be less robust.

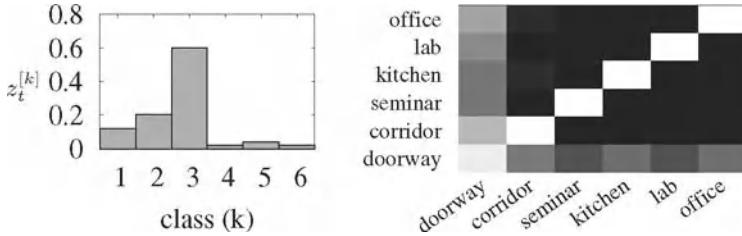


Fig. 4. The left image illustrates a classification output z . The right image depicts probabilities of possible transitions between places in the environment. To increase the visibility, we used a logarithmic scale. Dark values indicate low probability.

3 Probabilistic Classification of Trajectories

The approach described so far is able to classify single observations only but does not take into account past classifications when determining the type of place the robot is currently at. However, whenever a mobile robot moves through an environment, the semantic labels of nearby places are typically identical. Furthermore, certain transitions between classes are unlikely. For example, if the robot is currently in a kitchen then it is rather unlikely that the robot ends up in an office given it moved a short distance only. In many environments, to get from the kitchen to the office, the robot has to move through a doorway first.

To incorporate such spatial dependencies between the individual classes, we apply a hidden Markov model (HMM) and maintain a posterior $Bel(l_t)$ about the type of the place l_t the robot is currently at

$$Bel(l_t) = \alpha P(z_t | l_t) \sum_{l_{t-1}} P(l_t | l_{t-1}, u_{t-1}) Bel(l_{t-1}). \quad (3)$$

In this equation, α is a normalizing constant ensuring that the left-hand side sums up to one over all l_t . To implement this HMM, three components need to be known. First, we need to specify the observation model $P(z_t | l_t)$ which is the likelihood that the classification output is z_t given the actual class is l_t . Second, we need to specify the transition model $P(l_t | l_{t-1}, u_{t-1})$ which defines the probability that the robot moves from class l_{t-1} to class l_t by executing action u_{t-1} . Finally, we need to specify how the belief $Bel(l_0)$ is initialized.

In our current system, we choose a uniform distribution to initialize $Bel(l_0)$. Furthermore, the classification output z_t is represented by a histogram, as illustrated in the left image of Figure 4. In this histogram, the k -th bin stores the probability that the classified location belongs to the k -th class according to the sequence of classifiers in our decision list (compare Figure 2). To compute the individual values for each bin of that histogram, we use the approach by Friedman *et al.* [8]. It determines a confidence value $C \in [0, 1]$ for a positive binary classification

$$C = P(y = +1 | x) = \frac{e^{F(x)}}{e^{-F(x)} + e^{F(x)}}, \quad (4)$$

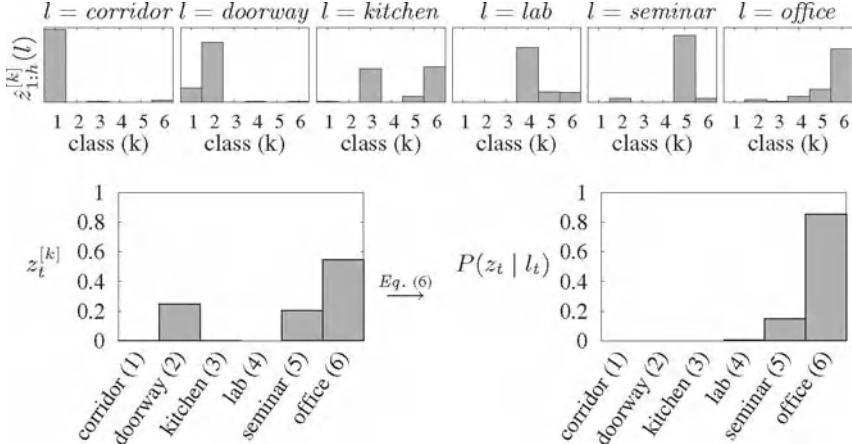


Fig. 5. The distributions depicted in the first row show the learned histograms $\hat{z}_{1:h}(l)$ for the individual classes (here corridor (1), doorway (2), kitchen (3), lab (4), seminar room (5), and office (6)). The left image in the second row depicts a possible classification output z_t . In the right image, each bar represents the corresponding likelihood $P(z_t | l_t)$ for the different estimates of l_t .

where $F(x)$ is the output of the AdaBoost algorithm according to Algorithm 0.1. Let C_k refer to the confidence value of the k -th binary classifier in our decision list. The probability that the location belongs to the k -th class is given by the k -th bin of the histogram z computed as

$$z^{[k]} = C_k \prod_{j=1}^{k-1} (1 - C_j). \quad (5)$$

Note that the confidence value C_K which is used to compute the last bin $z^{[K]}$ of the histogram holds $C_K = 1$ according to the structure of the decision list (compare Figure 2).

To determine $P(z_t | l_t)$, we use the KL-divergence [6] between two distributions. The first distribution is the current classification output z_t . The second one is learned from a statistics: for each class l , we compute a histogram $\hat{z}_{1:h}(l)$ using h observations recorded within a place belonging to class l (here $h = 50$). This histogram $\hat{z}_{1:h}(l)$ is obtained by averaging over the individual histograms $\hat{z}_1, \dots, \hat{z}_h$, which are computed according to Eq. (5). To determine $P(z_t | l_t)$, we use the KL-divergence $kld(\cdot \| \cdot)$ which provides a measure about the similarity of two distributions

$$P(z_t | l_t) = e^{-kld(z_t \| \hat{z}_{1:h}(l_t))}. \quad (6)$$

To illustrate the computation of the observation likelihood $P(z_t | l_t)$ consider Figure 5. The first row depicts examples for the histograms $\hat{z}_{1:h}(l)$. The left image in the second row depicts the output z_t of the sequential classifier while the robot was in an office. As can be seen, also the classes doorway and seminar room have a

probability significantly larger than zero. This output z_t and the histogram $\hat{z}_{1:h}(l_t)$ is than used to compute $P(z_t | l_t)$ according to Eq. (6). The result for all classes is depicted in the right image in the second row. In this image, each bin represents the likelihood $P(z_t | l_t)$ for the individual classes l_t . As can be seen, the observation likelihood given the robot is in a doorway is close to zero, whereas the likelihood given it is in an office is around 90%, which is actually the correct class.

To realize the transition model $P(l_t | l_{t-1}, u_{t-1})$, we only consider the two actions $u_{t-1} \in \{\text{Move}, \text{Stay}\}$. The transition probabilities were learned in a manually labeled environment by running 1000 simulation experiments. In each run, we started the robot at a randomly chosen point and orientation. We then executed a random movement so that the robot traveled between 20cm and 50cm. These values correspond to typical distances traveled by the robot between two consecutive updates of the HMM. The finally obtained transition probability matrix $P(l_t | l_{t-1}, u_{t-1})$ for the action *Move* is depicted in the right image of Figure 4. As can be seen, the probability of staying in a place with the same classification is higher than the probability of changing the place. Moreover, the probability of moving from a room to a doorway is higher than the probability of moving from a room directly to a corridor. This indicates that the robot typically has to cross a doorway first in order to reach a different room. Furthermore, the matrix shows a lower probability of staying in a doorway than staying at the same type of room. This is due to the fact that a doorway is usually a small area in which the robot never rests for a longer period of time.

4 Topological Map Building

The second application of our classification system is learning topological maps from occupancy grids. To take into account spatial dependencies between neighboring places, we apply a probabilistic relaxation labeling. Additionally, we describe how to perform the region extraction and the final creation of a graph representing the topological structure of the environment.

4.1 Probabilistic Relaxation Labeling

One of the key problems that need to be solved in order to learn accurate topological maps, in which the nodes correspond to the individual rooms an the environment, is to eliminate classification errors. In this section, we apply the probabilistic relaxation labeling, which has been introduced by Rosenfeld *et al.* [21], to smooth the classifications based on neighborhood relations.

Probabilistic relaxation labeling is defined as follows. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph consisting of nodes $\mathcal{V} = \{v_1, \dots, v_N\}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Let furthermore $\mathcal{L} = \{l_1, \dots, l_L\}$ be a set of labels. We assume that every node v_i stores a probability distribution about its label which is represented by a histogram P_i . Each bin $p_i(l)$ of that histogram stores the probability that the node v_i has the label l . Thus, $\sum_{l=1}^L p_i(l) = 1$.

For each node v_i , $\mathcal{N}(v_i) \subset \mathcal{V}$ denotes its neighborhood which consists of the nodes $v_j \neq v_i$ that are connected to v_i . Each neighborhood relation is represented by two values. Whereas the first one describes the compatibility between the labels of two nodes, the second one represents the influence between the two nodes. The term $\mathcal{R} = \{r_{ij}(l, l') \mid v_j \in \mathcal{N}(v_i)\}$ defines the compatibility coefficients between the label l of node v_i and the label l' of v_j . Finally, $\mathcal{C} = \{c_{ij} \mid v_j \in \mathcal{N}(v_i)\}$ is the set of weights indicating the influence of node v_j on node v_i .

Given an initial estimation for the probability distribution over labels $p_i^{(0)}(l)$ for the node v_i , the probabilistic relaxation method iteratively computes estimates $p_i^{(r)}(l)$, $r = 1, 2, \dots$, based on the initial probabilities $p_i^{(0)}(l)$, the compatibility coefficients \mathcal{R} , and the weights \mathcal{C} in the form

$$p_i^{(r+1)}(l) = \frac{p_i^{(r)}(l) \left[1 + q_i^{(r)}(l) \right]}{\sum_{l'=1}^L p_i^{(r)}(l') \left[1 + q_i^{(r)}(l') \right]}, \quad (7)$$

where

$$q_i^{(r)}(l) = \sum_{j=1}^M c_{ij} \left[\sum_{l'=1}^L r_{ij}(l, l') p_j(l') \right]. \quad (8)$$

Note that the compatibility coefficients $r_{ij}(l, l') \in [-1, 1]$ do not need to be symmetric. A value $r_{ij}(l, l')$ close to -1 indicates that label l' is unlikely at node v_j when label l occurs at node v_i , whereas values close to 1 indicate the opposite. A value of exactly -1 indicates that the relation is not possible and a value of exactly 1 means that the relation always occurs.

Probabilistic relaxation provides a framework for smoothing but does not specify how the compatibility coefficients are computed. In this work, we apply the coefficients as defined by Yamamoto [31]

$$r_{ij}(l, l') = \begin{cases} \frac{1}{1-p_i(l)} \left(1 - \frac{p_i(l)}{p_{ij}(l|l')} \right) & \text{if } p_i(l) < p_{ij}(l \mid l') \\ \frac{p_{ij}(l|l')}{p_i(l)} - 1 & \text{otherwise,} \end{cases} \quad (9)$$

where $p_{ij}(l \mid l')$ is the conditional probability that node v_i has label l given that node $v_j \in \mathcal{N}(v_i)$ has label l' .

So far we described the general method for relaxation labeling. It remains to describe how we apply this method for spatial smoothing of the classifications obtained by our AdaBoost classifier. To learn a topological map, we assume a given two-dimensional occupancy grid map [19] in which each cell $m_{(x,y)}$ stores the probability that it is occupied. We furthermore consider the eight-connected graph induced by such a grid. Let $v_i = v_{(x,y)}$ be a node corresponding to a cell $m_{(x,y)}$ from the map. Then, this node is connected to all immediate neighbors of that cell

$$\mathcal{N}_8(v_{(x,y)}) = \{ v_{(x-1,y-1)}, v_{(x-1,y)}, v_{(x-1,y+1)}, v_{(x,y-1)}, \\ v_{(x,y+1)}, v_{(x+1,y-1)}, v_{(x+1,y)}, v_{(x+1,y+1)} \}. \quad (10)$$

For the initial probabilities $p_{(x,y)}^{(0)}(l)$, we use the output of the classifier described in Section 2.1. Our set of labels \mathcal{L} is composed of the labels *corridor*, *doorway*, *room*, and *wall*. For each node $v_{(x,y)}$ in the free space of the occupancy grid map, we calculate the expected laser scan by ray-casting in the map. We then classify the observation and obtain a probability distribution z over all the possible places according to Equation (5). The classification output z for each pose (x, y) is used to initialize the probability distribution $P_{(x,y)}^{(0)}$ of node $v_{(x,y)}$.

For the nodes lying in the free space, the probability $p_{(x,y)}^{(0)}(\text{wall})$ of being a wall is initialized with 0. Accordingly, the nodes corresponding to occupied cells in the map are initialized with $p_{(x,y)}^{(0)}(\text{wall}) = 1$.

Each of the weights $c_{ij} \in \mathcal{C}$ is initialized with the value $\frac{1}{8}$, indicating that all the eight neighbors v_j of node v_i are equally important. The compatibility coefficients are calculated using Equation (9). The values $p_i(l)$ and $p_{ij}(l | l')$ are obtained from statistics in the given (occupancy grid) map corresponding to the training data as will be described in Section 5.

4.2 Region Extraction and Topological Mapping

We define a region λ_l on an adjacency graph \mathcal{A} as a set of eight-connected nodes with the same label l . For example, the region λ_{room} represents a room in the corresponding occupancy grid map. If there is a different region with the label *room*, this will represent a different room in the map. For each label $l \in \{\text{corridor}, \text{room}, \text{doorway}\}$, regions are extracted from the adjacency graph using the algorithm by Rosenfeld and Pfaltz [22]. In an analog way, we extract the connections between regions.

Finally, a topological graph $\mathcal{T} = (\mathcal{V}_{\mathcal{T}}, \mathcal{E}_{\mathcal{T}})$ is constructed so that each node $v_i \in \mathcal{V}_{\mathcal{T}}$ represents a region and each edge $e_s \in \mathcal{E}_{\mathcal{T}}$ represents a connection. The topological graph forms the resulting topological map. We finally apply a heuristic region correction step to the topological map to increase the classification rate:

1. We mark each region corresponding to a room or a corridor whose size does not exceed a given threshold of 1m^2 compared to the training set as classification error and assign the label of one of its connected regions to it.
2. We mark each region labeled as doorway whose size does not exceed a given threshold of 0.1m^2 square meters or that is connected to only one region as false classification and assign the label of one of its connected regions to it.

5 Experiments

The approach described above has been implemented and tested on real robots as well as in simulation. The robots used to carry out the experiments were an Activ-Media Pioneer 2-DX8 equipped with two SICK laser range finders as well as an iRobot B21r robot which is additionally equipped with a camera system.

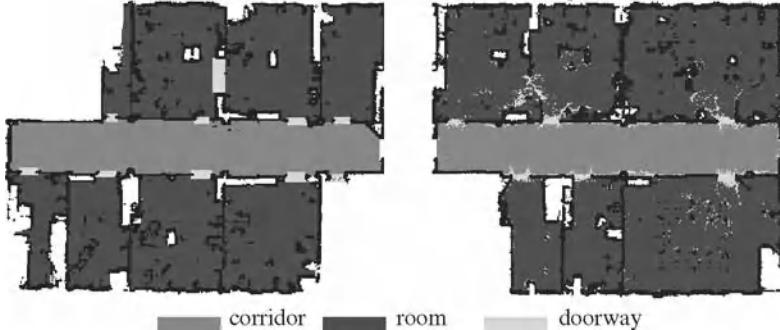


Fig. 6. Whereas the left image depicts the training data, the right image shows the classification result on the test set. The training and test data were obtained by simulating laser range scans in the map.

The goal of the experiments is to demonstrate that our simple features can be boosted to a robust classifier of places. Additionally, we analyze whether the resulting classifier can be used to classify places in environments for which no training data was available. Furthermore, we demonstrate the advantages of utilizing the vision information to distinguish between different rooms like, e.g., kitchens, offices, or seminar rooms. Additionally, we illustrate the advantages of the HMM filtering for classifying places with a moving mobile robot. Throughout these experiments, the term classification result refers to the most likely class reported by the HMM or respectively by the sequence of binary classifiers. Furthermore, we present results applying our method for semantic topological maps. We first show the results for a typical office environment. Then, we present an experiment illustrating that our approach is able to construct a topological map of a completely new environment.

5.1 Results with the Sequential Classifier Using Laser Data

The first experiment was performed using simulated data from our office environment in building 79 at the University of Freiburg. The task was to distinguish between three different types of places, namely rooms, doorways, and a corridor based on laser range data only. In this experiment, we solely applied the sequential classifier without the HMM filtering. For the sake of clarity, we separated the test from the training data by dividing the overall environment into two areas. Whereas the left part of the map contains the training examples, the right part includes only test data (see Figure 6). The optimal decision list for this classification problem, in which the robot had to distinguish between three classes, is room-doorway. This decision list correctly classifies 93.9% of all test examples (see right image of Figure 6). For alternative training and test sets we obtained similar success rates. The worst configurations of the decision list are those in which the doorway classifier is in the first place. This is probably due to the fact, that doorways are hard to detect because typically most parts of a range scan obtained in a doorway cover the adjacent room and

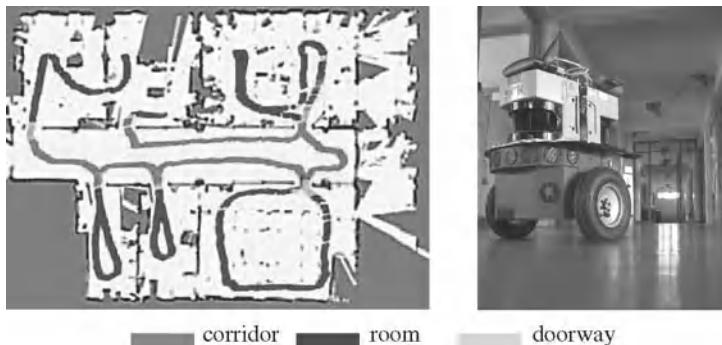


Fig. 7. The left image depicts a trajectory of a robot and the corresponding classifications based on real laser data. The robot used in this experiment is depicted in the right image.

the corridor. The high error in the first element of the decision list then leads to a high overall classification error.

The next experiment has been carried out with a real mobile robot that we manually steered through the environment. We used the same classifier as in the previous experiment. The trajectory including the corresponding classification results as well as the mobile robot are depicted in Figure 7. As can be seen from this figure, the learned classifier yields a robust labeling also for real robot data.

Additionally, we performed an experiment using a map of the entrance hall at the University of Freiburg which contained four different classes, namely rooms, corridors, doorways, and hallways. The optimal decision list is corridor-hallway-doorway with a success rate of 89.5%.

5.2 Transferring the Classifiers to New Environments

The second experiment is designed to analyze whether a classifier learned in a particular environment can be used to successfully classify the places of a new environment. To carry out this experiment, we trained our sequential classifier in the left half of the map shown in Figure 1. In the right half of this environment, our approach was able to correctly classify 97% of all places. The resulting classifier was then evaluated on scans simulated given the map of the Intel Research Lab in Seattle depicted in Figure 8. Although the classification rate decreased to 86.0%, the result indicates that our algorithm yields good generalizations which can also be applied to correctly label places of so far unknown environments. Note that a success rate of 86.0% is quite high for this environment, since even humans typically cannot consistently classify the different places.

5.3 Classification of Trajectories Using HMM Filtering

The third experiment was performed using real laser and vision data obtained in an office environment, which contains six different types of places, namely offices,

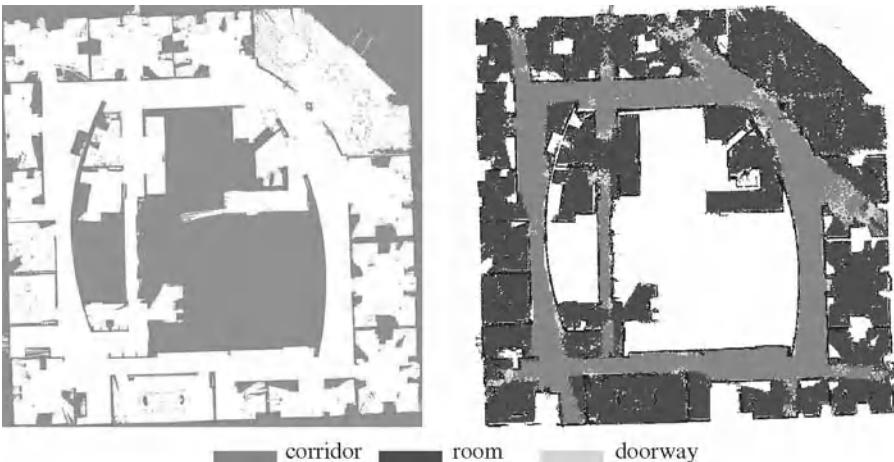


Fig. 8. The left map depicts the occupancy grid map of the Intel Research Lab and the right image depicts the classification results obtained by applying the classifier learned from the environment depicted in Figure 1 to this environment. The fact that 86.0% of all places could be correctly classified illustrates that the resulting classifiers can be applied to so far unknown environments.

doorways, a laboratory, a kitchen, a seminar room, and a corridor. The true classification of the different places in this environments is shown in Figure 9.

The classification performance of the classifier along a sample trajectory taken by a real robot is shown in left image of Figure 10. The classification rate in this experiment is 82.8%. If we additionally apply the HMM for temporal filtering, the classification rate increases up to 87.9%. The labeling obtained with the HMM is shown in the right image of Figure 10.

A further experiment was carried out using test data obtained in a different part of the same building. We applied the same classifier as in the previous experiment. Whereas the sequential classifier yields a classification rate of 86.0%, the combination with the HMM generated the correct answer in 94.7% of all cases. A two-sample t-test applied to the classification results obtained along the trajectories for both experiments showed that the improvements introduced by the HMM are significant on the $\alpha = 0.05$ level. Furthermore, we classified the same data based solely on the laser features and ignoring the vision information. In this case, only 67.7% could be classified correctly without the HMM. The application of the HMM increases the classification performance to 71.7%. These three experiments illustrate that the HMM seriously improves the overall rate of correctly classified places. Moreover, the third experiment shows that only the laser information is not sufficient to distinguish robustly between places with similar structure (see *office* and *kitchen* in Figure 10).

Finally we studied how the HMM improves the final classification rate according to the output of AdaBoost. For this purpose, we analyzed the improvement of the HMM using different classification rates from AdaBoost. This is achieved by

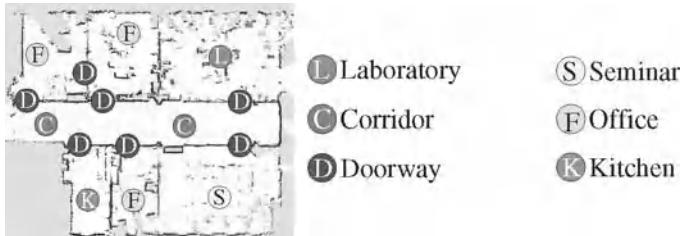


Fig. 9. Ground truth labeling of the individual areas in the environment.



Fig. 10. The left image depicts a typical classification result for a test set obtained using only the output of the sequence of classifiers. The right image shows the resulting classification in case a HMM is additionally applied to filter the output of the sequential classifier.

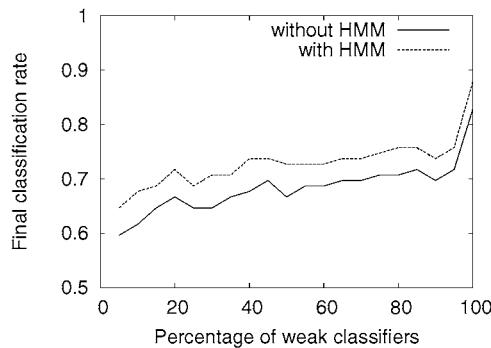


Fig. 11. Improvement of the HMM according to the percentage of weak classifiers used in each of the binary AdaBoost classifiers.

increasing the percentage of weak classifiers used in each binary classifier of the AdaBoost decision list. Here, 100% corresponds to the number of weak classifiers used in the previous experiments (Figure 10). For example, the classification rate decreases to 60% if only 5% of the weak classifiers are used. The results are shown in Figure 11. In average, the HMM improves the classification rate by 5.0%.

5.4 Building Topological Maps

The next experiment is designed to analyze our approach to building topological maps. It was carried out in the office environment depicted in the motivating example shown in Figure 1. The length of the corridor in this environment is approx. 20 m. After applying the sequential AdaBoost classifier (see Figure 12(a)), the classification of the test set was 97%. Then, we applied the probabilistic relaxation method for 50 iterations. As can be seen from Figure 12(b), this method generates more compact regions and eliminates noise. Finally, the topological map is created using the connections between regions. Some regions detected as doorways do not correspond to real doorways and are marked with circles. After applying the steps described in Section 4.2 on the corresponding topological map, these false doorways are eliminated. The final result gives a classification rate of 98.7% for all data points. The different steps of the process are illustrated as colors/grey levels in Figure 12. The doorway between the two right-most rooms under the corridor is correctly detected (Figure 12(c)). Therefore, the rooms are labeled as two different regions in the final topological map.

5.5 Topological Maps of New and Unknown Indoor Environments

This experiment is designed to analyze whether our approach can be used to create a topological map of a new unseen environment. To carry out the experiment we trained a sequential AdaBoost classifier using the training examples of the maps shown in Figure 6 and Figure 12 with different scales. In this case only the classes *room* and *corridor* were used in the training process. The resulting classifier was then evaluated on scans simulated in the map denoted as “SDR site B” in Radish [10]. This map represents an empty building in Virginia, USA. The corridor is approx. 26 meters long. The whole process for obtaining the topological map is depicted in Figure 13. The Adaboost classifier gives a first classification of 92.4%. As can be seen in Figure 13(d), rooms number 11 and 30 are originally part of the corridor, and thus falsely classified. Moreover, the corridor is detected as only one region, although humans potentially would prefer to separate it into six different corridors: four horizontal and two vertical ones. In the final topological map, 96.9% of the data points are correctly classified.

We also analyzed the results obtained without applying the relaxation process. Not using relaxation had several effects. Firstly, omitting the relaxation procedure reduces the classification rate. Secondly, the finally obtained regions are typically more sparse and do not represent the original ones as well as with relaxation. Finally, omitting the relaxation procedure increases the number of errors in the resulting topological map. For example, the map of the SDR building contained four incorrect nodes without relaxation, whereas there were only two incorrect nodes when we used the probabilistic relaxation.

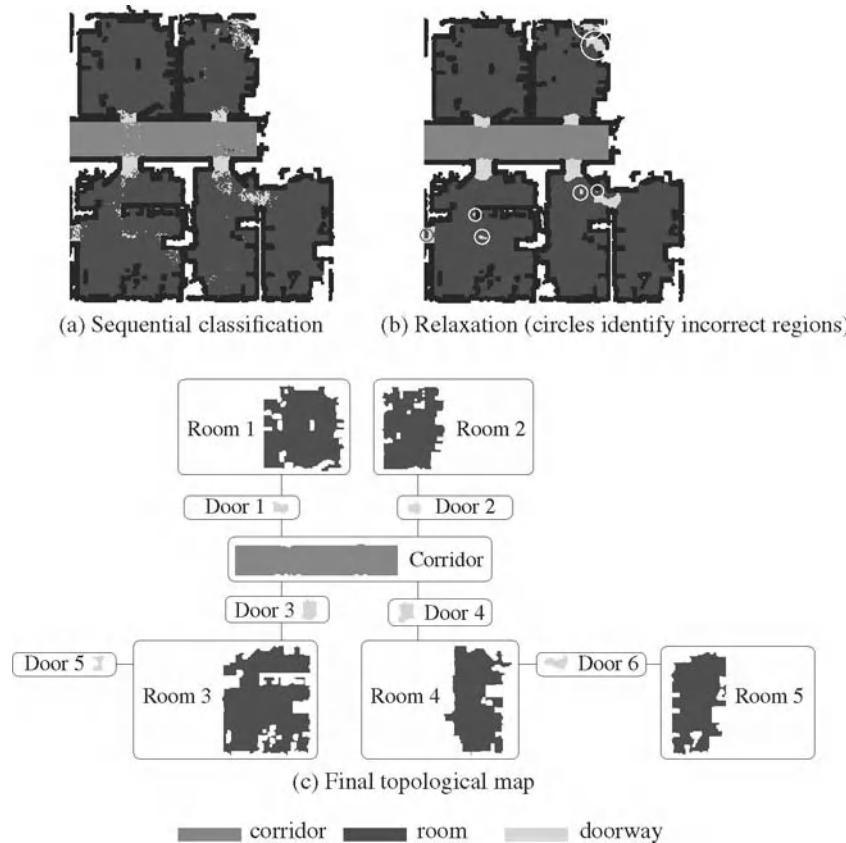


Fig. 12. This figure shows in image (a) the result of applying the sequential AdaBoost with a classification rate of 97%. (b) the result after applying relaxation including some incorrectly labeled regions (marked with circles), and finally in image (c) the final topological map with the corresponding regions.

6 Conclusion

In this paper, we presented a novel approach to classify different places in the environment of a mobile robot into semantic classes, like rooms, hallways, corridors, offices, kitchens, or doorways. Our algorithm uses simple geometric features extracted from a single laser range scan and information extracted from camera data and applies the AdaBoost algorithm to form a binary strong classifier. To distinguish between more than two classes, we use a sequence of strong binary classifiers arranged in a decision list.

We presented two applications of our approach. Firstly, we perform an online classification of the positions along the trajectories of a mobile robot by filtering the classification output using a hidden Markov model. Secondly, we present a new ap-

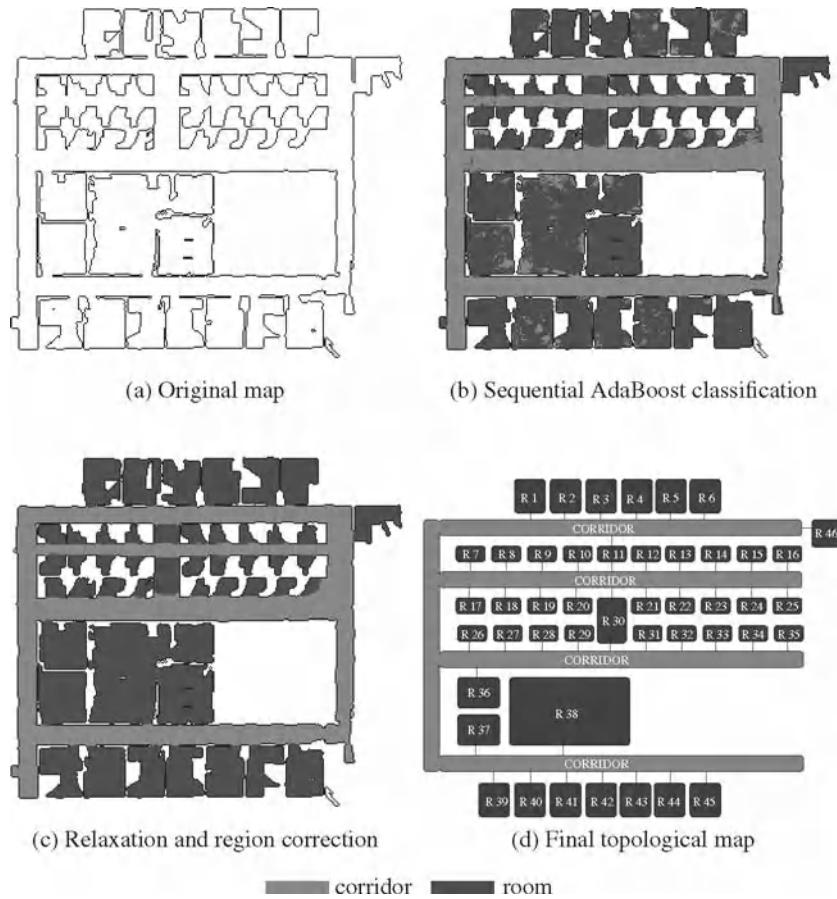


Fig. 13. This figure shows (a) the original map of the building, (b) the results of applying the sequential AdaBoost classifier with a classification rate of 92.4%, (c) the resulting classification after the relaxation and region correction, and (d) the final topological map with semantic information. The regions are omitted in each node. The rooms are numbered left to right and top to bottom with respect to the map in (a). For the sake of clarity, the corridor-node is drawn maintaining part of its region structure.

proach to create topological graphs from occupancy grids by applying a probabilistic relaxation labeling to take into account dependencies between neighboring places to improve the classifications.

Experiments carried out using real robots as well as in simulation illustrate that our technique is well-suited to reliably label places in different environments. It allows us to robustly separate different semantic regions and in this way it is able to learn topologies of indoor environments. Further experiments illustrate that a learned classifier can even be applied to so far unknown environments.

Acknowledgment

This work has partly been supported by the German Research Foundation (DFG) under contract number SFB/TR-8 (A3) as well as under contract number GRK 1103/1 and by the EC under contract number FP6-004250-CoSy. Furthermore, we would like to thank Andrew Howard for providing the map of the SDR building.

References

1. P. Althaus and H.I. Christensen. Behaviour coordination in structured environments. *Advanced Robotics*, 17(7):657–674, 2003.
2. D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, and S. Thrun. Learning hierarchical object maps of non-stationary environments with mobile robots. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2002.
3. D. Anguelov, D. Koller, Parker E., and S. Thrun. Detecting and modeling doors with mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2004.
4. P. Buschka and A. Saffotti. A virtual sensor for room detection. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 637–642, 2002.
5. H. Choset. Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 2001.
6. T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & sons, 1991.
7. Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
8. J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 2000.
9. R.C. Gonzalez and P. Wintz. *Digital Image Processing*. Addison-Wesley Publishing Inc., 1987.
10. A. Howard and N. Roy. Radish: The robotics data set repository.
11. S. Koenig and R. Simmons. Xavier: A robot navigation architecture based on partially observable markov decision process models. In D. Kortenkamp, R. Bonasso, and R. Murphy, editors, *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, pages 91–122. MIT-Press, 1998.
12. D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proc. of the Twelfth National Conference on Artificial Intelligence*, pages 979–984, 1994.
13. B. Kuipers and P. Beeson. Bootstrap learning for place recognition. In *Proc. of the Nat. Conf. on Artificial Intelligence (AAAI)*, 2002.
14. B. Kuipers and Y.T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8 1981.
15. R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *DAGM, 25th Pattern Recognition Symposium*, 2003.
16. B. Limketkai, L. Liao, and D. Fox. Relational object maps for mobile robots. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1471–1476, Edinburgh, Scotland, 2005.

17. O. Martínez Mozos. Supervised learning of places from range data using adaboost. Master's thesis, University of Freiburg, Department of Computer Science, 2004.
18. O. Martínez Mozos, C. Stachniss, and W. Burgard. Supervised learning of places from range data using adaboost. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1742–1747, Barcelona, Spain, 2005.
19. H.P. Moravec and A.E. Elfes. High resolution maps from wide angle sonar. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1985.
20. S. Oore, G.E. Hinton, and G. Dudek. A mobile robot that learns its place. *Neural Computation*, 9(3):683–699, 1997.
21. A. Rosenfel, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Systems. Man. Cybernet.*, 6(6):420–433, 1976.
22. A. Rosenfeld and J. L. Pfaltz. Sequential operations in digital picture processing. *Journal of the Association for Computing Machinery*, 13(4):471–494, 1966.
23. A. Rottmann. Bild- und laserbasierte klassifikation von umgebungen mit mobilen robotern. Master's thesis, University of Freiburg, Department of Computer Science, 2005. In German.
24. J.C. Russ. *The Image Processing Handbook*. CRC Press, 1992.
25. R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.*, 37(3):297–336, 1999.
26. H. Shatkey and L.P. Kaelbling. Learning topological maps with weak local odometric information. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 1997.
27. S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
28. A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. In *Proc. of the Int. Conf. on Computer Vision (ICCV)*, 2003.
29. A. Treptow, A. Masselli, and A. Zell. Real-time object tracking for soccer-robots without color information. In *Proc. of the Europ. Conf. on Mobile Robots (ECMR)*, 2003.
30. P. Viola and M.J. Jones. Robust real-time object detection. In *Proc. of IEEE Workshop on Statistical and Theories of Computer Vision*, 2001.
31. H. Yamamoto. A method of deriving compatibility coefficients for relaxation operators. *Compt. Graph. Image Processing*, 10:256–271, 1979.

Emergence, Exploration and Learning of Embodied Behavior

Yasuo Kuniyoshi¹, Shinsuke Suzuki¹, and Shinji Sangawa¹

Laboratory for Intelligent Systems and Informatics, Department of Mechano-Informatics, School of Information Science and Technology, The University of Tokyo, Tokyo, Japan kuniyosh@isi.imi.i.u-tokyo.ac.jp

1 Introduction

The real world is full of unexpected changes, contingencies and opportunities. Thus it is virtually impossible to perfectly specify in advance all the conditions, states and outcomes for all the possible actions. The so-called “frame problem” was originally discovered with symbolic reasoning agents [6], but essentially it affects any “intelligent” system that relies on explicit descriptions about the states and actions. For example, in control theory terms, the target system can abruptly deviate from the assumed model of the system dynamics, making the pre-defined control law invalid.

The above problem shows up in a wide range of robot behavior, particularly when the situation is complex and fluid. Cognitive interactive tasks such as recognizing another agent’s behavior and imitating the task or generating helpful/competing responses often involves high unpredictability due to the caprice and complexity of human behavior and the mutual dependency between the agents’ behavior. Even at the level of physical motion control, the situation can be highly complex and unpredictable with a complex body such as a humanoid and the characteristics of the dynamics such as non-linearity, under-actuation, contact states, and rough terrain.

Various adaptive methods have been developed in the past with successful robotic experiments. However, they are either too slow to converge or too narrow in terms of the adaptation range. For example, the most popular learning methods such as reinforcement learning and genetic algorithm both require vast number of trials to converge, and when the bodily or environmental condition changes, they need thousands of trials again to adapt. Moreover, these methods require careful design of the state representation which is not always straightforward unless the characteristics of the body and the environment is well understood.

This paper proposes a novel alternative method for motor behavior emergence. Our model assumes no predefined motion primitives nor state representation. It discovers and exploits the natural dynamics of body-environment

interaction. It adapts to the dynamic change of the bodily or environmental structures very quickly, in a *few seconds*. It has biological correlates such as spine/medulla circuit and general movements (GM) that play an important role in very early motor development of human babies.

In the following sections, we first review the issue of exploiting natural body-environment dynamics. Then we present our model which facilitates the emergence of behavior exploiting such dynamics, with some experimental results. In the final part we present our ongoing effort on simulating early human motor development based on the model.

2 Exploiting Embodied Dynamical Structures

In dynamic motion control, exploiting the property of natural body-environment dynamics is very important in order to achieve robustness and efficiency. A well-known example of a meaningful behavior based on pure body-environment dynamics is the passive dynamic walker [5]. And one way to successfully exploit and extend the natural dynamics is to combine it with neural oscillators [10]. These and related issues are gaining more and more interests with quickly accumulating knowledge.

Another related example is juggling. It is also a rhythmic and cyclic motion but somewhat simpler than biped walking. Its dynamics is well understood and effective control methods are proposed [8, 1].

Recently, we presented an example of exploiting acyclic dynamics of whole-body humanoid motion called "roll-and-rise" [3]. Our adult-size humanoid robot first lies flat on the floor, then swings up and down both of the legs, rolling on the back and achieving a crouching posture very quickly. The task requires exploitation and switching of multiple body-environment dynamics with different constraints.

The above and other related examples show that very simple controllers can realize very robust and efficient motion if they properly exploit the natural body-environment dynamics. An outstanding question is how to automatically discover the dynamics and exploit it. This can be a very difficult problem if we assume a body with many degrees of freedom and changing constraints.

3 Emergent Coordination of Multiple Degrees of Freedom

We propose a novel model in which a distributed set of chaotic elements are coupled with the multi-element musculo-skeletal system. Consistent motor behavior patterns emerge from embodied interactions. The same principle gives rise to immediate adaptation capability to changing constraints and switching to different/novel motion patterns. It requires no training or evaluation function. The system autonomously explores, discovers, and exploits possible motion patterns.

3.1 Coupled Chaotic System

Coupled Map Lattice(CML) and Globally Coupled Map(GCM)[2] have been investigated in complex systems science for their rich dynamics properties. They follow (1)-(2). CML is a coupled chaotic system with local interaction (1). GCM is one with global interaction (2).

$$x_{n+1}^i = (1 - \varepsilon) f(x_n^i) + \frac{\varepsilon}{2} \{ f(x_n^{i+1}) + f(x_n^{i-1}) \} \quad (1)$$

$$x_{n+1}^i = (1 - \varepsilon) f(x_n^i) + \frac{\varepsilon}{N} \sum_{j=1}^N f(x_n^j) \quad (2)$$

Where, x_n^i denotes the internal state of i th element at time n , N the total number of elements, and ε the connection weight between elements. $f(x)$ can be any chaos function. In this paper, we adopt a standard *logistic map* represented as the following.

$$f(x) = 1 - ax^2 \quad (3)$$

With no interaction between the elements, all of them behave chaotically. But with interaction, depending on the parameters (a, ε) , a rich variety of dynamical structures emerge such as ordered phases (with clusters of resonating elements) and partially ordered phases (configuration of the clusters changes with time).

This phenomenon is essentially caused by a competition of two tendencies; (1) A tendency to synchronize each other by the effect of the mean-field, and (2) a tendency to take arbitrarily different values due to the nature of chaos dynamics.

3.2 Body and Environment as an Interaction Field of Chaotic Elements

Figure 1 shows our model of chaos coupling through robotic embodiment.

N chaotic elements are connected with actuators and sensors of the robot body. Each element drives a corresponding actuator based on its current internal state. The effect of N actuators collectively change the physical state of the body which is constrained by and interacting with the environment. In other words, the output of N chaotic elements are mixed together and transformed by the embodied dynamics. The result is then sensed at each site of the actuator, e.g. in terms of joint angle or muscle length. Each sensor value is then input to the corresponding chaotic element. Then each element updates, by chaotic mapping, its internal state from the new sensor value and the previous internal state.

The important points of our model are as follows :

- A chaotic element connect each sensor and actuator.

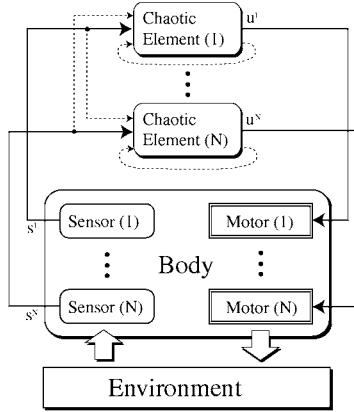


Fig. 1. Outline of our model

- Each actuator is coupled to the body via a spring, simulating a muscle. Each sensor measures the deformation of the spring. Thus, the actuators collectively affect the dynamic state of the body. And the sensors get the mixed effects of the corresponding actuator and the global state of the body.
- The body and the environment interacts. Together, they serve as the interaction field of the chaotic elements.

In our model, body-environment interaction dynamics, or *embodiment*, serves as the chaos coupling field, which is non-linear and time-varying. Theoretically very little is known about such cases, but since the coupling field directly reflects the current body-environment dynamics, we believe that the emergent ordered patterns correspond to useful motor coordination patterns which immediately get reorganized in response to dynamically changing environmental situation.

We devised 3 types of formula to update the internal state of an element: (4), (5), and (6). Where, u denotes the internal state, s the sensor value, and \bar{s} the mean of sensor values. The 2nd and the 3rd terms in f of (4) and (6) are intended to be GCM-like connection and CML-like connection. $\varepsilon_1, \varepsilon_2$ are the weight of each connection. We used logistic map (3) for $f(x)$ ¹. Initial condition of u is a random value within $(0, 1)$.

Table 1 shows the interpretation of each formula².

¹ In implementation, to avoid divergence, x is constrained as follows : $if(x > 1) x = 1$, $if(x < -1) x = -1$

² In order to understand the “adjustment” effect, the GCM/CML equations should be transformed by applying f on both sides and re-arranged to match (4)-(6)

$$\text{Type-A} : u_n^i = f \left\{ u_{n-1}^i + \varepsilon_1 (\bar{s}_{n-1} - s_{n-1}^i) + \varepsilon_2 \left(\frac{s_{n-1}^{i+1} + s_{n-1}^{i-1}}{2} - s_{n-1}^i \right) \right\} \quad (4)$$

$$\text{Type-B} : u_n^i = f(s_n^i) \quad (5)$$

$$\text{Type-C} : u_n^i = f \left\{ s_{n-1}^i + \varepsilon_1 (\bar{s}_{n-1} - s_{n-1}^i) + \varepsilon_2 \left(\frac{s_{n-1}^{i+1} + s_{n-1}^{i-1}}{2} - s_{n-1}^i \right) \right\} \quad (6)$$

Table 1. Interpretation of the update rules of the coupled chaotic systems

GCM	Each element follows its own pure chaos dynamics with some adjustment to approach the global mean value of all the other pure chaos elements.
CML	Each element follows its own pure chaos dynamics with some adjustment to approach the local mean value of the adjacent pure chaos elements.
Type-A	Each element follows its own pure chaos dynamics with some adjustment to reduce the difference of the corresponding sensor value from the global and the local means of other sensor values.
Type-B	Each element is updated by a chaos map of its sensor value. The sensor value contains the effects of the self and the other elements mixed together through the embodiment. The mixing function does not appear explicitly in the equation. It is a non-linear and time-varying function, reflecting the physical dynamics of the body-environment interaction.
Type-C	In addition to the Type-B, some adjustment is applied in order to reduce the deviation of the corresponding sensor value from the global and the local means of other sensor values.

4 Experiments

Our model of behavior emergence is quite simple. However, its behavior is extremely complex. Even theoretically, a behavior of coupled chaotic systems with time-varying non-linear coupling is very poorly understood. Moreover, there has been no attempt so far to exploit this phenomena for robotic behavior generation. Therefore, we carried out a series of experiments in order to investigate the following points.

1. How to design the connection between the body and the chaotic elements?
2. How does the system behave in case the structure of body dynamics changes?
3. How does the system behave in case the structure of environment changes?
4. How can we impose “goal-directedness” onto the behavior while maintaining the emergent property?

In the following, we present some results from our preliminary experiments. Further details should be found in another paper [4].

We use dynamics simulation library ODE[9] to simulate the dynamics of a robot and environment. The time step size of ODE was 0.01 and that of couple chaotic system was T_c . In implementation, u and s in section 3.2 were associated with s_{raw} and m ((7), (8), (9), (10)), where s_{raw} denotes the raw value of a sensor and m the motor output of an actuator. Note that the gains $g_u, g_{u_{out}}, g_s, g_{s_{in}}$ and the offsets $o_u, o_{u_{out}}, o_s, o_{s_{in}}$ are independent of the element index i . They are constant parameters.

$$u_{out} = g_u \cdot u + o_u \quad (7)$$

$$m = g_{u_{out}} \cdot u_{out} + o_{u_{out}} \quad (8)$$

$$s_{in} = g_{s_{in}} \cdot s_{raw} + o_{s_{in}} \quad (9)$$

$$s = g_s \cdot s_{in} + o_s \quad (10)$$

4.1 Experiments with a Muscle-Joint Model

Configuration

Firstly, we experiment with a muscle-joint model shown in Fig. 2 which consists of two cylindrical rigid bodies and 12 muscle fibers. The base link is fixed to the ground, and the upper link is connected by a ball-joint to the base link. It can be bent in any direction within the limit of 0.5 [rad]. The 12 muscle fibers are attached between the two links isotropically.

Each muscle fiber is modelled with Hill’s characteristic equation [7]. m in (8) corresponds to the activation level of a muscle fiber in this model. The sensor value s_{raw} is provided by either a “length-sensor” measuring the normalized length of the muscle fiber or a “tension sensor”³ measuring the

³ In case of tension sensor, before the process of (10), s_{in} is constrained as follows:
 $if(s_{in} > 1) s_{in} = 1, if(s_{in} < -1) s_{in} = -1$

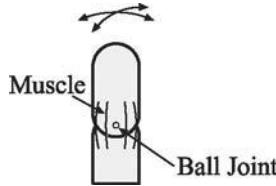


Fig. 2. Appearance of the muscle-joint model

normalized tension of the muscle fiber. In all experiments, $(g_{u_{out}}, o_{u_{out}})$ was set to $(0.5, 0.5)$ respectively. In case of tension sensor, $(g_{s_{in}}, o_{s_{in}})$ was set to $(-2.5, 3.0)$. In case of length sensor, $(g_{s_{in}}, o_{s_{in}})$ was set to $(1.0, 0.0)$.

Experiments with/without sensor feedback

Firstly, when there is no sensor feedback (Fig. 3), the motion of the joint was chaotic and no cluster structure was observed.

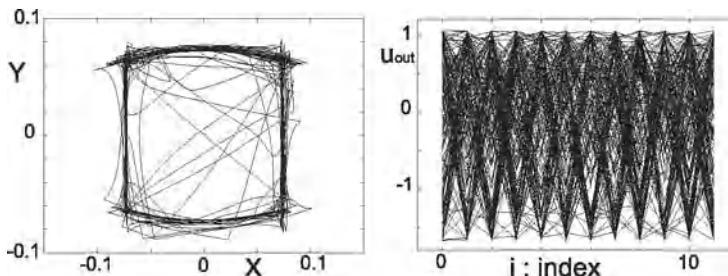


Fig. 3. Experiment with no sensor feedback. Trajectory of the center of mass of the upper link projected on $x - y$ plane (left graph). Cluster plot of the chaotic elements (right). For each element with index i , its motor output u_{out} is plotted superposedly for $n = 10, 11, 12, \dots$. The points of all the elements are connected with a line for each time step. (Type-A, $a = 1.6$, $\varepsilon_1 = 0.0$, $\varepsilon_2 = 0.0$, $T_c = 0.21$, $g_u = 1.7$, $g_s = 2.0$, $o_u = -0.65$, $o_s = -1.0$)

Secondly, in case of an experiment with tension sensor feedback, the motion was chaotic for the initial several steps. But after a time, it changed to the ordered rhythmical motion. Fig. 4 is the graph while the motion was rhythmical. Cluster structure is observed.

In case of an experiment with length sensor feedback, the motion was ordered and rhythmical from the beginning (Fig. 5). In the same experiment with a different parameter set, the motion was rhythmical in the beginning, then after a while, the direction of oscillation changed and it began another

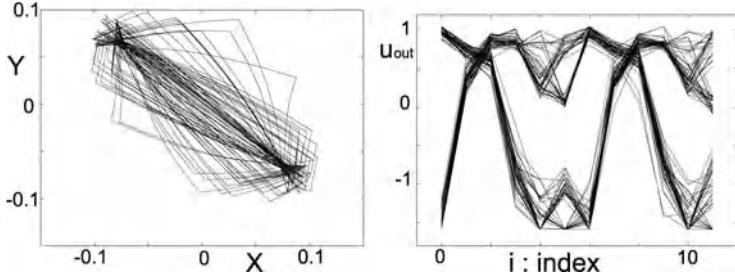


Fig. 4. Experiment with feedback of tension sensor (Type-A, tension sensor, $a = 1.55$, $\varepsilon_1 = 0.3$, $\varepsilon_2 = 0.3$, $T_c = 0.21$, $g_u = 1.7$, $g_s = 2.0$, $o_u = -0.65$, $o_s = -1.0$)

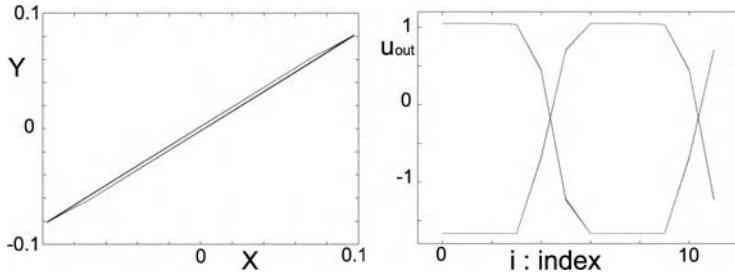


Fig. 5. Experiment with feedback of length sensor (Type-B, length sensor, $a = 1.6$, $T_c = 0.21$, $g_u = 1.7$, $g_s = 1.0$, $o_u = -0.65$, $o_s = 0.0$)

rhythymical motion (Fig. 6). The change of oscillating direction occurred aperiodically.

Experiments with a dynamic change of the environmental structure

The environment makes a part of the interaction field for the chaotic elements. In this experiment, we observed the system's behavior when the structure of the environment is dynamically changed by bringing in an obstacle disturbing the oscillation of the muscle-joint system (Fig. 7).

The obstacle was brought in at $t = 3$. Fig. 8 shows the result : beginning at the top, from $t = 0.42$ to $t = 3.15$, from $t = 3.15$ to $t = 6.93$, and from $t = 6.93$ to $t = 12.6$. A little while after colliding against the obstacle, the joint made a complex motion that it repeated colliding in a short period of time and the motor commands were chaotic. But soon after that, within about 3 seconds, it began to oscillate orderly in a new collision-free direction.

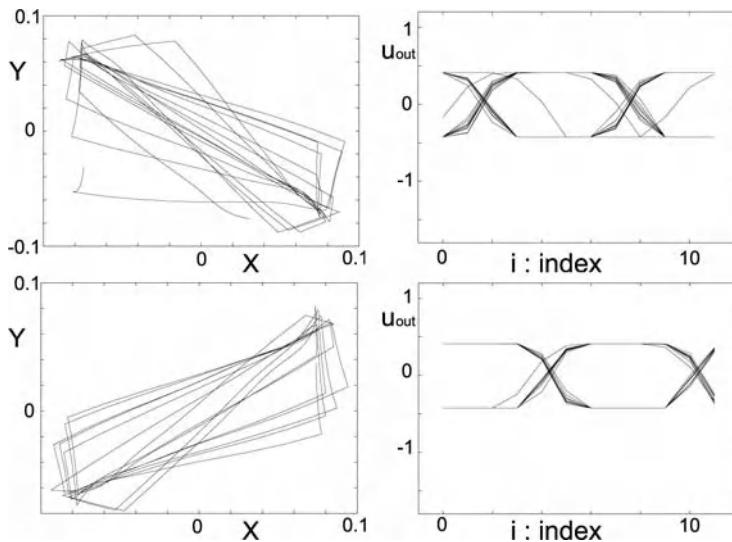


Fig. 6. Experiment when dynamic transitions could be seen. The upper graph shows the behavior before transition and the lower one shows that after transition. (Type-A, length sensor, $a = 1.6$, $T_c = 0.21$, $g_u = 0.52$, $g_s = 1.0$, $o_u = -0.107$, $o_s = 0.0$)

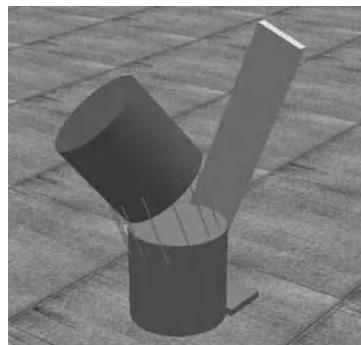


Fig. 7. The muscle joint model and obstacle

4.2 Experiments with an Insect-Like Multi-legged Robot

Configuration

In order to investigate the effects of our model in a more meaningful behavior with more complex interactions with the environment, we defined an insect-like multi-legged robot. The robot has a disc-shaped body with 12 legs attached on its fringe with regular spacing (Fig. 9). Each leg is connected to the body by a rotational joint and 2 springs whose spring constant is K . Each

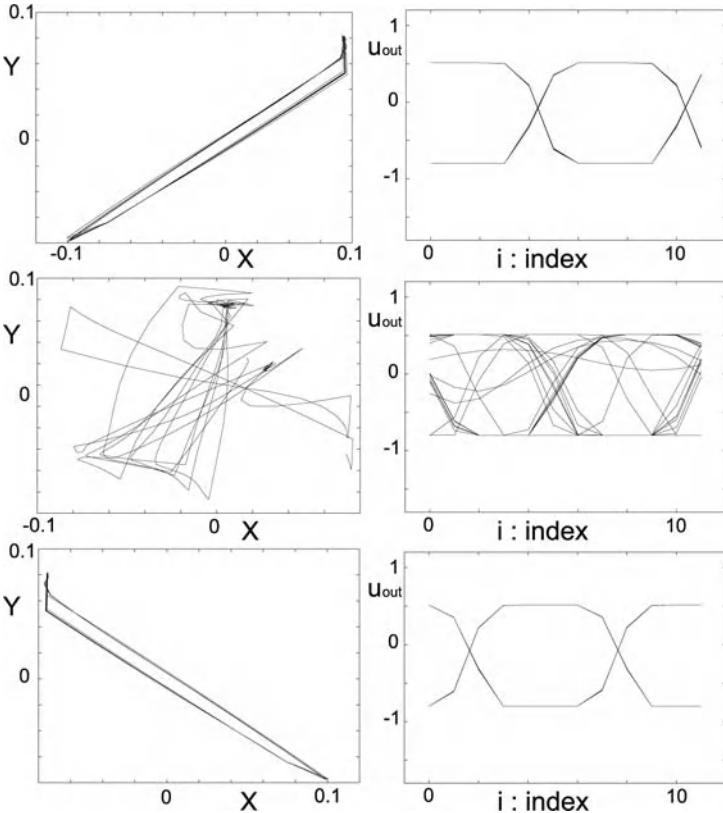


Fig. 8. Experiment with obstacle ('Type-B, length sensor, $a = 1.6$, $T_c = 0.21$, $g_u = 0.82$, $g_s = 1.0$, $o_u = -0.3$, $o_s = 0.0$)

leg can swing only in the direction shown in the middle of Fig. 9, and its joint angle is constrained to be less than $\pm\theta_{lim}$. The environment has a standard gravity and a constant friction (with the static friction coefficient μ). m in (8) corresponds to the torque τ of each joint. s_{raw} in (9) corresponds to the angle θ . Table 2 shows the parameters common to all the experiments using the above robot model.

Table 2. Parameters for the insect-like robot

T_c	K	θ_{lim}	$g_{u_{out}}$	g_s	$g_{s_{in}}$	$o_{u_{out}}$	o_s	$o_{s_{in}}$
0.17	1.0	0.8	1.0	0.5	1.25	0.0	0.5	0.0

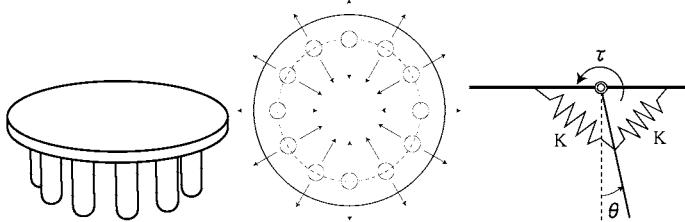


Fig. 9. Appearance of the insect-like robot (left), direction of leg motion(middle), and the mechanism of a leg (right).

Experiments with sensor feedback

With no sensor feedback, no order was observed in the motion of the robot. It just kept on randomly struggling around the same spot on the ground.

On the other hand, when the sensor feedback is introduced, after the initial chaotic period (a few seconds), the robot started to move in a certain direction, and then finally showed a stable locomotive behavior with a constant speed in a stable direction. The locomotive behavior was realized by synchronizing the 3 or 4 hind legs and kicking the ground with them. Fig. 10 is the graph while the locomotive behavior was observed.

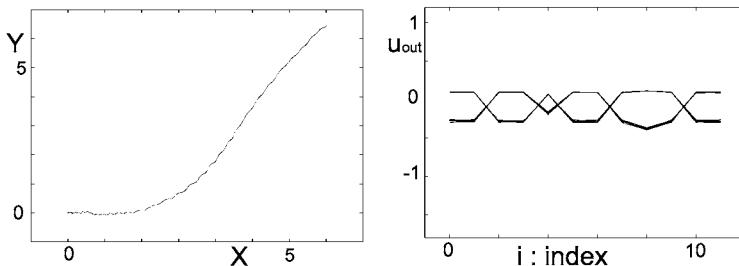


Fig. 10. Experiment with sensor feedback ($a = 1.47$, $\varepsilon_1 = 0.1$, $\varepsilon_2 = 0.1$, $g_u = 0.4$, $o_u = -0.28$, $\mu = 0.1$)

4.3 Summary

The proposed model exhibited a capability to quickly discover various motion patterns in accordance with the body-environment dynamics. It can cope with dynamically changing constraints. In other experiments [4], we confirmed that the model can adapt to changes of the muscle arrangements, the capability persists over a range of parameters, and a possibility of imposing goal-directedness on the emergent behavior.

5 Simulated Baby

The above model correlates with the essential structure of vertebrates, i.e. the spine/medulla circuit and the musculo-skeletal body. It is well established that parts of spine/medulla circuit acts as non-linear oscillators, called CPG (central pattern generator). Under certain conditions, a coupled system of non-linear oscillators act as a coupled chaotic system. Therefore, it is quite plausible that vertebrates exploit the similar principle as our model for acquisition and adaptation of motor behavior.

Since our model explores and discovers motion patterns that fit the natural property of the body, it may be a good candidate for simulating the initial mechanism of motor development. It may be able to start with very little pre-defined knowledge and autonomously acquire appropriate motor primitives.

A human body is so complex, and a systematic search for all possible motion patterns is virtually impossible. However, our model should be able to discover appropriate motions very quickly. Moreover, the cluster emergence in pure CML and GCM are known to scale to thousands of elements. This is a good reason to expect that our model can handle the musculo-skeletal system of a human body.

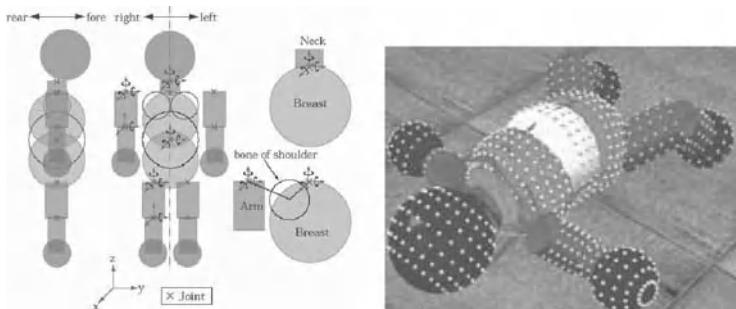


Fig. 11. Simulated baby.

Figure 11 shows the view of our simulated baby. The outlook is crude as we invest little effort on the quality of graphics. However, the musculo-skeletal system is modeled at a highly detailed level. Our model has 198 muscles. We omitted the wrist, ankle, fingers, toes, neck and face. But the body stem and the limbs are quite faithfully modeled. The dimensions, mass, and inertial parameters of all the body parts are defined according to the measurements of real babies. The proprioceptive sensing organs, i.e. the muscle spindles and Golgi tendon organs, are also modeled as precisely as possible. The muscles are also modeled to match the performance of real babies. All the physical body parameters are modeled as functions of week age after gestation (in the

uterus). So the body can simulate physical growth of the fetal and neonatal periods.

As the first step of neural modeling, we adopted the same coupled chaos model (in section 4) for the spinal circuit. In addition, we added self-organizing maps to simulate sensory and motor areas of the cerebral cortex. All the connections are continuously updated by Hebbian learning while the neural system drives the body.

The simulated baby body is placed in two types of simulated environments; The “fetus” is placed in a simulated uterus with a pushable wall, filled with liquid. The “neonate” is placed on a flat floor surrounded by flat walls (like a playpen). We are starting to observe emergence of patterned motions and stable clustering of cortical neurons.

6 Summary and Discussions

We proposed a novel framework for highly (quick) adaptive motor behavior. The core mechanism is based on coupled chaotic system, which autonomously explores and generates various coordination patterns of multiple degrees of freedom. The emergent motion patterns exploit and resonate with the body-environment dynamics. Therefore our model is a very good candidate as the initial core mechanism to simulate very early motor development of human babies. It should be important for human babies to acquire motor primitives exploiting the characteristics of body-environment dynamics.

The above model correlates with real human babies because the CPG in spine/medulla can generate high dimensional chaos under certain conditions, and the resulting whole body movement has the similar property as the general movement (GM) which appears in early motor development of human babies.

We are now constructing and experimenting with a simulated baby. It is designed to be very close to real human babies in terms of musculo-skeletal system. The coupled chaotic system model is adopted as the basic mechanism of behavior emergence. When an emergent motion pattern persists for certain time duration, the learning in the cortex model and other neural connections fixates it in the neural connections. This way the system should be able to explore, discover and learn various motor primitives which fully exploit the natural body-environment dynamics. It is still an open question how to design a mechanism that appropriately integrate the learning and emergence.

The above approach may provide a solution to avoid the frame problem, as the system does not rely on static (or very slowly adapting) internal representations, and can immediately adapt to changing situations.

Acknowledgement

This work was partially supported by Grant-in-Aid for Scientific Research from JSPS, Japan, and JST ERATO Asada Project. The authors would like to

express their thanks to Prof. Ichiro Tsuda, Prof. Rolf Pfeifer, Prof. Kazuyuki Aihara, Prof. Minoru Asada and many other researchers for the valuable discussions.

References

1. R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *Int. J. Robotics Research*, 18(6):534–555, 1999.
2. K. Kaneko and I. Tsuda. *Complex Systems: Chaos and Beyond*. Springer, 2001.
3. Y. Kuniyoshi, Y. Ohmura, K. Terada, and A. Nagakubo. Dynamic roll-and-rise motion by an adult-size humanoid robot. *Int. J. of Humanoid Robotics*, 1(3):497–516, 2004.
4. Y. Kuniyoshi and S. Suzuki. Dynamic emergence and adaptation of behavior through embodiment as coupled chaotic field. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pages 2042–2049, 2004.
5. T. McGeer. Passive dynamic walking. *Int. J. of Robotics Research*, 9(2), 1990.
6. Zenon W. Pylyshyn. *The Robot's Dilemma: The Frame Problem in Artificial Intelligence*. Ablex, 1987.
7. Nakamura R. and Saito H. *Kisoundougaku*. Ishiyakushuppan, 4 edition, 1992.
8. A. A. Rizzi and D. E. Koditschek. Further progress in robot juggling: The spatial two-juggle. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 919–924, 1993.
9. Russell Smith. Open dynamics engine(ode).
<http://opende.sourceforge.net/ode.html>.
10. G. Taga, Y. Yamaguchi, and H. Shimizu. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65:147–159, 1991.

Hierarchical Conditional Random Fields for GPS-Based Activity Recognition

Lin Liao, Dieter Fox, and Henry Kautz

University of Washington, Department of Computer Science & Engineering, Seattle, WA

Summary. Learning patterns of human behavior from sensor data is extremely important for high-level activity inference. We show how to extract a person’s activities and significant places from traces of GPS data. Our system uses hierarchically structured conditional random fields to generate a consistent model of a person’s activities and places. In contrast to existing techniques, our approach takes high-level context into account in order to detect the significant locations of a person. Our experiments show significant improvements over existing techniques. Furthermore, they indicate that our system is able to robustly estimate a person’s activities using a model that is trained from data collected by other persons.

1 Introduction

The problem of learning patterns of human behavior from sensor data arises in many applications, including intelligent environments [4], surveillance [5], human robot interaction [2], and assistive technology for the disabled [18]. A focus of recent interest is the use of data from wearable sensors, and in particular, GPS (global positioning system) location data, to learn to recognize the high-level activities in which a person is engaged over a period of many weeks, and to further determine the relationship between activities and locations that are important to the user [1, 12, 14]. The goal of this research is to segment a user’s day into everyday activities such as “working,” “visiting,” “travel,” and to recognize and label significant locations that are associated with one or more activity, such as “workplace,” “friend’s house,” “user’s bus stop.” Such activity logs can be used, for instance, for automated diaries or long-term health monitoring. Previous approaches to location-based activity recognition suffer from design decisions that limit their accuracy and flexibility.

Restricted activity models: Ashbrook and colleagues [1] only reason about moving between places, without considering different types of places or different routes between places. In the context of indoor mobile robotics, Bennewitz *et al.* [2] showed how to learn different motion paths between places. However, their approach does not model different types of places and does not estimate the user’s activities when moving between places. In our previous work [12, 19] we developed a hierarchical dynamic Bayesian network model that can reason about different transportation routines between places. In separate work, we developed an approach that can learn to

distinguish between different types of places, such as work place, home, or restaurant [14]. However, this model is limited in that it is not able to consider information about motion *between* places and about activities occurring at each point in time.

Inaccurate place detection: Virtually all previous approaches address the problem of determining a person's significant places by assuming that a geographic location is significant if and only if the user spends at least θ minutes there, for some fixed threshold θ [1, 12, 14, 2]. In practice, unfortunately, there is no threshold that leads to a satisfying detection of all significant locations. For instance, locations such as the place where the user drops off his children at school may be visited only briefly, and so would be excluded when using a high threshold θ . A low threshold, on the other hand, would include too many insignificant locations, for example, a place where the user waited at a traffic light. Such detection errors can only be resolved by taking additional context information into account, such as the user's current activity.

In this paper we present a novel, unified approach to automated activity and place labeling which overcomes these limitations. Key features of our system are:

- It achieves *high accuracy in detecting significant places* by taking a user's context into account when determining which places are significant. This is done by simultaneously estimating a person's activities over time, identifying places that correspond to significant activities, and labeling these places by their type. This estimation is performed in a unified, conditionally trained graphical model (conditional random field). As a result, our approach does not rely on arbitrary thresholds regarding the time spent at a location or on a pre-specified number of significant places.
- It creates a *rich interpretation of a user's data*, including transportation activities as well as activities performed at particular places. It allows different kinds of activities to be performed at the same location, and vice-versa.
- This complex estimation task requires *efficient, approximate inference and learning algorithms*. Our system performs inference using loopy belief propagation, and parameter learning is done using pseudo-likelihood. In order to efficiently reason about aggregations, such as how many different places are labeled as a person's home, we apply Fast Fourier Transforms to compute aggregation messages within belief propagation.

This paper is organized as follows. We begin with a discussion of conditional random fields (CRFs) and how to apply them to the problem of location-based activity recognition. Then, we explain how to perform efficient inference and parameter learning in CRFs. Finally, we present experimental results on real-world data that demonstrate significant improvement in coverage and accuracy over previous work.

2 Hierarchical Activity Model

The basic concept underlying our activity model is shown in Figure 1. Each circle indicates an object such as a GPS reading, a location in the map, or a significant place. The edges illustrate probabilistic dependencies between these objects.

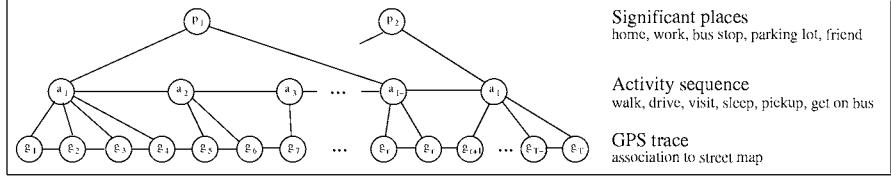


Fig. 1. The concept hierarchy for location-based activity recognition. For each day of data collection, the lowest level typically consists of several thousand GPS measurements.

GPS readings are the input to our model — a typical trace consists of approximately one GPS reading per second; each reading is a point in 2D space. We segment a GPS trace in order to generate a discrete sequence of activity nodes at the next level of the model. This segmentation is done *spatially*, that is, each activity node represents a set of consecutive GPS readings that are within a certain area. If a street map is available, then we perform the segmentation by associating the GPS readings to a discretized version of the streets in the map (in our experiments we used 10m for discretization). This spatial segmentation is very compact and convenient for estimating high-level activities. For instance, our model represents a 12 hour stay at a location by a single node. Our model can also reason explicitly about the duration of a stay, for which dynamic models such as standard dynamic Bayesian networks or hidden Markov models have only limited support [6].

Activities are estimated for each node in the spatially segmented GPS trace, as illustrated in Figure 1. In other words, our model labels a person’s activity whenever she passes through or stays at a 10m patch of the environment. We distinguish two main groups of activities, *navigation activities* and *significant activities*. Activities related to navigation are walking, driving a car, or riding a bus. Significant activities are typically performed while a user stays at a location, such as work, leisure, sleep, visit, drop off / pickup, or when the user switches transportation modes, such as getting on/off a bus, or getting in/out of a car. To determine activities, our model relies heavily on temporal features, such as duration or time of day, extracted from the GPS readings associated with each activity node.

Significant places are those locations that play a significant role in the activities of a person. Such places include a person’s home and work place, the bus stops and parking lots the person typically uses, the homes of friends, stores the person frequently shops in, and so on. Note that our model allows different activities to occur at the same significant place. Furthermore, due to signal loss and noise in the GPS readings, the same significant place can comprise multiple, different locations.

Our activity model poses two key problems for probabilistic inference. First, the model can become rather complex, including thousands of probabilistic nodes with non-trivial probabilistic constraints between them. Second, a person’s significant places depend on his activities and it is therefore not clear how to construct the model deterministically from a GPS trace. As we will show in Section 3.3, we solve the first problem by applying efficient, approximate inference algorithms for conditional

random fields. The second problem is solved by constructing the model as part of this inference. We do this by generating the highest level of the activity model (significant places) based on the outcome of inference in the lower level (activity sequence). Inference is then repeated using both levels connected appropriately.

3 Conditional Random Fields for Activity Recognition

3.1 Preliminaries

Our goal is to develop a probabilistic temporal model that can extract high-level activities from sequences of GPS readings. One possible approach is to use generative models such as hidden Markov models (HMM) or dynamic Bayesian networks. However, discriminative models such as conditional Random fields (CRF), have recently been shown to outperform generative techniques in areas such as natural language processing [10, 23], web page classification [24], and computer vision [9, 21]. We therefore decided to investigate the applicability of such models for activity recognition.

CRFs are undirected graphical models that were developed for labeling sequence data [10]. Instead of relying on Bayes rule to estimate the distribution over hidden states from observations, CRFs *directly* represent the conditional distribution over hidden states given the observations. Unlike HMMs, which assume that observations are independent given the hidden state, CRFs make no assumptions about the dependency structure between observations. CRFs are thus especially suitable for classification tasks with *complex* and *overlapped* observations.

Similar to HMMs and Markov random fields, the nodes in CRFs represent a sequence of observations (*e.g.*, GPS readings), denoted as $\mathbf{x} = \langle x_1, x_2, \dots, x_T \rangle$, and corresponding hidden states (*e.g.*, activities), denoted as $\mathbf{y} = \langle y_1, y_2, \dots, y_T \rangle$. These nodes, along with the connectivity structure imposed by undirected edges between them, define the conditional distribution $p(\mathbf{y}|\mathbf{x})$ over the hidden states \mathbf{y} . The fully connected sub-graphs of a CRF, called *cliques*, play a key role in the definition of the conditional distribution represented by a CRF. Let \mathcal{C} be the set of all cliques in a given CRF. Then, a CRF factorizes the conditional distribution into a product of *clique potentials* $\phi_c(\mathbf{x}_c, \mathbf{y}_c)$, where every $c \in \mathcal{C}$ is a clique of the graph and \mathbf{x}_c and \mathbf{y}_c are the observed and hidden nodes in such a clique. Clique potentials are functions that map variable configurations to non-negative numbers. Intuitively, a potential captures the “compatibility” among the variables in the clique: the larger the potential value, the more likely the configuration. Using clique potentials, the conditional distribution over the hidden state is written as

$$p(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{x}_c, \mathbf{y}_c), \quad (1)$$

where $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{x}_c, \mathbf{y}_c)$ is the normalizing partition function. The computation of this partition function is exponential in the size of \mathbf{y} since it requires summation over all possible configurations of hidden states \mathbf{y} . Hence, exact inference is possible for a limited class of CRF models only.

Without loss of generality, potentials $\phi_c(\mathbf{x}_c, \mathbf{y}_c)$ are described by log-linear combinations of *feature functions* $f_c()$, *i.e.*,

$$\phi_c(\mathbf{x}_c, \mathbf{y}_c) = \exp(\mathbf{w}_c^T \cdot f_c(\mathbf{x}_c, \mathbf{y}_c)), \quad (2)$$

where \mathbf{w}_c^T is the transpose of a weight vector \mathbf{w}_c , and $f_c(\mathbf{x}_c, \mathbf{y}_c)$ is a function that extracts a vector of features from the variable values. The feature functions, which are often binary or real valued, are typically designed by the user (combinations of such functions can be learned from data [15]). As we will show in Section 3.3, the weights are learned from labeled training data. Intuitively, the weights represent the importance of different features for correctly identifying the hidden states. The log-linear feature representation (2) is very compact and guarantees the non-negativeness of potential values. We can write the conditional distribution (1) as

$$p(\mathbf{y} | \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in \mathcal{C}} \exp \left\{ \mathbf{w}_c^T \cdot f_c(\mathbf{x}_c, \mathbf{y}_c) \right\} \quad (3)$$

$$= \frac{1}{Z(\mathbf{x})} \exp \left\{ \sum_{c \in \mathcal{C}} \mathbf{w}_c^T \cdot f_c(\mathbf{x}_c, \mathbf{y}_c) \right\} \quad (4)$$

(4) follows by moving the products into the exponent. Before we describe how to perform efficient inference and learning in CRFs, we will now show how CRFs can be used to implement our hierarchical activity model.

3.2 Application to Activity Recognition

GPS to street map association

As mentioned above, we segment GPS traces by grouping consecutive GPS readings based on their spatial relationship. Without a street map, this segmentation can be performed by simply combining all consecutive readings that are within a certain distance from each other (10m in our implementation). However, it might be desirable to associate GPS traces to a street map, for example, in order to relate locations to addresses in the map. Street maps are represented by graph structures, where one edge typically represents a city block section of a street, and a vertex is an intersection between streets [12].

To jointly estimate the GPS to street association and the trace segmentation, we associate each GPS measurement to a 10m patch on a street edge¹. As shown in Fig. 5(a) in Section 4, GPS traces can deviate significantly from the street map, mostly because of measurement errors and inaccuracies in street maps. One straightforward way to perform this association is to snap each GPS reading to the nearest street patch. However, such an approach would clearly give wrong results in situations such as the one shown in Fig. 5(a). To generate a consistent association, we construct a CRF

¹ In [12], we showed how to perform such an association using Rao-Blackwellised particle filters with multiple Kalman filters moving through the street graph. Since the focus of this work is on high level activities and places rather than accurate tracking, we use this more straightforward and efficient approach to trace segmentation.

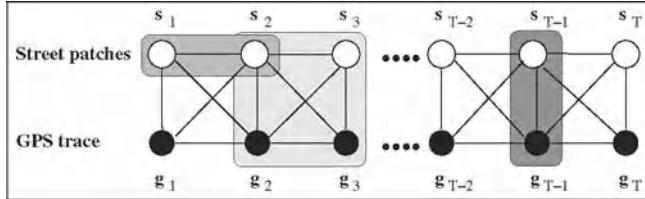


Fig. 2. CRF for associating GPS measurements to street patches. The shaded areas indicate different types of cliques.

that takes into account the spatial relationship between GPS readings. The structure of this CRF is shown in Figure 2. The observed, solid nodes correspond to GPS readings g_t , and the white nodes represent the street patches s_t , which correspond to the hidden state y in Section 3.1. The values of each s_t range over the street patches in the map that are within a certain distance of the GPS reading g_t . The lines in Figure 2 define the clique structure of the CRF. We distinguish three types of cliques, for which potentials are defined via the following feature functions:

- **Measurement cliques** (dark grey in Figure 2): GPS noise and map uncertainty are considered by cliques whose features measure the squared distance between a GPS measurement and the center of the patch it is associated with:

$$f_{\text{meas}}(g_t, s_t) = \frac{\|g_t - s_t\|^2}{\sigma^2}$$

where g_t is the location of the t -th GPS reading. With slight abuse of notation, we denote by s_t the center of one of the street patches in the vicinity of g_t (s_t and g_t are instantiated to a value). σ is used to control the scale of the distance (note that this feature function corresponds to a Gaussian noise model for GPS measurements). Obviously, when combined with a negative weight, this feature prefers associations in which GPS readings are snapped to nearby patches. The feature f_{meas} is used for the potential of all cliques connecting GPS readings and their street patches.

- **Consistency cliques** (light grey): Temporal consistency is ensured by four node cliques that compare the spatial relationship between consecutive GPS readings and the spatial relationship between their associated patches. The more similar these relationships, the more consistent the association. This comparison is done via a feature function that compares the vectors between GPS readings and associated patches:

$$f_{\text{temp}}(g_t, g_{t+1}, s_t, s_{t+1}) = \frac{\|(g_{t+1} - g_t) - (s_{t+1} - s_t)\|^2}{\sigma^2}$$

Here, s_t and s_{t+1} are the centers of street patches associated at two consecutive times.

- **Smoothness cliques** (medium grey): These cliques prefer traces that do not switch frequently between different streets. For instance, it is very unlikely that a person drives down a street and switches for two seconds to another street at an

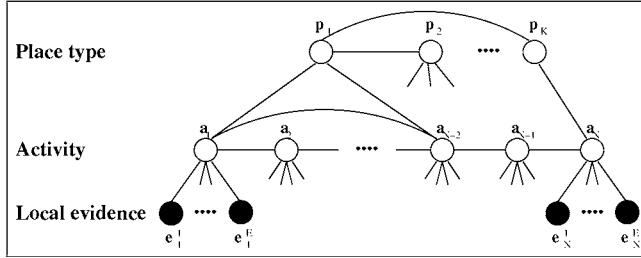


Fig. 3. CRF for labeling activities and places. Activity nodes a_i range over activities, and place nodes p_i range over types of places. Each activity node is connected to E observed local evidence nodes e_i^1 to e_i^E . Local evidence comprises information such as time of day, duration, and motion velocity. Place nodes are generated based on the activities inferred at the activity level. Each place is connected to all activity nodes that are within a certain range.

intersection. To model this information, we use binary features that test whether consecutive patches are on the same street, on neighboring streets, or in the same direction. For example, the following binary feature examines if both street and direction are identical:

$$f_{\text{smooth}}(s_t, s_{t+1}) = \delta(s_t.\text{street}, s_{t+1}.\text{street}) \cdot \delta(s_t.\text{dir}, s_{t+1}.\text{dir}) \quad (5)$$

where $\delta(u, v)$ is the indicator function which equals 1 if $u = v$ and 0 otherwise.

Using the feature functions defined above, the conditional distribution of the CRF shown in Figure 2 can be written as

$$p(\mathbf{s}|\mathbf{g}) = \frac{1}{Z(\mathbf{x})} \exp \left\{ \sum_{t=1}^T \mathbf{w}_m \cdot \mathbf{f}_{\text{meas}}(g_t, s_t) + \sum_{t=1}^{T-1} \left(\mathbf{w}_t \cdot \mathbf{f}_{\text{temp}}(g_t, g_{t+1}, s_t, s_{t+1}) + \mathbf{w}_s \cdot \mathbf{f}_{\text{smooth}}(s_t, s_{t+1}) \right) \right\} \quad (6)$$

where \mathbf{w}_m , \mathbf{w}_t and \mathbf{w}_s are the corresponding feature function weights. The reader may notice that the weights and feature functions are independent of the time index. In the context of parameter learning, this independence is often referred to as parameter sharing, which we will discuss briefly in Section 3.4. Figure 5(a) illustrates the maximum a posteriori association of a GPS trace to a map. Intuitively, this sequence corresponds to the MAP sequence that results from tracking a person's location on the discretized street map. Such an association also provides a unique segmentation of the GPS trace. This is done by combining consecutive GPS readings that are associated to the same street patch.

Inferred activities and types of significant places

Once a GPS trace is segmented, our system estimates the activity performed at each segment and a person's significant places. To do so, it generates a new CRF that contains a hidden activity node for every segment extracted from the GPS trace. This CRF consists of the two lower levels of the one shown in Figure 3. Each activity node is connected to various features, summarizing information resulting from the GPS segmentation. These features include:

- Temporal information such as time of day, day of week, and duration of the stay. These measures are discretized in order to allow more flexible feature functions. For example, time of day can be *Morning*, *Noon*, *Afternoon*, *Evening*, or *Night*. The feature functions for the cliques connecting each activity node to one of the solid nodes in the CRF shown in Figure 3 are binary indicator functions, one for each possible combination of temporal feature and activity. For instance, one such function returns 1 if the activity is work and the time of day is morning, and 0 otherwise.
- Average speed through a segment, which is important for discriminating different transportation modes. The speed value is also discretized and indicator features are used, just as with temporal information. This discretization has the advantage over a linear feature function that it is straightforward to model multi-modal velocity distributions.
- Information extracted from geographic databases, such as whether a patch is on a bus route, whether it is close to a bus stop, and whether it is near a restaurant or grocery store. Again, we use indicator features to incorporate this information.
- Additionally, each activity node is connected to its neighbors. These features measure compatibility between types of activities at neighboring nodes in the trace. For instance, it is extremely unlikely that a person will get on the bus at one location and drive a car at the neighboring location right afterwards. The corresponding feature function is $f(a_i, a_{i+1}) = \delta(a_i, \text{OnBus}) \cdot \delta(a_{i+1}, \text{Car})$, where a_i and a_{i+1} are specific activities at two consecutive activity nodes. The weight of this feature should be a negative value after supervised learning, thereby giving a labeling that contains this combination a lower probability.

Our model also aims to determine those places that play a significant role in the activities of a person, such as home, workplace, friends' home, grocery stores, restaurants, and bus stops. The nodes representing such *significant places* comprise the upper level of the CRF shown in Figure 3. However, since these places are not known *a priori*, we must additionally detect a person's significant places. To incorporate place detection into our system, we use an iterative algorithm that re-estimates activities and places. Before we describe this algorithm, let us first look at the features that are used to determine the types of significant places under the assumption that the location and number of these places is known. In order to infer place types, we use the following features for the cliques connected to the place nodes p_i in the CRF:

- The activities that occur at a place strongly indicate the type of the place. For example, at grocery stores people mainly do shopping, and at a friends' home people either visit or pick up / drop off someone. Our features consider the *frequency* of the different activities occurring at a place. This is done by generating a clique for each place that contains all activity nodes in its vicinity. For example, the nodes p_1 , a_1 , and a_{N-2} in Figure 3 form such a clique. The model then counts the different activities occurring at each place. In our experiments, we discretize the counts into four categories: count = 0, count = 1, $2 \leq \text{count} \leq 3$, and $\text{count} \geq 4$. Then for each combination of type of place, type of activity, and frequency category, we have an indicator feature.

Table 1. Algorithm for jointly inferring significant places and activities.

```

1. Input: GPS trace  $\langle g_1, g_2, \dots, g_T \rangle$ 
2.  $i := 0$ 
3. // Generate activity segments and evidence by grouping GPS readings
 $(\langle a_1, \dots, a_N \rangle, \langle e_1^1, \dots, e_1^E, e_2^1, \dots, e_N^E \rangle) := \text{spatial\_segmentation}(\langle g_1, \dots, g_T \rangle)$ 
4. // Generate CRF containing activity and local evidence nodes
 $\text{CRF}_0 := \text{instantiate\_crf}(\langle \rangle, \langle a_1, \dots, a_N \rangle, \langle e_1^1, \dots, e_N^E \rangle)$ 
5. // Determine MAP sequence of activities
 $\mathbf{a}^*_0 := \text{MAP\_inference}(\text{CRF}_0)$ 
6. do
7.    $i := i + 1$ 
8.   // Generate places by clustering significant activities
 $\langle p_1, \dots, p_K \rangle_i := \text{generate\_places}(\mathbf{a}^*_{i-1})$ 
9.   // Generate complete CRF with instantiated places
 $\text{CRF}_i := \text{instantiate\_crf}(\langle p_1, \dots, p_K \rangle_i, \langle a_1, \dots, a_N \rangle, \langle e_1^1, \dots, e_N^E \rangle)$ 
10.  // Perform MAP inference in complete CRF
 $\langle \mathbf{a}_i^*, \mathbf{p}_i^* \rangle := \text{MAP\_inference}(\text{CRF}_i)$ 
11. until  $\mathbf{a}_i^* = \mathbf{a}_{i-1}^*$ 
12. return  $\langle \mathbf{a}_i^*, \mathbf{p}_i^* \rangle$ 

```

- A person usually has only a limited number of different homes or work places. To use this knowledge to improve labeling places, we add two additional summation cliques that count the number of different homes and work places. These counts provide soft constraints that bias the system to generate interpretations that result in reasonable numbers of different homes and work places. The features are simply the counts, which make the likelihood of labelings decrease exponentially as the counts increase.

Note that the above two types of features can generate very large cliques in the CRF. This is because we must build a clique for all the activities at a place to count the frequencies of activities, and connect all the place nodes to count the number of homes or workplaces. In [13] we show how such features can be computed efficiently, even for large cliques.

Place Detection and Labelling Algorithm

The CRF discussed so far assumes that the location and number of a person's significant places is known in advance. Since these places are not known, it is necessary to additionally infer the *structure* of the hierarchical CRF shown in Figure 3. Table 1 summarizes our algorithm for efficiently constructing this CRF. The algorithm takes as input a GPS trace. In Step 3, this trace is segmented into activity nodes a_i . Each such node is characterized by local evidence e_i^j , which is extracted from the GPS readings associated to it. As discussed above, segmentation of a trace is performed by either clustering consecutive GPS readings that are nearby or associating the GPS

trace to a discretized street map using the CRF shown in Figure 2. The activity nodes and their evidence are then used in Step 4 to generate a CRF such as the one shown in Figure 3. However, since significant places are not yet known at this stage, CRF_0 contains no place nodes. Maximum a posteriori inference is then performed in this restricted CRF so as to determine the MAP activity sequence a^*_0 , which consists of a sequence of locations and the activity performed at that location (Step 5). Within each iteration of the loop starting at Step 6, such an activity sequence is used to extract a set of significant places. This is done by classifying individual activities in the sequence according to whether or not they belong to a significant place. For instance, while walking, driving a car, or riding a bus are not associated with significant places, working or getting on or off the bus indicate a significant place. All instances at which a *significant activity* occurs generate a place node. Because a place can be visited multiple times within a sequence, we perform clustering and merge duplicate places into the same place node. This classification and clustering is performed by the algorithm `generate_places()`, which returns a set of K place nodes p_k in Step 8. These places, along with the activity nodes a_i and their local evidence e_i^j are used to generate a complete CRF. Step 10 performs MAP estimation in this new CRF. Since this CRF has a different structure than the initial CRF_0 , it might generate a different MAP activity sequence. If this is the case, then the algorithm returns to Step 6 and re-generates the set of places using this improved activity sequence. This process is repeated until the activity sequence does not change, which is tested in Step 11. Finally, the algorithm returns the MAP activity sequence along with the set of places and their MAP types. In our experiments we observed that this algorithm converges very quickly, typically after three or four iterations. Our experiments also show that this algorithm is extremely efficient and robust.

3.3 Inference

In this section we will provide an overview of inference techniques for CRFs. We will use x to denote observations and y to denote hidden states. Given a set of observations, inference in a CRF can have two tasks: to estimate the marginal distribution of each hidden variable, or to estimate the most likely configuration of the hidden variables (*i.e.*, the maximum a posteriori, or MAP, estimation). Both tasks can be solved under a framework called *belief propagation* (BP), which works by sending local messages through the graph structure of the model. The BP algorithm was originally proposed in the context of Bayesian networks [20], and was formulated equivalently in models such as factor graphs [8] and Markov networks (including CRFs) [25]. BP generates provably correct results if the graph has no loops, such as trees or polytrees [20]. If the graph contains loops, in which case BP is called loopy BP, then the algorithm is only approximate and might not converge to the correct probability distribution [16].

Without loss of generality, we only describe the BP algorithm for pairwise CRFs, which are CRFs that only contain cliques of size two. We will briefly discuss how to use BP in non-pairwise CRFs in the last paragraph of this section. Before running the inference algorithm in a pair-wise CRF, it is possible to remove all observed nodes x by merging their values into the corresponding potentials; that is, a potential $\phi(x, y)$ can be written as $\phi(y)$ because x is fixed to one value. Therefore, the only

potentials in a pair-wise CRF are local potentials, $\phi(y_i)$, and pair-wise potentials, $\phi(y_i, y_j)$. Corresponding to the two types of inference problems, there are two types of BP algorithms: *sum-product* for marginal estimation and *max-product* for MAP estimation.

Sum-product for marginal estimation

In the BP algorithm, we introduce a “message” $m_{ij}(y_j)$ for each pair of neighbors y_i and y_j , which is a distribution (not necessarily normalized) sent from node i to its neighbor j about which state variable y_j should be in. The messages propagate through the CRF graph until they (possibly) converge, and the marginal distributions can be estimated from the stable messages. A complete BP algorithm defines how to initialize messages, how to update messages, how to schedule the message updates, and when to stop passing messages.

- **Message initialization:** All messages $m_{ij}(y_j)$ are initialized as uniform distributions over y_j .
- **Message update rule:** The message $m_{ij}(y_j)$ sent from node i to its neighbor j is updated based on local potentials $\phi(y_i)$, the pair-wise potential $\phi(y_i, y_j)$, and all the messages to i received from i ’s neighbors other than j (denoted as $n(i) \setminus j$). More specifically, for sum-product, we have

$$m_{ij}(y_j) = \sum_{y_i} \phi(y_i) \phi(y_i, y_j) \prod_{k \in n(i) \setminus j} m_{ki}(y_i) \quad (7)$$

- **Message update order:** The algorithm iterates the message update rule until it (possibly) converges. At each iteration, it usually updates each message once, where the update order might affect the convergence speed.
- **Convergence conditions:** To test whether the algorithm converged, BP measures the difference between the previous messages and the updated ones. The convergence condition is met when all the differences are below a given threshold ϵ .

In the sum-product algorithm, after all messages are converged, it is easy to calculate the marginals of each node and each pair of neighboring nodes as

$$b(y_i) \propto \phi(y_i) \prod_{j \in n(i)} m_{ji}(y_i) \quad (8)$$

$$b(y_i, y_j) \propto \phi(y_i) \phi(y_j) \phi(y_i, y_j) \prod_{k \in n(i) \setminus j} m_{ki}(y_i) \prod_{l \in n(j) \setminus i} m_{lj}(y_j) \quad (9)$$

The above algorithm can be applied to any topology of pair-wise CRFs. When the network structure does not have a loop (for example, when it is a tree), the obtained marginals are guaranteed to be exact. When the structure has loops, the BP algorithm usually cannot obtain exact marginals, or it may even not converge. Fortunately, empirical experiments show that loopy belief propagation often converges to a good approximation of the correct posterior.

Max-product for MAP estimation

We denote the messages sent in the max-product algorithm as $m_{ij}^{\max}(y_j)$. The whole algorithm of max-product is very similar to sum-product, except that in the message update rule summation is replaced by maximization. The new rule becomes

$$m_{ij}^{\max}(y_j) = \max_{y_i} \phi(y_i) \phi(y_i, y_j) \prod_{k \in n(i) \setminus j} m_{ki}^{\max}(y_i). \quad (10)$$

We run the max-product algorithm in the same way as for sum-product. After the algorithm converges, we calculate the MAP belief at each node y_i as

$$b(y_i) \propto \phi(y_i) \prod_{j \in n(i)} m_{ji}^{\max}(y_i). \quad (11)$$

If there is a unique MAP configuration \mathbf{y}^* , then the components of \mathbf{y}^* are simply the most likely values according to the MAP belief (11).

So far, we explained the two BP algorithms in the context of pairwise CRFs. For non-pairwise CRFs, there is a standard way to convert them to pairwise ones [25]. Intuitively, this conversion generates a new node for each clique of size greater than two. The state space of the new node consists of the joint state of the nodes it was generated from. Thus, the complexity of belief propagation is exponential in the number of nodes in the largest clique of the CRF.

In our application, the summation (or counting) features could introduce large cliques containing up to 30 nodes. Standard belief propagation would be intractable for such cliques. Fortunately, it is possible to convert cliques generated for summation features to tree-structured CRFs. In such structures, BP inference can be done in polynomial time, and for sum-product it is even possible to apply the Fast Fourier Transform (FFT) to further speed up message passing (see [13] for details).

3.4 Parameter Learning

The goal of parameter learning is to determine the weights of the feature functions used in the conditional likelihood (4). CRFs learn these weights *discriminatively*, that is, the weights are determined so as to maximize the conditional likelihood $p(\mathbf{y}|\mathbf{x})$ of labeled training data. This is in contrast to generative learning, which aims to learn a model of the joint probability $p(\mathbf{y}, \mathbf{x})$. Ng and Jordan [17] present a discussion and comparison of these two learning regimes, concluding that discriminative learning asymptotically reaches superior performance but might require more training examples until its performance converges.

Maximum Likelihood (ML) Estimation

As can be seen in (4), given labeled training data (\mathbf{x}, \mathbf{y}) , the conditional likelihood $p(\mathbf{y}|\mathbf{x})$ only depends on the feature weights \mathbf{w}_c . In the derivation of the learning algorithm it will be convenient to re-write (4) as

$$p(\mathbf{y} | \mathbf{x}, \mathbf{w}) = \frac{1}{Z(\mathbf{x})} \exp \left\{ \sum_{c \in C} \mathbf{w}_c^T \cdot \mathbf{f}_c(\mathbf{x}_c, \mathbf{y}_c) \right\} \quad (12)$$

$$= \frac{1}{Z(\mathbf{x}, \mathbf{w})} \exp \{ \mathbf{w}^T \cdot \mathbf{f}(\mathbf{x}, \mathbf{y}) \}, \quad (13)$$

where \mathbf{w} and \mathbf{f} are the vectors resulting from “stacking” the weights and the feature functions for all cliques in the CRF, respectively. In order to make the dependency on \mathbf{w} more explicit, we write the conditional likelihood as $p(\mathbf{y}|\mathbf{x}, \mathbf{w})$. A common parameter estimation method is to search for the \mathbf{w} that maximizes this likelihood, or equivalently, that minimizes the *negative log-likelihood*, $-\log p(\mathbf{y}|\mathbf{x}, \mathbf{w})$ [10, 24, 14]. To avoid overfitting, one typically imposes a so-called shrinkage prior on the weights to keep them from getting too large. More specifically, we define the objective function to minimize as follows:

$$L(\mathbf{w}) \equiv -\log p(\mathbf{y} | \mathbf{x}, \mathbf{w}) + \frac{\mathbf{w}^T \mathbf{w}}{2\sigma^2} \quad (14)$$

$$= -\mathbf{w}^T \cdot \mathbf{f}(\mathbf{x}, \mathbf{y}) + \log Z(\mathbf{x}, \mathbf{w}) + \frac{\mathbf{w}^T \mathbf{w}}{2\sigma^2} \quad (15)$$

The rightmost term in (14) serves as a zero-mean, Gaussian prior with variance σ^2 on each component of the weight vector. (15) follows directly from (14) and (13). While there is no closed-form solution for maximizing (15), it can be shown that (15) is convex relative to \mathbf{w} . Thus, L has a global optimum which can be found using numerical gradient algorithms. It can be shown that the gradient of the objective function $L(\mathbf{w})$ is given by

$$\nabla L(\mathbf{w}) = -\mathbf{f}(\mathbf{x}, \mathbf{y}) + E_{P(\mathbf{y}'|\mathbf{x}, \mathbf{w})}[\mathbf{f}(\mathbf{x}, \mathbf{y}')] + \frac{\mathbf{w}}{\sigma^2} \quad (16)$$

where the second term is the expectation over the distribution $P(\mathbf{y}' | \mathbf{x}, \mathbf{w})$. Therefore, the gradient is the difference between the *empirical feature values* $\mathbf{f}(\mathbf{x}, \mathbf{y})$ and the *expected feature values* $E_{P(\mathbf{y}'|\mathbf{x}, \mathbf{w})}[\mathbf{f}(\mathbf{x}, \mathbf{y}')]$, plus a prior term. To compute the expectation over the feature values it is necessary to run inference in the CRF using the current weights \mathbf{w} . This can be done via belief propagation as discussed in the previous section. Sha and Pereira [23] showed that numerical optimization algorithms, such as conjugate gradient or quasi-Newton techniques, typically converge reasonably fast to the global optimum.

Maximum Pseudo-Likelihood (MPL) Estimation

Maximizing the likelihood requires running an inference procedure at each iteration of the optimization, which can be very expensive. An alternative is to maximize the *pseudo-likelihood* of the training data [3], which is the sum of all the *local likelihoods*, $p(\mathbf{y}_i | \text{MB}(\mathbf{y}_i))$, where $\text{MB}(\mathbf{y}_i)$ is the Markov blanket of variable \mathbf{y}_i containing the immediate neighbors of \mathbf{y}_i in the CRF graph (note that the value of each node is known during learning). The pseudo-likelihood can be written as

$$\sum_{i=1}^n p(\mathbf{y}_i | \text{MB}(\mathbf{y}_i), \mathbf{w}) = \sum_{i=1}^n \frac{1}{Z(\text{MB}(\mathbf{y}_i), \mathbf{w})} \exp\{\mathbf{w}^T \cdot \mathbf{f}(\mathbf{y}_i, \text{MB}(\mathbf{y}_i))\}, \quad (17)$$

where $\mathbf{f}(\mathbf{y}_i, \text{MB}(\mathbf{y}_i))$ are the local feature values involving variable \mathbf{y}_i , and $Z(\text{MB}(\mathbf{y}_i), \mathbf{w}) = \sum_{\mathbf{y}'_i} \exp\{\mathbf{w}^T \cdot \mathbf{f}(\mathbf{y}'_i, \text{MB}(\mathbf{y}'_i))\}$ is the *local* normalizing function. Computing pseudo-likelihood is much more efficient than computing likelihood

$p(\mathbf{y}|\mathbf{x}, \mathbf{w})$, because pseudo-likelihood only requires computing local normalizing functions and avoids computing the global partition function $Z(\mathbf{x}, \mathbf{w})$.

As with ML, in practice we minimize the negative log-pseudo-likelihood and a shrinkage prior, and the objective function becomes

$$PL(\mathbf{w}) \equiv -\sum_{i=1}^n \log p(\mathbf{y}_i | MB(\mathbf{y}_i), \mathbf{w}) + \frac{\mathbf{w}^T \mathbf{w}}{2\sigma^2} \quad (18)$$

$$= \sum_{i=1}^n (-\mathbf{w}^T \cdot \mathbf{f}(\mathbf{y}_i, MB(\mathbf{y}_i)) + Z(MB(\mathbf{y}_i), \mathbf{w})) + \frac{\mathbf{w}^T \mathbf{w}}{2\sigma^2} \quad (19)$$

Again, $PL(\mathbf{w})$ is a convex function and it is possible to use gradient-based algorithms to find the \mathbf{w} that minimizes $PL(\mathbf{w})$. The gradient can be computed as

$$\nabla PL(\mathbf{w}) = \sum_{i=1}^n \left(-\mathbf{f}(\mathbf{y}_i, MB(\mathbf{y}_i)) + E_{P(\mathbf{y}'_i | MB(\mathbf{y}_i), \mathbf{w})} [\mathbf{f}(\mathbf{y}'_i, MB(\mathbf{y}_i))] \right) + \frac{\mathbf{w}}{\sigma^2}. \quad (20)$$

As we can see, (20) can be expressed as the difference between empirical feature values and expected feature values, similar to (16). However, the key difference is that (20) can be evaluated very efficiently without running a complete inference procedure. Learning by maximizing pseudo likelihood has been shown to perform very well in several domains [9, 22]. In our experiments we found that this type of learning is extremely efficient and consistently achieves good results. The reader may notice that this technique cannot be used for inference, since it assumes that the hidden states \mathbf{y} are known.

Parameter Sharing

The definition of the weight vector and its gradient described above does not support *parameter sharing*, which requires the learning algorithm to learn the same parameter values (weights) for different cliques in the CRF. For instance, the conditional likelihood (5) of the CRF described in Section 3.2 only contains three different weights, one for each *type* of feature. The same weight \mathbf{w}_m is used for each clique containing a street patch node s_t and a GPS reading node g_t . To learn such kinds of models, one has to make sure that all the weights belonging to a certain type of feature are identical. As it turns out, the gradients with respect to such shared weights are almost identical to the gradients (16) and (20). The only difference lies in the fact that the gradient for a shared weight is given by the sum of all the gradients computed for the individual cliques in which this weight occurs [24, 14].

Parameter sharing can be modeled conveniently using probabilistic relational models such as relational Markov networks [24, 14]. These techniques allow the automatic specification and construction of CRF models using so-called clique templates, which enable the specification of parameter sharing for inference and learning.

4 Experimental Results

In our experiments we evaluate how well our system can *extract and label a person's activities and significant places*. We also demonstrate that it is feasible to learn models from data collected by a set of people and to apply this model to another person.

We collected GPS data traces from four different persons, approximately six days of data per person. The data from each person consisted of roughly 40,000 GPS measurements, resulting in about 10,000 10m segments per person. We then manually labeled all activities and significant places in these traces. We used leave-one-out cross-validation for evaluation, that is, learning was performed based on the data collected by three persons and the learned model was evaluated on the fourth person. We used pseudo-likelihood for learning, which took (on a 1.5 GHz PC) about one minute to converge on the training data. Pseudo-likelihood converged in all our experiments. We did not use loopy belief propagation for learning since it did not always converge (even after several hours). This is most likely due to the fact that the approximation of this algorithm is not good enough to provide accurate gradients for learning. However, we successfully used loopy BP as inference approach in all our evaluation runs. For each evaluation, we used the algorithm described in Table 1, which typically extracted the MAP activities and places from one week's trace within one minute of computation. When a street map was used, the association between GPS trace and street map performed in Step 3 of the algorithm took additional four minutes (see also Section 3.2).

Example analysis

The different steps involved in the analysis of a GPS trace are illustrated in Figure 4. The second panel (b) shows the GPS trace snapped to 10m patches on the street map. This association is performed by Step 3 of the algorithm given in Table 1, using the CRF shown in Figure 2. The visited patches, along with local information such as time of day or duration, are used to generate the activity CRF. This is done by Step 4 in Table 1, generating the activity level of Figure 3. MAP inference in this CRF determines one activity for each patch visit, as shown in panel (c) of Figure 4 (Step 5 of the algorithm). Note that this example analysis misses the get-off-bus activity at the left end of the bus trip. The significant activities in the MAP sequence are clustered and generate additional place nodes in a new CRF (Steps 8 and 9 in Table 1). MAP inference in this CRF provides labels for the detected places, as shown in Figure 4(d). The algorithm repeats generation of the CRFs until the MAP activity sequence does not change any more. In all experiments, this happens within four iterations.

Figure 5(a) provides another example of the quality achieved by our approach to snapping GPS traces to street maps. Note how the complete trace is snapped consistently to the street map. Table 2 shows a typical summary of a person's day provided by the MAP sequence of activities and visited places. Note that the system determines where the significant places are, how the person moves between them, and what role the different places play for this person.

Extracting significant places

In this experiment we compare our system's ability to detect significant places to the results achieved with a widely-used approach that applies a time threshold to determine whether or not a location is significant [1, 7, 12, 14]. Our approach was trained on data collected by three people and tested on the fourth person. For the threshold method, we generated results for different thresholds from 1 minute to 10

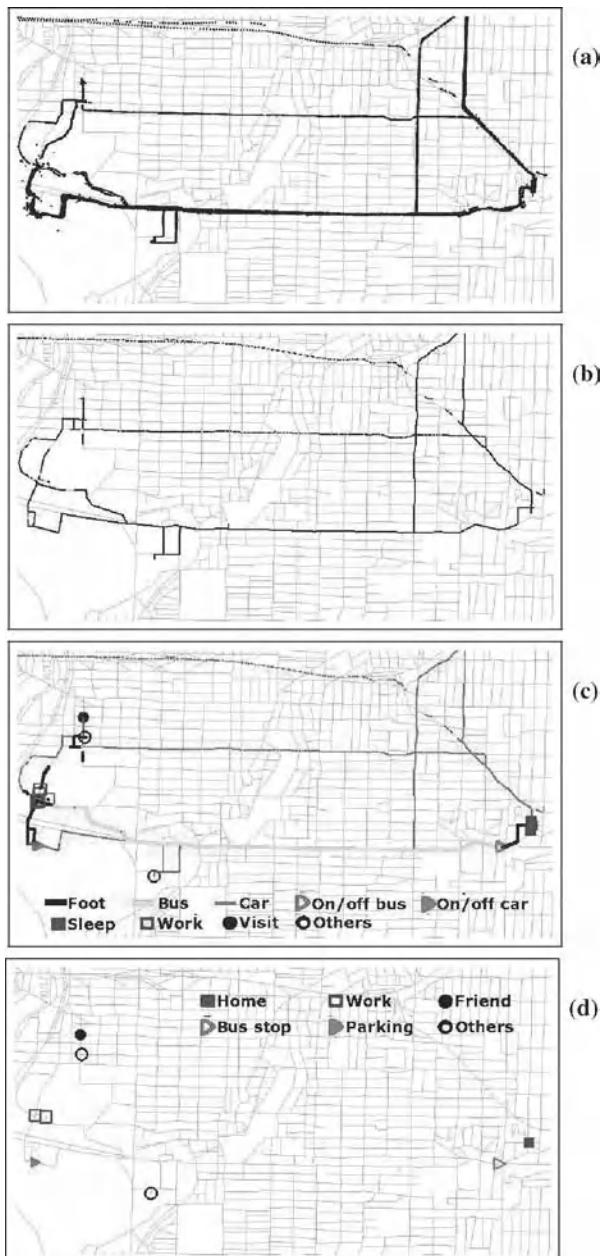
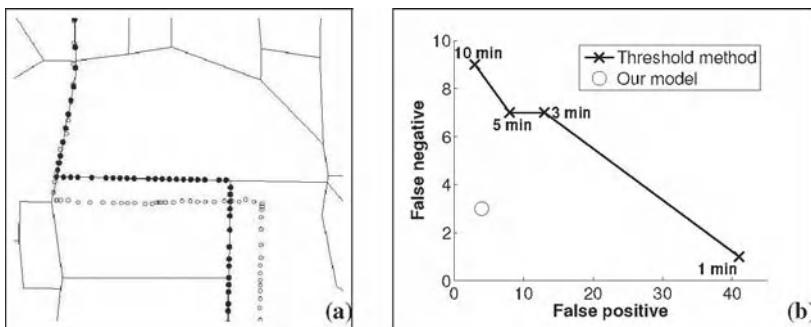


Fig. 4. Illustration of inference on part of a GPS trace, which visited this 4km x 2km area several times. (a) The raw GPS data has substantial variability due to sensor noise. (b) GPS trace snapped to 10m street patches, multiple visits to the same patch are plotted on top of each other. (c) Activities estimated for each patch. (d) Places generated by clustering significant activities, followed by a determination of place types.

Table 2. Summary of a typical day based on the inference results.

Time	Activity and transportation
8:15am - 8:34am	Drive from home ₁ to parking lot ₂ , walk to workplace ₁ ;
8:34am - 5:44pm	Work at workplace ₁ ;
5:44pm - 6:54pm	Walk from workplace ₁ to parking lot ₂ , drive to friend ₃ 's place;
6:54pm - 6:56pm	Pick up/drop off at friend ₃ 's place;
6:56pm - 7:15pm	Drive from friend ₃ 's place to other place ₂ ;
7:15pm - 9:01pm	Other activity at other place ₂ ;
9:01pm - 9:20pm	Drive from other place ₂ to friend ₁ 's place;
9:20pm - 9:21pm	Pick up/drop off at friend ₁ 's place;
9:21pm - 9:50pm	Drive from friend ₁ 's place to home ₁ ;
9:50pm - 8:22am	Sleep at home ₁ .

**Fig. 5.** (a) GPS trace (gray circles) and the associated grid cells (black circles) on the street map (lines). (b) Accuracy of extracting significant places.

minutes. The data contained 51 different significant places. Figure 5(b) shows the false positive and false negative rates achieved with the two approaches. As can be seen, our approach clearly outperforms the threshold method. Any fixed threshold is not satisfactory: low thresholds have many false negatives, and high thresholds result in many false positives. In contrast, our model performs much better: it only generates 4 false positives and 3 false negatives.

Labeling places and activities using models learned from others

Table 3 and Table 4 summarize the results achieved with our system on the cross-validation data. Table 3 shows activity estimation results on the significant activities only. An instance was considered a false positive (FP) if a significant activity was detected when none occurred, and was considered false negative (FN) if a significant activity occurred but was labeled as non-significant such as walking. The results are given for models with and without taking the detected places into account. More specifically, without places are results achieved by CRF₀ generated by Step 5 of the algorithm in Table 1, and results with places are those achieved after model convergence. When the results of both approaches are identical, only one number is given; otherwise, the first number gives results achieved with the complete model. The table shows two main results. First, the accuracy of our approach is quite high, especially when considering that the system was evaluated on only one week of

Table 3. Activity confusion matrix of cross-validation data with (left values) and without (right values) considering places for activity inference.

Truth	Inferred labels							FN
	Work	Sleep	Leisure	Visit	Pickup	On/off car	Other	
Work	12 / 11	0	0 / 1	0	0	0	1	0
Sleep	0	21	1	2	0	0	0	0
Leisure	2	0	20 / 17	1 / 4	0	0	3	0
Visiting	0	0	0 / 2	7 / 5	0	0	2	0
Pickup	0	0	0	0	1	0	0	2
On/Off car	0	0	0	0	1	13 / 12	0	2 / 3
Other	0	0	0	0	0	0	37	1
FP	0	0	0	0	2	2	3	-

data and was trained on only three weeks of data collected by different persons. Second, performing joint inference over activities and places increases the quality of inference. The reason for this is that a place node connects all the activities occurring in its spatial area so that these activities can be labeled in a more consistent way.

These results were generated when taking a street map into account. We also performed an analysis of the system without using the street map. In this case, the GPS trace was segmented into 10m segments solely based on the raw GPS values. We found that the results achieved without the street map were consistently almost identical to those achieved when a street map is available. In both cases, our system achieved above 90% accuracy for navigation activities such as car, walk, or bus, and above 85% accuracy in estimating significant activities.

Table 4. Place confusion matrix.

Truth	Inferred labels					FN
	Work	Home	Friend	Parking	Other	
Work	5	0	0	0	0	0
Home	0	4	0	0	0	0
Friend	0	0	3	0	2	0
Parking	0	0	0	8	0	2
Other	0	0	0	0	28	1
FP	0	0	1	1	2	-

The confusion matrix shown in Table 4 summarizes the results achieved on detecting and labeling significant places. As can be seen, the approach commits zero errors in labeling the home and work locations of the persons (one person had two work places). The overall accuracy in place detection and labeling is 90.6%. The place detection results were identical with and without using a street map.

5 Conclusions

We provided a novel approach to performing location-based activity recognition. In contrast to existing techniques, our approach uses one consistent framework for

both low-level inference and the extraction of a person's significant places. This is done by iteratively constructing a hierarchical conditional random field, where the upper level is generated based on MAP inference on the lower level. Once a complete model is constructed, we perform joint inference in the complete CRF. Discriminative learning using pseudo-likelihood and inference using loopy belief propagation can be performed extremely efficiently in our model: The analysis of a GPS trace collected over a week takes approximately one minute on a standard desktop PC.

Our experiments based on traces of GPS data show that our system significantly outperforms existing approaches. In addition to being able to learn a person's significant locations, it can infer low level activities such as walking, working, or getting into a bus. We demonstrate that the model can be trained from a group of persons and then applied successfully to a different person, achieving more than 85% accuracy in determining low-level activities and above 90% accuracy in detecting and labeling significant places. Our model achieves virtually identical accuracy both with and without a street map. The output of our system can also be used to generate textual summaries of a person's daily activities.

The system described here opens up various research directions. For instance, our algorithm constructs the hierarchical CRF using MAP estimation. We are currently investigating a technique that generates multiple models using an MCMC or a k-best approach. The different models can then be evaluated based on their overall data likelihood. We expect this more flexible model searching approach to generate better results especially in more complex scenarios. We are currently adding more types of sensors to our model, including data collected by a wearable multi-sensor board [11]. This sensor device collects measurements such as 3-axis acceleration, audio signals, barometric pressure, and light. Using the additional information provided by these sensors, we will be able to perform extremely fine-grained activity recognition.

Acknowledgments

The authors would like to thank Jeff Bilmes for useful comments. This work has partly been supported by DARPA's ASSIST and CALO Programme (contract numbers: NBCH-C-05-0137, SRI subcontract 27-000968) and by the NSF under grant number IIS-0093406.

References

1. D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5), 2003.
2. M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun. Learning motion patterns of people for compliant robot motion. *International Journal of Robotics Research*, 24(1), 2005.
3. J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24, 1975.
4. B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer. Easyliving: Technologies for intelligent environments. *Handheld and Ubiquitous Computing*, 2000.
5. H.H. Bui, S. Venkatesh, and G. West. Tracking and surveillance in wide-area spatial environments using the abstract hidden markov model. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1), 2001.

6. K. Gopalratnam, H. Kautz, and D. Weld. Extending continuous time bayesian networks. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2005.
7. R. Hariharan and K. Toyama. Project Lachesis: parsing and modeling location histories. In *Geographic Information Science*, 2004.
8. F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 2001.
9. S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proc. of the International Conference on Computer Vision (ICCV)*, 2003.
10. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the International Conference on Machine Learning (ICML)*, 2001.
11. J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford. A hybrid discriminative-generative approach for modeling human activities. In *Proc. of the International Joint Conference on Artificial Intelligence*, 2005.
12. L. Liao, D. Fox, and H. Kautz. Learning and inferring transportation routines. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2004.
13. L. Liao, D. Fox, and H. Kautz. Location-based activity recognition. In *Advances in Neural Information Processing Systems*, 2005.
14. L. Liao, D. Fox, and H. Kautz. Location-based activity recognition using relational Markov networks. In *Proc. of the International Joint Conference on Artificial Intelligence*, 2005.
15. A. McCallum. Efficiently inducing features of conditional random fields. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2003.
16. K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 1999.
17. A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems*, 2002.
18. D. Patterson, O. Etzioni, D. Fox, and H. Kautz. Intelligent ubiquitous computing to support Alzheimer's patients: Enabling the cognitively disabled. In *UbiCog '02: First International Workshop on Ubiquitous Computing for Cognitive Aids*, 2002.
19. D. Patterson, L. Liao, K. Gajos, M. Collier, N. Livic, K. Olson, S. Wang, D. Fox, and H. Kautz. Opportunity Knock: a system to provide cognitive assistance with transportation services. In *International Conference on Ubiquitous Computing*, 2004.
20. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.
21. A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems*, 2004.
22. M. Richardson and P. Domingos. Markov logic networks. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 2004. Conditionally accepted for publication in *Machine Learning*.
23. F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *Proc. of Human Language Technology-NAACL*, 2003.
24. B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2002.
25. J. S. Yedidia, W. T. Freeman, and Y. Weiss. *Exploring Artificial Intelligence in the New Millennium*, chapter Understanding Belief Propagation and Its Generalizations. Morgan Kaufmann Pub, 2001.

Networked Robotics

Session Overview

Networked Robotics

Tomomasa Sato¹ and Ray Jarvis²

¹ University of Tokyo

² Monash University

Robot systems with network capability is a promising frontier of robotics not only to realize new services by combining multiple RT(robot technology) and IT(information technology) components but also to know the human worlds and natural environments. This program session covers networked sensing/actuation, networked intelligence and networked control of robot systems.

The presented paper of “Networked Robotic Cameras for Collaborative Observation of Natural Environments” focuses on the camera work for collaborative observation. A novel algorithm how to collaborate multiple cameras is presented.

The future session of Networked Robotics should cover networked embedded sensing/actuation, robotic network system, integrated intelligence of the environment, and control of distributed system, including consistent interaction between human and components of the environment.

Interfaces and Interaction

Networked Robotic Cameras for Collaborative Observation of Natural Environments

Dezhen Song¹ and Ken Goldberg²

¹ CS Department, Texas A&M University dzsong@cs.tamu.edu

² IEOR Department, UC Berkeley goldberg@ieor.berkeley.edu

1 Introduction

Scientific study of animals *in situ* requires vigilant observation of detailed animal behavior over weeks or months. When animals live in remote and/or inhospitable locations, observation can be an arduous, expensive, dangerous, and lonely experience for scientists. Emerging advances in robot cameras, long-range wireless networking, and distributed sensors make feasible a new class of portable robotic “observatories” that can allow groups of scientists, via the internet, to remotely observe, record, and index detailed animal activity. As a shorthand for such an instrument, we propose the acronym *CONE: Collaborative Observatory for Natural Environments*.

One challenge is to develop a mathematical framework for collaborative observation. Collaborative observation includes (1) collaboration between humans of different backgrounds, skill sets, and authority/permission levels and (2) collaboration between humans and automated agents whose behavior arises from sensor inputs and/or computation. As illustrated in Figure 4, our framework uses a *panoramic image* and set of *activity frames* to provide a unified representation for output and for input from both human observers and sensors.

2 Related Work

Since Nikola Tesla demonstrated the first radio-controlled boat in 1898 and Goertz demonstrated a bilateral manipulator in 1954 [7], remotely operated machines have been widely desired for use in inhospitable environments such as radiation sites, undersea [1] and space exploration [3, 24, 33]. Today, teleoperation is being developed for medical diagnosis [2], manufacturing [6] and micromanipulation [27]. See Sheridan [28] for an excellent review of the extensive literature on teleoperation and telerobotics. Most of these systems require fairly complex hardware at the human interface: exoskeleton master

linkages are operated by highly trained specialists. In contrast, the Internet can provide public access by using only the interface available in standard browsers.

The hypertext transfer protocol developed at CERN in 1992 [4], provides a low-cost and publicly available network interface. In the Spring of 1994, we conjectured that we could use it to offer public access to a teleoperated robot via the Internet.

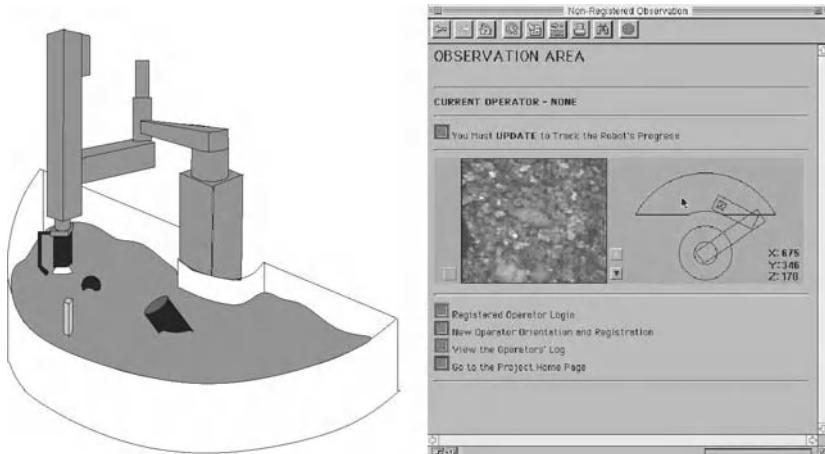


Fig. 1. Mercury Project (1994-1995). Above: Robot, camera and air nozzle above sandbox filled with buried artifacts. Below: Browser Interface using vanilla HTTP 1.0.

As illustrated in Figure 1, we set up an IBM SCARA robot arm over a semi-annular workspace containing sand and buried artifacts. We attached a CCD camera to the end of the arm along with a nozzle to direct air bursts into the sand. We then developed a HTTP 1.0 (Mosaic) browser interface to the hardware. The Mercury Project was operated by over 10,000 people and is widely regarded as the first Internet robot [11, 10].

Our subsequent project, the Telegarden, allowed users to view and interact with a remote garden filled with living plants. We incorporated a much faster Adept-1 industrial robot arm and allowed the robot to be multi-tasked to eliminate the user queue. The Telegarden was installed at a museum in Austria where it operated around the clock for nine years was operated by over 100,000 people online.

In 1994, working independently, a team led by K. Taylor and J. Trevelyan at the University of Western Australia demonstrated a remotely controlled six-axis telerobot in September 1994 [5, 17]. There are now dozens of Internet robots online, a book from MIT Press [12], and an IEEE Technical Committee

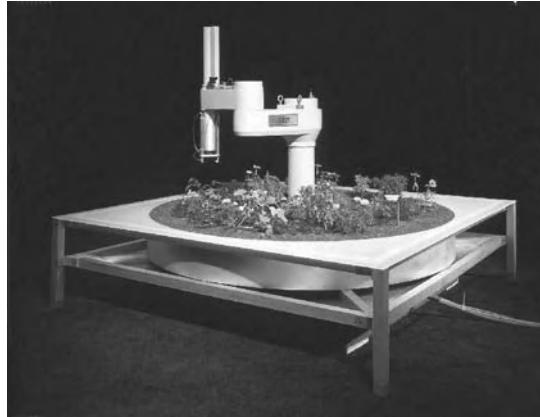


Fig. 2. The Tele-Garden (1995-2004). (with Joseph Santarromana, George Bekey, Steven Gentner, Rosemary Morris Carl Sutter, Jeff Wiegley, Erich Berger, and Thomas Steindl).

on Networked Robots that has over 200 members. See [18, 26, 20, 19, 21, 23, 15, 25, 22] examples of recent projects.

3 The Tele-Actor and ShareCam

In 1999 we began exploring other models of access control, where user inputs are combined rather than sequenced. In [9, 8], we describe an Internet-based Multiple Operator Single Robot system that use vector averaging to combine multiple mouse inputs to simultaneously control a single industrial robot arm. In [13, 14], we describe a Java-based “Spatial Dynamic Voting” (SDV) interface that collects, displays, and analyzes a sequence of spatial votes from multiple online operators at their Internet browsers. The votes can drive the motion of a single mobile robot or, for increased mobility and agility, a human “Tele-Actor”.

4 The Collaborative Frame Selection Problem

We are now developing systems based on robotic pan, tilt, zoom cameras controllable by many simultaneous viewers over the Internet. Since there is one camera and many viewers, the challenge is to resolve contention about where to point the camera.



Fig. 3. Spatial Dynamic Voting Interface and the Tele-Actor (2001-2004).

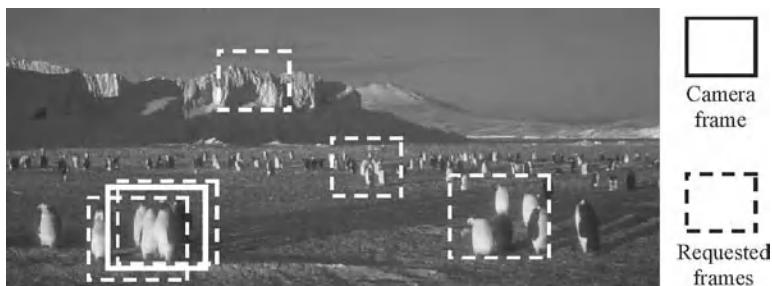


Fig. 4. Panoramic image and user or machine-requested “activity frames”.

Collaborative observation includes (1) collaboration between humans of different backgrounds, skill sets, and authority/permission levels and (2) collaboration between humans and automated agents whose behavior arises from sensor inputs and/or computation. We propose using a *panoramic image* and set of *activity frames* to provide a unified representation for output and for input from both human observers and sensors. On the output (display) side, the wide-field panoramic image provides a relative spatial context for close-up camera views.

On the input side, each activity frame is a rectangular region with the aspect ratio of the camera. As illustrated in Figure 4, human users specify activity frames of interest by drawing them with standard mouse over the panoramic image; the boundaries of the frame intuitively match each desired camera view. Below we review algorithms we've developed that efficiently process a set of activity frames to compute optimal frames for the camera.

Each activity frame is a rectangular region with the aspect ratio of the camera. As illustrated in Figure 4, human users specify activity frames of interest by drawing them with standard mouse over the panoramic image; the boundaries of the frame intuitively match each desired camera view.

Let $c = [x, y, z]$ define a rectangular camera frame (the camera has a fixed aspect ratio of 4:3). User i requests a desired frame r_i . Given requests from n users, the system must compute a single global frame c^* that will best satisfy the set of requests. Clearly simple averaging will work poorly as it can produce centered camera frames that satisfy none of the users.

We define the “Coverage-Resolution Ratio (CRR)” as a reward, or “satisfaction” metric $s(c, r_i)$ based on how closely the requested frame r_i compares with a candidate camera frame c . One sample CRR metric is described below,

$$s_i(c) = \frac{\text{Area}(r_i \cap c)}{\text{Area}(r_i)} \min\left(\frac{z_i}{z}, 1\right). \quad (1)$$

Equation 1 characterizes the intuition that satisfaction has to be an increasing function of coverage ratio $\frac{\text{Area}(r_i \cap c)}{\text{Area}(r_i)}$. In our definition, larger z or z_i means larger in frame size but lower in resolution. Therefore, an extremely large camera frame can increase coverage ratio but will decrease the resolution ratio $\frac{z_i}{z}$.

Each of n users submits a request. In the collaborative camera control, we want to find c^* , the value of c that maximizes overall satisfaction based only on the current set of requests:

$$\max_c \sum_{i=1}^n s_i(r_i, c) = \sum_{i=1}^n \frac{\text{Area}(r_i \cap c)}{\text{Area}(r_i)} \min\left(\frac{z_i}{z}, 1\right). \quad (2)$$

In each motion cycle, we servo the camera to the computed position and zoom level.

Since the reward metric is non-concave and non-differentiable, efficiently computing the optimal solution for Equation 2 is non-trivial as illustrated in Figure 5. In [31], we show that the shape of the objective function for a single user has a plateau-like shape. To efficiently compute the summation of a set of plateaus, we developed an $O(mn^2)$ exact algorithm based on idea of sweeping and incremental computation. Since the camera may have a continuously variable zoom and user requests are not necessarily rectangular, we have developed a series of algorithms as summarized in Table 1.

Activity frames can also provide a natural representation for input from sensors. For example, pyroelectric motion sensors respond to activity within a convex spatial region that can be projected onto the image plane and conservatively bounded by a rectangular activity frame. The same is true for optical beam sensors, pressure pads, and directional microphones.

For example, consider a set of commercial pyroelectric motion sensors configured to detect animal motion (eg. motion of warm bodies > 50 lbs). Each sensor has an associated field of view and responds with different quantitative

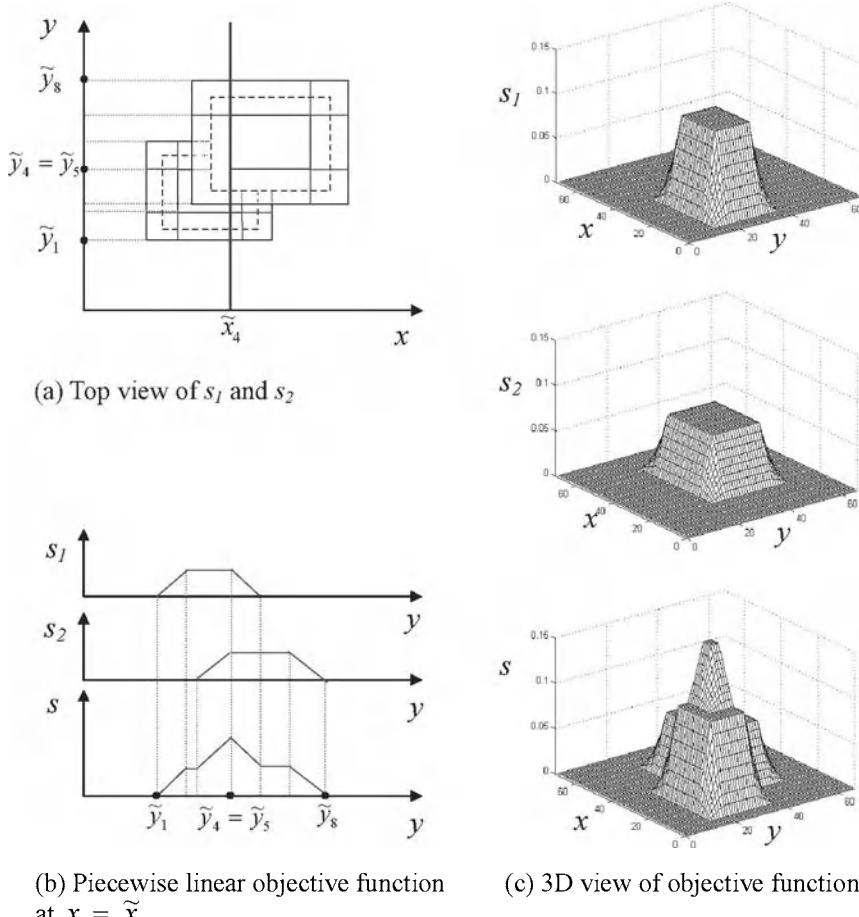


Fig. 5. Shape of reward metric for a fixed camera zoom level. For each user, their specified activity frame gives rise to an objective function that is plateau-like as illustrated in (c). The function consists of 5 planar and 4 quadratic surfaces at the corners. The overall objective function is the summation of plateaus generated by activity frames from all users.

levels based on mass and velocity of movement. When several sensors go off simultaneously, a series of camera positions may be selected as proposed above. It is also important not to “starve” any sensor that may indicate a crucial observation. Similar “starve” effect can also happen to a minority user, whose observing interests may be different from the majority.

We can augment the frame selection model in Equation 2 by introducing time variable t and, for each sensor, a linear gain function ω_i . The gain is a function of camera motion history, sensor reliability, and scientific observation priorities.

Table 1. Algorithms developed for Collaborative Frame Selection, where n is number of activity frames specified, and m is the number of camera zoom levels.

No.	Type	Zoom	Request	Solution	Complexity	Pub.
1	Centralized	m levels	Rectangle	Exact	$O(mn^2)$	[31]
2	Centralized	m levels	Rectangle	Approximation	$O(mn^{3/2} \log n)$	[16]
3	Distributed	m levels	Rectangle	Exact	Server: $O(mn)$ Client: $O(n)$	[32]
4	Centralized	Continuous	Rectangle	Exact	$O(n^3)$	[30]
5	Centralized	Continuous	Polygon	Approximation	$O((n + 1/\epsilon^3) \log^2 n)$	[30]
6	Distributed	Continuous	Polygon	Approximation	Server: $O(n)$ Client $O(1/\epsilon^3)$	[29]

$$\max_{c(t)} \sum_{i=1}^n \omega_i(t) s_i(r_i(t), c(t)) = \sum_{i=1}^n \omega_i(t) \frac{\text{Area}(r_i(t) \cap c(t))}{\text{Area}(r_i(t))} \min\left(\frac{z_i(t)}{z(t)}, 1\right). \quad (3)$$

We propose a gain function based on camera history as follows. We define a “dissatisfaction” value for each user (in this case each sensor) based on how poorly the last camera frame was aligned with the sensor’s last activity frame request: $u_i(t) = 1 - s_i(r_i(t), c(t))$. This “dissatisfaction” gain can accumulate over time: $\omega_i(t) = \sum_{k=0}^{t-1} \frac{u_i(k)}{2^{t-1-k}}$, so that when other sensors are satisfied with consistent camera motion, the neglected sensor gradually gains in influence. This can be defined in a recursive format,

$$\omega_i(t) = u_i(t-1) + \omega_i(t-1)/2.$$

Effectively, the weight of the un-observed region will increase until it is observed. Preliminary experiments suggest that this approach is robust, insuring that all sensors contribute and preventing the system from having observation driven by only a small number of dominating sensors (or users!).

5 Conclusion and Future Work

This paper reviews a series of prototype networked robotic systems and associated algorithms for collaborative observation.

We are currently extending our framework to consider resource limited observation, heterogenous user groups, optimizing camera trajectory, temporal modeling, sensor modeling, sensor monitoring and fault detection, and robotic actuation. We will develop automated agents based on sensors, robotic calibration for rapid deployment, and a video database for archiving, indexing, and query of observed scientific data.

Acknowledgments

This work was funded in part by National Science Foundation IIS-0534848/0535218, by Intel, Adept, Panasonic, Qualcomm, and Microsoft Corporations, and UC Berkeley's Center for Information Technology Research in the Interest of Society (CITRIS). Thanks to Ni Qin, Zane Goodwin, Dana Plautz, Frank van der Stappen, Steven Gentner, Carl Sutter, Jeff Wiegley, Michael Mascha, and Nick Rothenberg. Joseph Santarromana, George Bekey, Steven Gentner, Rosemary Morris Carl Sutter, Jeff Wiegley, Erich Berger, Thomas Steindl, Howard Moraff, John Canny, Eric Paulos, Judith Donath, Bob Farzin, Eric Mankin, Peter Lunenfeld, Roland Siegwart, Ruzena Bajcsy, Paul Wright, and Anthony Levandowski.

References

1. R. D. Ballard. A last long look at titanic. *National Geographic*, 170(6), December 1986.
2. A. Bejczy, G. Bekey, R. Taylor, and S. Rovetta. A research methodology for tele-surgery with time delays. In *First International Symposium on Medical Robotics and Computer Assisted Surgery*, Sept. 1994.
3. A. K. Bejczy. Sensors, controls, and man-machine interface for advanced tele-operation. *Science*, 208(4450), 1980.
4. Tim Berners-Lee, Robert Cailliau, Jean-Francios Groff, and Bernd Pollerman. World-wide web: The information universe. *Electronic Networking:Research, Applications and Policy*, 1(2), Westport CT, Spring 1992.
5. Barney Dalton and Ken Taylor. A framework for internet robotics. In *IEEE International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots*, Victoria, Canada, 1998.
6. Matthew Gertz, David Stewart, and Pradeep Khosla. A human-machine interface for distributed virtual laboratories. *IEEE Robotics and Automation Magazine*, December 1994.
7. Raymond Goertz and R. Thompson. Electronically controlled manipulator. *Nucleonics*, 1954.
8. K. Goldberg and B. Chen. Collaborative control of robot motion: Robustness to error. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 655–660, October 2001.
9. K. Goldberg, B. Chen, R. Solomon, S. Bui, B. Farzin, J. Heitler, D. Poon, and G. Smith. Collaborative teleoperation via the internet. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 2019–2024, April 2000.
10. K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and Jeff Wiegley. Beyond the web: Manipulating the physical world via the www. *Computer Networks and ISDN Systems Journal*, 28(1), December 1995. Archives can be viewed at <http://www.usc.edu/dept/raiders/>.
11. K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and Jeff Wiegley. Robot teleoperation via www. In *International Conference on Robotics and Automation*. IEEE, May 1995.

12. K. Goldberg and R. Siegwart, editors. *Beyond Webcams: An Introduction to Online Robots*. MIT Press, 2002.
13. K. Goldberg, D. Song, Y. Khor, D. Pescovitz, A. Levandowski, J. Himmelstein, J. Shih, A. Ho, E. Paulos, and J. Donath. Collaborative online teleoperation with spatial dynamic voting and a human “tele-actor”. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1179–84, May 2002.
14. K. Goldberg, D. Song, and A. Levandowski. Collaborative teleoperation using networked spatial dynamic voting. *The Proceedings of The IEEE*, 91(3):430–439, March 2003.
15. K. Han, Y. Kim, J. Kim, and S. Hsia. Internet control of personal robot between kaist and uc davis. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.
16. Sariel Har-Peled, Vladlen Koltun, Dezhen Song, and Ken Goldberg. Efficient algorithms for shared camera control. In *19th ACM Symposium on Computational Geometry, San Diego, CA*, June 2003.
17. <http://telerobot.mech.uwa.edu.au/>.
18. H. Hu, L. Yu, P. W. Tsui, and Q. Zhou. Internet-based robotic systems for teleoperation. *Assembly Automation*, 21(2):143–151, May 2001.
19. S. Jia, Y. Hada, G. Ye, and K. Takase. Distributed telecare robotic systems using corba as a communication architecture. In *IEEE International Conference on Robotics and Automation (ICRA), Washington, DC, United States*, 2002.
20. S. Jia and K. Takase. A corba-based internet robotic system. *Advanced Robotics*, 15(6):663–673, Oct 2001.
21. J. Kim, B. Choi, S. Park, K. Kim, and S. Ko. Remote control system using real-time mpeg-4 streaming technology for mobile robot. In *IEEE International Conference on Consumer Electronics*, 2002.
22. R.C. Luo and T. M. Chen. Development of a multibehavior-based mobile robot for remote supervisory control through the internet. *IEEE/ASME Transactions on Mechatronics*, 5(4):376–385, 2000.
23. T. Mirfakhrai and S. Payandeh. A delay prediction approach for teleoperation over the internet. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.
24. R. S. Mosher. Industrial manipulators. *Scientific American*, 211(4), 1964.
25. L. Ngai, W.S. Newman, and V. Liberatore. An experiment in internet-based, human-assisted robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.
26. R. Safaric, M. Debevc, R. Parkin, and S. Uran. Telerobotics experiments via internet. *IEEE Transactions on Industrial Electronics*, 48(2):424–31, April 2001.
27. T. Sato, J. Ichikawa, M. Mitsuishi, and Y. Hatamura. A new micro-teleoperation system employing a hand-held force feedback pencil. In *International Conference on Robotics and Automation*. IEEE, May 1994.
28. Thomas B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992.
29. D. Song. *Systems and Algorithms for Collaborative Teleoperation*. PhD thesis, Department of Industrial Engineering and Operations Research, University of California, Berkeley, 2004.
30. D. Song, A. Pashkevich, and K. Goldberg. Sharecam part II: Approximate and distributed algorithms for a collaboratively controlled robotic webcam. In *IEEE/RSJ International Conference on Intelligent Robots (IROS)*, Nov. 2003.

31. D. Song, A. F. van der Stappen, and K. Goldberg. Exact and distributed algorithms for collaborative camera control. In J.-D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, editors, *Algorithmic Foundations of Robotics V, Springer Tracts in Advanced Robotics 7*, pages 167–183. Springer-Verlag Berlin Heidelberg, Dec. 2002.
32. D. Song, A.F. van der Stappen, and K. Goldberg. An exact algorithm optimizing coverage-resolution for automated satellite frame selection. In *IEEE International Conference on Robotics and Automation (ICRA 2004), New Orleans, US*, May 2004.
33. R. Tomovic. On man-machine control. *Automatica*, 5, 1969.

Session Overview

Interfaces and Interaction

Makoto Kaneko¹ and Hiroshi Ishiguro²

¹ Graduate School of Engineering, Hiroshima University

² Graduate School of Engineering, Osaka University

The main focus on this section is “Interfaces and Interaction”. Computer display is an interface showing the information to human visually. For providing with information from human to computer (or robots), there are many interface devices, such as force sensor, acceleration sensor, velocity sensor, position sensor, tactile sensor, vision and so forth. EMG and EEG signals are also utilized as an interface signal from handicapped people to robots. By utilizing these interfaces, interactive motion between human and robot can be achieved. Interaction with variety is extremely important for entertainment robots, amusement robots, and social robots. Since the capability of these robots strongly depends upon the reaction and the expression, both sensors and actuators are key components for advancing them. Three papers are presented in this section. The first is concerned with haptic based communication between human and robots. The second deals with the vestibular sensor that can detect head motion of human. The final paper deals with diagnosing autism through the interaction between human and robot. While these three papers are largely unrelated to each other in the purpose, the common key word is interaction between human and robot. Especially, in the first and the third paper, the interaction between human and robot is strongly intended.

In the paper “Haptic Communication between Humans and Robots,” the authors present a detailed design of tactile sensor and utilization of haptic data for a robot to estimate human position and posture. They are eventually interested in obtaining a map describing the relationship between the tactile information and human positions/postures from the records of haptic interaction taken by tactile sensors and a motion capturing system composed of plural number of cameras installed in the environment. They first develop tactile sensors where each sensor element is film-type piezoelectric polymer sensors inserted between the thin and thick silicone rubber layers. The film-type sensor, consisting of polyvinylidene fluoride (PVDF) and sputtered silver, outputs a high voltage proportional to the pressure applied. The tactile sensor covers the entire robot body, so that any contact between human and robot can be detected. A couple of makers are attached to human body, so

that the motion capturing system can take human positions/postures. At the same time, tactile information is captured so that the correlation map between tactile and visual information can be generated. Once the map is obtained, the robot can estimate the position and posture of human by using tactile information.

In the second paper “A Vestibular Interface for Natural Control of Steering in the Locomotion of Robotic Artifacts: Preliminary Experiments,” the authors develop a novel interface capable of detecting human’s motion intention obtained from anticipatory movements that naturally accompany more complex motor behaviors. To validate the idea, they develop a prototype vestibular interface that can detect head linear accelerations and angular velocities along three axes. In order to investigate the principle of a vestibular interface based on head anticipatory movements, head motions with the interface are compared with the actions on traditional input interface, during driving tasks. Preliminary experiments are executed to confirm whether the head anticipatory movements associated with steering is observed or not. An interesting observation is that head motion always is slightly in advance compared with steering command time when subjects are executing a driving game. The similar tendency is also observed in right-left steering in case of a mobile robot.

The paper “How Social Robots will Help Us to Diagnose, Treat, and Understand Autism” intends to diagnose and understand autism through the interaction between a candidate (or a patient) and social robots. The ESRA robot with the capability of three facial expressions is used with a playtest button. The robot has no sensory capabilities and does not respond to anything that the child does. ESRA is programmed to perform a short script with both a set of actions and an accompanying audio file. The robot performs behaviors from this same repertoire with the initiation of these behaviors triggered by an experimenter. Even with the extremely limited capabilities of ESRA, the children seemed to thoroughly enjoy the session. An interesting observation is that while children are universally engaged with the robot and often spend the majority of the session touching the robot, vocalizing at the robot, and smiling at the robot, these positive proto-social behaviors are rarely seen for the children with autism. The authors point out that quantitative and objective evaluation for diagnosis can be accomplished through both passive observation of the child and active interactions with robots. Chasing gaze direction during a picture observing is a good example of passive sensing and observation during interaction with robot by using the playtest button is a good example of the latter case.

Haptic Communication Between Humans and Robots

Takahiro Miyashita¹, Taichi Tajika^{1,2}, Hiroshi Ishiguro^{1,2}, Kiyoshi Kogure¹,
and Norihiro Hagita¹

¹ Intelligent Robotics and Communication Labs., ATR, Kyoto, JAPAN
`{miyashita,kogure,hagita}@atr.jp`

² Dept. of Adaptive Machine Systems, Osaka University, Osaka, JAPAN
`{Taichi.TAJIKA,ishiguro}@ams.eng.osaka-u.ac.jp`

Summary. This paper introduces the haptic communication robots we developed and proposes a method for detecting human positions and postures based on haptic interaction between humanoid robots and humans. We have developed two types of humanoid robots that have tactile sensors embedded in a soft skin that covers the robot's entire body as tools for studying haptic communication. Tactile sensation could be used to detect a communication partner's position and posture even if the vision sensor did not observe the person. In the proposed method, the robot obtains a map that statistically describes relationships between its tactile information and human positions/postures from the records of haptic interaction taken by tactile sensors and a motion capturing system during communication. The robot can then estimate its communication partner's position/posture based on the tactile sensor outputs and the map. To verify the method's performance, we implemented it in the haptic communication robot. Results of experiments show that the robot can estimate a communication partner's position/posture statistically.

1 Introduction

Haptic communication is as important as vision and voice. It allows blind people to acquire a certain autonomy in their everyday life, since it is largely redundant with vision for the acquisition of spatial knowledge of the environment and object properties [1]. Moreover, people who are familiar with each other often pat each other's back or hug each other; such haptic interaction reinforces their familiarity.

If a communication robot equipped with tactile sensors over its entire body could have the same capability of haptic interaction as human do, we would feel greater familiarity with the robot, thus shortening its communicative distance from people. There has been much research on developing tactile sensors that cover the entire body of a robot [2, 3, 4, 5, 6]. Pan et al. [2] and Inaba et al. [4] proposed tactile sensor suits made of electrically conductive fabric.

Hakozaki et al. [5] proposed telemetric robot skin based on sensor chips that consist of an LC resonance circuit. In particular, Inaba et al. developed a full-body sensor suit to detect binary touching information for their remote-brained small humanoid robot. Iwata et al. [3] also proposed force-detectable surface-cover systems for humanoid robots and developed an actual humanoid robot, named WENDY, with the systems. Their systems are based on a six-axis force torque sensor and force sensing registers (FSR sensors) used to measure the external force vector and contact positions on the cover accurately. Regarding haptic communication, Naya et al. collected data of tactile information from a pet-like robot and proposed a method that could classify several human touching motions based on the tactile sensor values [7]. By using that method, a robot can classify human touching motion and establish a relationship with a person by giving appropriate responses to the person.

Let us consider some other aspects of haptic interaction. An infant is hugged by or plays with a caretaker. During that interaction, the caretaker acquires the infant's body geometry information in order to carefully control his or her motions. People often pat a communication partner on his/her body instead of calling him/her. In this case, since the partner is able to easily turn his/her face to the patting person directly, the partner can roughly estimate the position and the posture of the patting person without looking. This estimation makes human haptic interaction natural and safe. If we could realize such estimation for humanoid robots, the haptic interaction between humans and the robots would thus become more natural and safer. The previous researches, however, have focused on sensing the contact locations on the robot, and no method has been proposed to estimate position and posture by using only tactile information. In the field of computer vision, several methods have been developed to estimate position and posture [8, 9]. Under the situation of haptic interaction between a human and a robot, however, the distance between the human and the robot will be short, and images taken from the robot's cameras will only include a part of the human's body. Thus, it is difficult to use these methods for haptic interaction.

This paper proposes a method for a robot to detect human positions and postures by using tactile sensor data while the person is touching the robot. The key idea for handling tactile information is that the possible combinations of tactile information and human position/posture are quite limited in the above situations. In this method, the robot acquires a map that describes the correspondences between the tactile information and human positions/postures from the records of haptic interaction taken in situations of communication with humans. Using the map, it is possible to estimate position and posture based only on the information provided from the tactile sensors. We demonstrate the validity of the method in experiments on a robot covered with tactile sensors.

2 Tactile Sensors Covering an Entire Robot Body

2.1 Robovie-IIS

This section introduces the architecture of the tactile communication robot named Robovie-IIS. We have been developing communication robots, each named Robovie, for the study of communication between individual humans as well as between humans and robots. Robovie-IIS is designed to study tactile communication used in friendly relationships. This robot is based on Robovie-II [10], with tactile sensor elements embedded in a soft skin that covers the robot's entire body. Figure 1 shows overall views of two types of Robovie-IIS and scenes of its communication with a human.

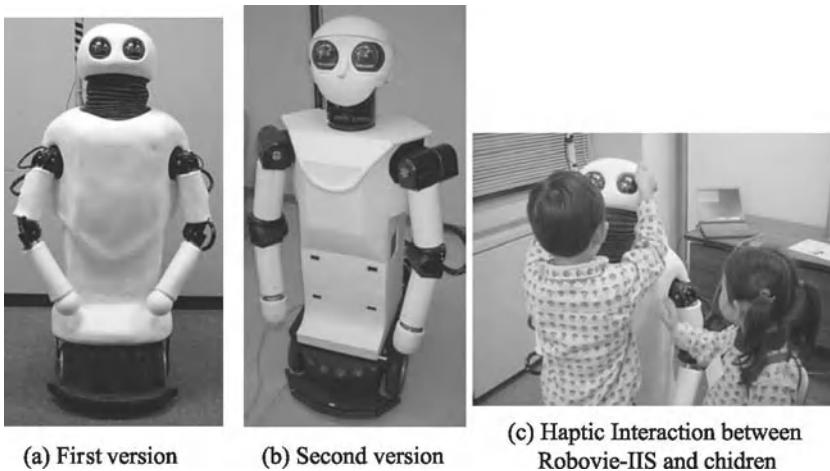


Fig. 1. Two types of tactile communication robot “Robovie-IIS”

2.2 Tactile Sensor Elements Embedded in Soft Skin

Figure 2 shows the hardware architecture of a tactile sensor element embedded in the soft skin. As the figure clearly illustrates, the soft skin consists of three layers. The outside layer is made of thin silicone rubber (thickness: 5 mm), and the middle layer is made of thick silicone rubber (thickness: 10 mm). We use these silicone rubber layers to achieve human-like softness. Moreover, the sense of touch and friction of the surface of the silicone rubber are similar to that of human skin. The thickness of the silicone rubber also absorbs the physical noise made by the robot's actuators. The inner layer is made of urethane foam, which insulates against heat from inside the robot and has a different surface friction from human skin. Its density is lower than that of the silicone rubber; the densities of the urethane foam and the silicone rubber are 0.03 g/cm³ and

1.1 g/cm³, respectively. The total density of the soft skin consisting of all layers is 0.6 g/cm³. The tactile sensor elements are film-type piezoelectric polymer sensors inserted between the thin and thick silicone rubber layers. The film-type sensor, consisting of polyvinylidene fluoride (PVDF) and sputtered silver, outputs a high voltage proportionate to changes in applied pressure. Figure 3 shows the arrangement of the sensor elements for the first type of Robovie-IIS, of which there are 48 in its soft skin. The second type of Robovie-IIS has 276 sensor elements in the skin.

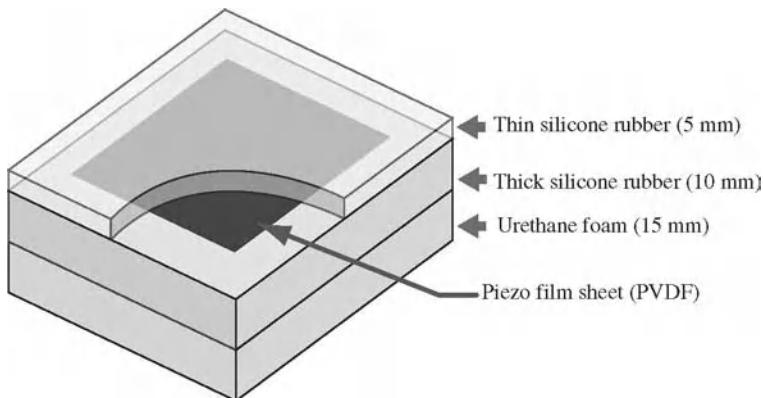


Fig. 2. Architecture of Soft Skin Sensor

Although the sensor element outputs a high voltage, the signal is weak compared to electric noise disturbance since its electric current is weak. Therefore, we distribute A/D converters (ADCs) with sensor-signal amplifiers to each body part. The ADCs are installed next to the sensor elements to convert the analog sensor signals to digital data. We developed two types of ADC, which are shown in Fig. 4. The dimensions of the first type of ADC are 23 × 137 × 8 mm. On this board, there are four A/D converters (each channel has six bits) to obtain the outputs of four sensor elements. We also use a microcomputer (PIC) to convert the digital data to a serial signal (RS-232c). By adding other boards' serial signals to it, we can realize a daisy-chain connection between the boards, as shown in Fig. 5 (a). These boards allow us to sense all sensor elements embedded in the soft skin from a serial port of the host computer.

As for the second type of ADC, its dimensions are 22 × 76 × 10.2 mm. This board has 16 A/D converters (each channel has 16 bits) and a micro-processor (SH2, Renesas Technology Corp.). We can connect 16 sensor elements to the board and preprocess the raw data of tactile sensor outputs, such as low-pass-filtering on the processor. The preprocessed data are converted to serial signals and sent to the host computer via a serial bus (RS-485), as shown in Fig. 5 (b).

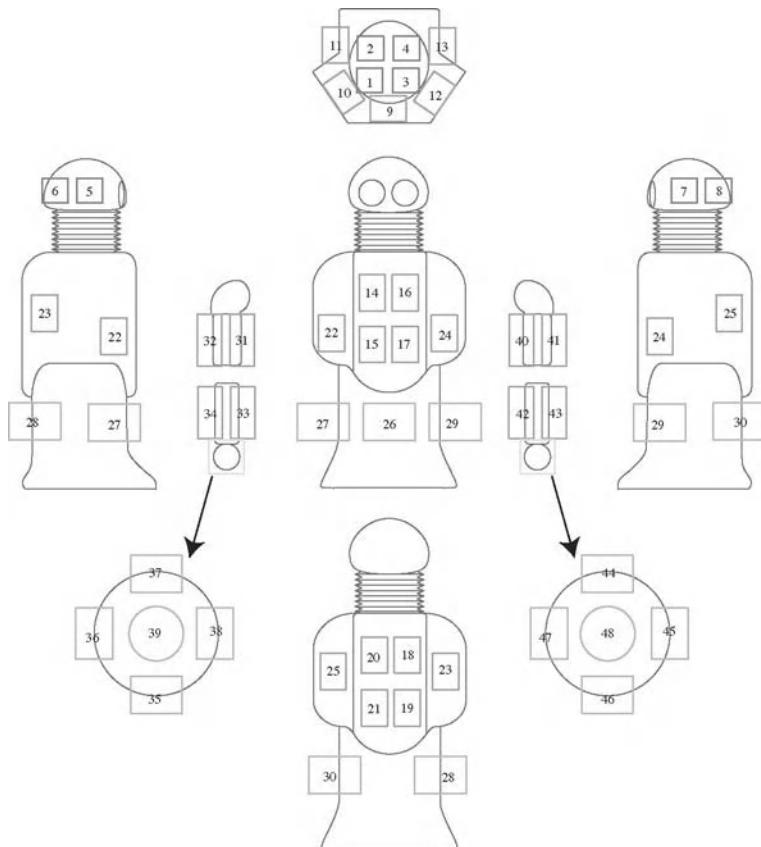


Fig. 3. Arrangement of sensor elements on Robovie-IIIs

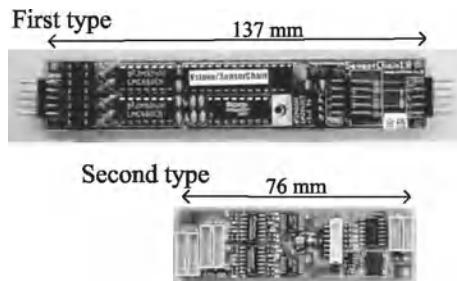


Fig. 4. Distributed A/D Converters

3 Human Position and Posture Detection

In this section, we describe a method to estimate human position and posture from the tactile sensor outputs.

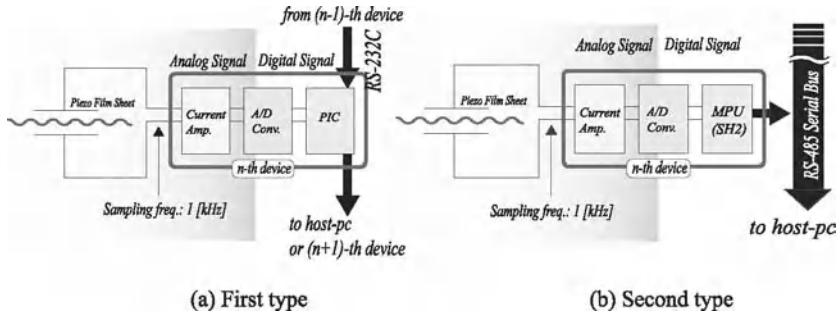


Fig. 5. Architecture of A/D Converters

3.1 Measuring Position and Posture of Humans

We employed an optical motion capturing system (VICON, Vicon Motion Systems Ltd.) to measure body movements. The motion capturing system consists of 12 pairs of infrared cameras and infrared lights and markers that reflect infrared signals. These cameras were set around the environment of the experiment as shown in Figure 6 (a). The system calculates each marker's 3-D position from all of the camera images, and it features high resolution in both time (60 Hz) and space (accuracy is 1 mm in the room). In this paper, we use three markers to describe the position and the posture of humans with respect to the coordinates fixed to the robot. These markers are attached to the waist and the left and right fingertips of the human.

3.2 Mapping Between Tactile Sensor Outputs and Probability Distribution of Human Positions/Postures

We calculate probability distributions of positions/postures of humans that correspond to tactile sensor outputs and build a map between them. The mapping process is carried out as follows. Figure 6 shows an outline of the process.

- (i) Record time series data of the robot's tactile sensor outputs and positions/postures of subjects simultaneously while they communicate with each other. In this paper, we used 46 tactile sensors along with three markers that were attached to the waist and both hands of the subject for the motion capture system as positions/postures of the subject. Hence, the tactile sensor outputs and the marker positions of the waist, the left hand, and the right hand are described as $\mathbf{t}_i \in \mathbb{R}^{46}$, \mathbf{p}_i^{waist} , \mathbf{p}_i^{L-hand} , $\mathbf{p}_i^{R-hand} \in \mathbb{R}^3$, respectively, where i denotes time.
- (ii) From all tactile sensor data $\{\mathbf{t}_i\}$, select the tactile sensor data $\{\mathbf{t}_j\}$ that are greater than a threshold for use while touching. The threshold is determined by preliminary experiments.

- (iii) Classify selected data $\{t_j\}$ into typical clusters $\{C_k\}$ by using the ISO-DATA clustering method [12].
- (iv) Calculate distributions of marker positions $\{{}^k p_j^*\}$ that correspond to the tactile sensor data $\{{}^k t_j\}$ at each cluster C_k by the following steps.
 - a) Classify the marker position data $\{{}^k p_j^*\}$ into clusters $\{{}^k D_l^*\}$ by using the ISO-DATA. We first classify the waist marker position data $\{{}^k p_j^{waist}\}$ into clusters $\{{}^k D_l^{waist}\}$. At each ${}^k D_l^{waist}$, we assume that the distribution of the marker position data conforms to a normal distribution $N(\mu, \sigma^2)$. Under this assumption, we calculate a mean μ and a standard deviation σ of $\{{}^k p_j^{waist}\}$, which are the elements of the cluster ${}^k D_l^{waist}$.
 - b) Calculate probabilities for the existence of the marker position at each cluster $\{{}^k D_l^*\}$ when the tactile sensor data belong to the cluster C_k . If the number of the elements $\{{}^k p_j^{waist}\}$ is m , and the total number of the waist marker positions that correspond to the tactile sensor outputs in the cluster C_k is n , we obtain the probability, $P_{k D_l^{waist}}$, as $\frac{m}{n}$.
 - c) Label the cluster ${}^k D_l^*$ effective if P_{D_l} becomes greater than threshold t_p and σ becomes less than threshold t_σ ; t_p and t_σ are determined by preliminary experiments.
 - d) Iterate these steps from (iv)-a) to (iv)-c) for the data of the left- and right-hand marker positions, $\{{}^k p_j^{L-hand}\}$ and $\{{}^k p_j^{R-hand}\}$, respectively.
- (v) Label the cluster C_k effective if the clusters $\{{}^k D_l^*\}$ that corresponded to C_k have more than one effective cluster.

3.3 Using the map

Once the map is obtained, the robot can use it as follows.

- (i) Obtain tactile sensor data vector $t \in \mathbb{R}^{46}$ during communication with a human.
- (ii) Calculate the distance between t and each cluster $\{C_k\}$ in the map, and select the cluster C_s for which the distance is shortest. Abandon the estimation if the distance is longer than a threshold.
- (iii) Obtain the probability distributions of the waist, the left- and right-hand positions that correspond to C_s if the cluster is effective.

4 Experiment

4.1 Acquiring Human Position and Posture

Figure 6(a) illustrates the system used to acquire the data of tactile sensor outputs and positions of the markers. The tactile sensor outputs were recorded

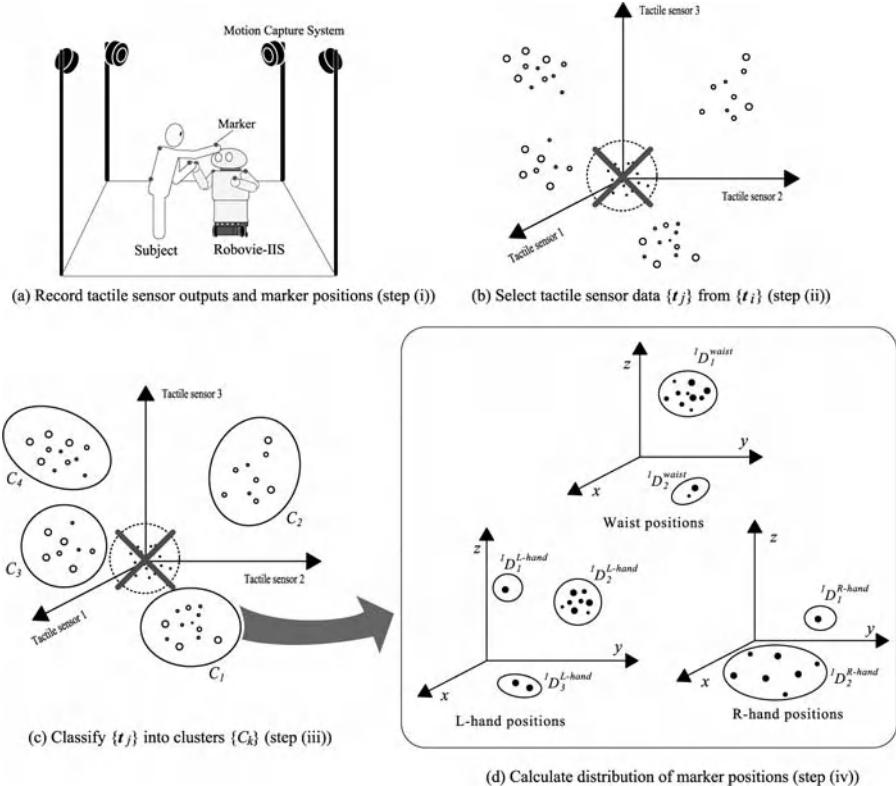


Fig. 6. Outline of Mapping process

on a hard-disk drive mounted on the robot's body. In this experiment, we used the first type of ADC described in Section 2.2 to obtain the outputs. The markers were attached to the waist and left/right fingertips of both the robot and a human subject. The motion capturing system was arranged to measure their motions representing haptic interaction. The sampling rate of the tactile sensor was 20 Hz, and the sampling rate of the positions of the markers was 60 Hz. In the experiments, Robovie-IIS continuously moved its joints, aside from its wheels, and communicated with the subject. The program used for its communication behavior was almost the same as that of Robovie-II [10], consisting of a behavior network based on situated modules that describe communication behaviors according to the situation. There are approximately 100 communication behaviors in Robovie's present behavior network.

The subjects of our experiment were 40 university students (males: 12, females: 28). An experimenter explained the purpose of the experiments as collecting haptic interaction data from the subjects and asked each of them to communicate with the robot for three minutes.

4.2 Results of Mapping Between Tactile Sensor Outputs and Probability Distribution of Human Positions/Postures

Table 1 shows the results of clustering the tactile sensor outputs and the evaluation of each cluster. The total number of data from the tactile sensor output, which was described as $t_i \in \mathbb{R}^{46}$ in section II-B, was 247,622. We used the first half of the data (123,811 data) for building the map between tactile sensor outputs and positions/postures of humans. The latter half of the data were used to verify the map.

First, we selected 14,265 touching data from the first-half data by employing the threshold of each tactile sensor. We then obtained 150 clusters using ISO-DATA. In this experiment, we set the threshold t_p to 0.1, t_σ for waist to 300 mm, and t_σ for both hands to 150 mm. Finally, we obtained 137 effective clusters for use in estimating human position and posture. Figure 7 describes in detail the number of effective clusters in a Venn diagram. We obtained 110 clusters for the waist position estimation, 90 clusters for left-hand position estimation, and 88 clusters for right-hand position estimation. As the figure shows, the robot can also estimate all positions, i.e. waist and both hand positions, from 54 clusters.

Table 1. Results of Clustering and Evaluation of each cluster

total # of skin sensor data	123,811
# of touching data	14,265
total # of clusters	150
# of effective clusters	137

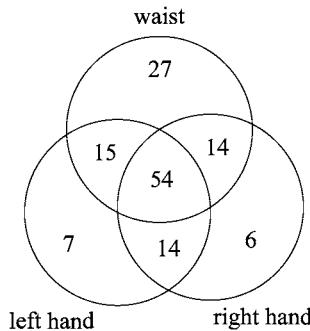


Fig. 7. Venn diagram of effective clusters

To verify the performance of the map, we used tactile sensor outputs of the latter-half data (123,811 data) as inputs of the robot and compared the estimation results of marker positions and the actual positions taken from the

motion capturing system. In this paper, we decided that the estimation would be successful if the actual marker position fell within the area from $\mu - 2\sigma$ to $\mu + 2\sigma$ at the estimated distribution conforming to a normal distribution, $N(\mu, \sigma^2)$. We obtained 14,314 touching data from the latter-half data, and there were 12,711 data (89%) that belonged to the tactile sensor cluster in the map. Success rates of the estimations for the waist, the left hand, and the right hand were 87%, 63%, and 72%, respectively.

To verify the effectiveness of the estimation based on the map, we applied reflexive behaviors to the robot so that it would look at the subject's face based only on the tactile sensor outputs and the map. This behavior is difficult to achieve for robots that do not have such a map. The photographs in Figs. 8 (a) and (b) show these reflexive behaviors. In these figures, the bar charts denote the tactile sensor outputs obtained during the haptic interaction shown in the photographs. The figures of the robot model show the distributions of waist and hand positions that correspond to the bar chart. As can be seen in these figures, the robot is able to estimate the positions of waist and hands statistically as information on human position and posture. The robot can look at the subject's face by utilizing the tactile sensor outputs, as the photographs indicate.

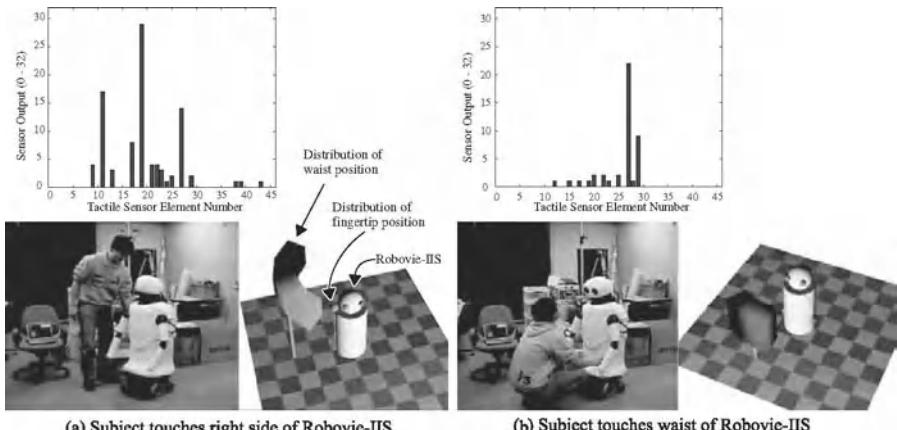


Fig. 8. Estimation results of human position and posture

5 Discussion and Conclusion

We proposed a method to estimate human position and posture by utilizing tactile information. In our method, the robot first acquires a relationship between its tactile information and human positions and postures from the

history of haptic interaction. In our experiments, we obtained the success rates of the estimation for the waist, the left hand, and the right hand as marker positions. The success rate for the left hand turned out to be the lowest because almost all of the subjects were right-handed persons. They used mainly their right hand for touching the robot. Thus the position of the left hand became unstable while touching with the right hand. If the robot obtained more data of haptic interaction with left-handed persons, the success rate of the estimation for the left hand would increase. This implies that the success rates depend on the robot's experiences of haptic communications.

In this paper, we used the communication partner's position and posture based on a 3-D motion capture system. If the robot could sense more information from the partner by accessing its passive-type sensors and correlating their data to tactile information, it would estimate the partner's state more precisely based only on the tactile information. In future work, we will use the information described above to estimate the partner's state more precisely.

Acknowledgment

This research was supported by the Ministry of Internal Affairs and Communications.

References

1. Hatwell Y, Streri A, and Gentaz E (eds) (2003) Touching for Knowing. Advances in Consciousness Research, Vol. 53. John Benjamins Publishing Company, Amsterdam The Netherlands
2. Pan Z, Cui H, and Zhu Z (2003) A Flexible Full-body Tactile Sensor of Low Cost and Minimal Connections. In: Proc. 2003 IEEE International Conference on Systems, Man, and Cybernetics (SMC'03) 3:2368–2373
3. Iwata H, Hoshino H, Morita T, and Sugano S (2001) Force Detectable Surface Covers for Humanoid Robots. In: Proc. 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM'01) 1205–1210
4. Inaba M, Hoshino Y, Nagasaka K, Ninomiya T, Kagami S and Inoue H (1996) A Full-Body Tactile Sensor Suit Using Electrically Conductive Fabric and Strings. In: Proc. 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 96) 2:450–457
5. Hakozaki M, Oasa H, and Shinoda H (1999) Telemetric Robot Skin. In: Proc. 1999 IEEE International Conference on Robotics and Automation (ICRA 99) 2:957–961
6. Yamada Y, Iwanaga Y, Fukunaga M, Fujimoto N, Ohta E, Morizono T, and Umetani Y (1999) Soft Viscoelastic Robot Skin Capable of Accurately Sensing Contact Location of Objects. In: Proc. 1999 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI'99) 105–110

7. Naya F, Yamato J, and Shinozawa K (1999) Recognizing Human 'Touching Behaviors using a Haptic Interface for a Pet-robot. In: Proc. 1999 IEEE International Conference on Systems, Man, and Cybernetics (SMC'99) II-1030–1034
8. Hongo H, Ohya M, Yasumoto M, Niwa Y, and Yamamoto K (2000) Focus of attention for face and hand gesture recognition using multiple cameras. In: Proc. 2000 IEEE International Conference on Automatic Face and Gesture Recognition 156–161
9. Sakagami Y, Watanabe R, Aoyama C, Matsunaga S, Higaki N, and Fujimura K (2002) The intelligent ASIMO; System overview and intergration. In: Proc. 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02) 2478–2483
10. Ishiguro H, Ono T, Imai M, Maeda T, Kanda T, and Nakatsu R (2001) Robovie: A robot generates episode chains in our daily life. In: Proc. of International Symposium on Robotics 1356–1361
11. Kandel E, Schwartz J, and Jessel T (eds) (2000) Principles of Neural Science, 4/e. McGraw-Hill, USA
12. Backer E (1995) Computer Assisted Reasoning in Cluster Analysis. Prentice-Hall, Englewood Cliffs, New Jersey USA

A Vestibular Interface for Natural Control of Steering in the Locomotion of Robotic Artifacts: Preliminary Experiments

Cecilia Laschi¹, Eliseo Stefano Maini¹, Francesco Patane^{'1}, Luca Ascari²,
Gaetano Ciaravella², Ulisse Bertocchi³, Cesare Stefanini³, Paolo Dario^{1,2},
and Alain Berthoz⁴

¹ ARTS Lab. Scuola Superiore Sant'Anna
Piazza Martiri della Libertà, 33 - 56127 Pisa, Italy cecilia@arts.sssup.it

² IMT Doctoral School in Biorobotics Science and Engineering
Via San Micheletto, 3 - 55100 Lucca, Italy

³ CRIM Lab. Scuola Superiore Sant'Anna
Piazza Martiri della Libertà, 33 - 56127 Pisa, Italy

⁴ CNRS UMR 7152 Collège de France
Place Marcelin Berthelot, 11 - 75231 Paris Cedex 05, France

This work addresses the problem of developing novel interfaces for robotic systems that can allow the most natural transmission of control commands and sensory information, in the two directions. A novel approach to the development of natural interfaces is based on the detection of the human's motion intention, instead of the movement itself, as in traditional interfaces. Based on recent findings in neuroscience, the intention can be detected from anticipatory movements that naturally accompany more complex motor behaviors.

This work is aimed at validating the hypothesis that head movements can be used to detect, slightly in advance, a person's intention to execute a steering during locomotion, and that a natural interface can be developed for controlling the navigation of a robotic artifact, based on this principle. A prototype 'vestibular' interface has been developed to this purpose, based on a 3-axial artificial vestibular system, developed by part of the authors for humanoid robotics applications. Three different experimental sessions have been carried out by using: (1) a driving video-game; (2) a robotic endoscope, with a 2-DOF steering tip; and (3) a mobile robot with a camera on-board.

The experiments showed that anticipatory head movements occur even when the person is driving a device, like those used in the experiments, and that such head movements always anticipate commands to the input device. The results indicate that the proposed hypothesis is valid and that a further research effort is worthwhile in the direction of using this novel principle to

develop natural interfaces, which in fact can be very useful in many tasks, with different devices.

1 Introduction

Robotics Technology is becoming more and more pervasive in human environments [1]. Robots are getting closer to human life in a variety of ways and shapes: not only as humanoids [2], but also as task-specific robotic tools, as smart robot appliances [3], and even as bionic robotic parts to be connected to the human brain and body [4]. This is in fact one of the front-edge challenges of robotics, which poses novel and critical problems not only in the design and development of human-like components, but also in the study and development of natural ways of interaction and interfacing between the natural body, especially the brain, and the robotic parts [5]. The main scientific problem in interfacing natural and robotic systems is to understand how the human brain can perceive the artificial parts as own parts and to what extent they can be controlled in a natural way by the brain. It is therefore crucial that the interfaces for bionic systems allow the most natural transmission of control commands and sensory information, in the two directions. This requires a novel approach and design method, which integrates multidisciplinary expertise and starts from models of human sensory-motor coordination for modeling and developing interfacing mechanisms that exploit them at the best, to obtain natural perception and control.

Traditional interfaces are based on user's motor actions, typically mapped onto a different geometry and kinematics, i.e. those of the input devices. Such cortical re-mapping between the motor areas involved in the use of the interface and those involved in the motor task at hand requires some learning and introduces an additional cognitive burden onto the users. Many authors suggest the adoption of multimodal devices to reduce the users' effort to communicate the intended commands, thus letting them more free to focus on the tasks and goals [6]. Furthermore, detecting the users' motor action on the input device and transmitting it to the robot introduce a delay from when the movement is planned in the human brain to when it is accomplished by the robot.

A more suitable approach to the development of interfaces is based on the detection of the human's motion intention. This can be detected as it originates in the brain, by means of brain-machine interfaces [7, 8] or when the control signal is transmitted in the nervous system to peripheral districts. On the other hand, it could also be suggested that anticipation of complex motor behaviour can be detected by recording *anticipatory movements*. In humans, simple movements anticipate, to some extents, other complex sensory-motor behaviors during postural and arm movements [9, 10, 11]. Such anticipatory movements may be used in a context-dependent manner for building natural and intuitive interfaces. The two main advantages of this approach are: (1)

the detected movements are naturally associated with motor behaviors and as such they would not put any additional cognitive burden on the person; (2) the detected movements occur well in advance of motor behaviors and therefore they would help obtain a timely reaction in the controlled robotic system.

This work is based on the demonstration that head movements can be used to detect, slightly in advance, a person's intention to execute a steering during locomotion. It has been shown that, during locomotion, the head anticipates the locomotor trajectory along simple trajectories like around a corner, and in more complex trajectories like triangles or circles or even more complex forms. This anticipation develops in the child after 4/5 years of age and can reach values as high as several hundred milliseconds. [22, 23, 24, 25, 26]

In our experiments, head movements are investigated to be used as a natural interface to control and to trigger steering in the navigation tasks performed by 3 different robotic artifacts. The experiments are aimed at identifying: (1) if head motion actually anticipates steering, even when driving different devices, instead of walking; (2) if the timely detection of head motion can be used to enhance the interface in driving.

2 Methods and Tools

2.1 Neuroscience Background

In many everyday activities, humans carry out more than one motor task simultaneously, even when the motor behavior appears relatively simple. Movement sequences, defined by both the component movements and the serial order in which they are produced, are the fundamental building blocks of the motor behavior. It is known that the serial order of sequence production is strongly encoded in medial motor areas even if understanding to what extent sequences are further elaborated or encoded in the primary motor cortex still remains controversial [12]. Over the last decades, several efforts were dedicated to understand how the central nervous system manages the serialization of movements and a consolidate finding is the existence of anticipatory movements that are likely to be acquired during developmental age [13]. Broadly speaking, anticipatory movements are motor activities that support the production of the main motor activity and that occur before likely sensory events. These movements are in contrast to reflexive actions and are necessary to compensate for delays present in sensory and motor systems. Smooth pursuit eye movements are often used as a paradigmatic example for the study of anticipation [14, 15]. Many authors have investigated various types of anticipation. For example, Land et al. [9] reported that during everyday activities, gaze fixations are always close to the object being manipulated, and very few fixations are irrelevant to the task occurred. Moreover, gaze arrives at the object

to be manipulated some 0.5 seconds before any indication of manipulation. Johansson et al. [10] demonstrated that gaze in manipulation tasks consistently fixates future object contact points well before the hand reaches these locations and anticipates reaching trajectory via-points. In a similar way, head movements are believed to anticipate body motions, such as turning while walking [11, 16]. It has been shown in these later papers that the head anticipation is in fact preceded, and probably coupled to, a gaze anticipation. Therefore head anticipation is an interesting measure when gaze recording is not available. Some neuroscientific bases that may explain the anticipatory triggering of orienting reactions may lie in the neural networks governing head direction in space during navigation. For instance, head direction neurons in the brain also show anticipatory activity [27] In this case, it is suggested that anticipatory orienting synergies belong to the behavioral repertoire of human navigation and may reflect the mental simulation of the desired trajectory or/and the need to prepare a stable reference frame for the intended action. In our work, we make use of some acquired findings of neuroscience research in order to provide a motivated novel approach to the design of innovative natural interfaces.

2.2 The Vestibular Interface

This work has been carried out by using a 3-axial artificial vestibular system, developed by the authors for humanoid robotics applications, to be mounted on anthropomorphic robotic heads [17]. The artificial vestibular system is inspired by the main functional characteristics of the human vestibular system and, in analogy with the latter one, it detects linear accelerations and angular velocities along 3 axes; to this purpose it integrates 1 tri-axial accelerometer and 3 mono-axial gyroscopes. All the electronic components are mounted on a single surface, thus limiting the total system dimension and weight and allowing a suitable mounting both on robotic and human heads. The sensors used for the design of the vestibular interface are: 2 GYROSTAR mono-axial Piezoelectric Vibrating Gyroscopes by muRata, 1 XV-3500CB mono-axial ultra small Vibration Gyro sensor by Epson and 1 H48C ultrasmall tri-axial accelerometer module by Hitachi, as shown in Fig. 1.

All the sensors used are extremely small and lightweight and provide movement information that are adequate to the requested application. The gyroscopes working principle is based on the detection of the Coriolis force, which is generated when a rotational angular velocity is applied to a vibrator inside the sensor. The Piezoelectric Vibrating Gyroscopes by muRata are used for the angular velocity detection around the Pitch and the Roll axis, while the mono-axial gyroscope XV-3500CB is employed for detecting the angular velocity around the Yaw axis. The latter device is a complete angular rate sensor with all of the required electronics on one chip and it has the particular feature of measuring the angular velocity around an axis orthogonal to

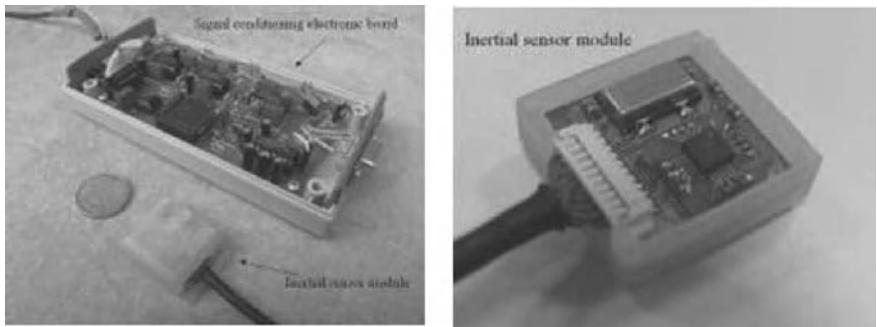


Fig. 1. Prototype of Artificial Vestibular System and signal conditioning electronic board

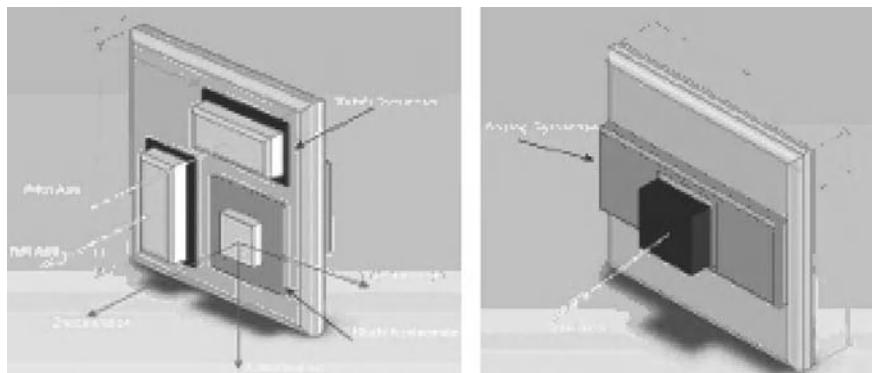


Fig. 2. CAD drawing of prototype artificial vestibular system: (left) top and (right) bottom view

its mounting surface. In this way, all the 3 gyroscopes can be integrated on a single plane, as shown in the CAD drawing (Fig.2).

The tri-axial accelerometer module H48C is composed of a MEMS technology sensor chip and of a CMOS-IC chip with the op-ampifiers. As for the 3 mono-axial gyroscopes even the tri-axial accelerometer can be placed on the same plane thus allowing a strong miniaturization of the total system.

The A/D conversion, amplification and filtering of the different signals are processed by an electronic board that was designed and developed purposively. This is composed of 14 operational amplifiers for the filtering and the amplification of the signals and by a 20 MHz PIC 16F877 microcontroller for the conversion of the signals from analog to digital. The board is connected to the PC by means of a standard RS-232 port using serial codification information.

All the channels are filtered by a high-pass filter, with a cut-off frequency of approximately 0.3 Hz in order to reduce the effect of temperature drift, while

a low-pass filter with a cut-off frequency of approximately 6 Hz, is used to suppress the output noise component. The filtered signals are then amplified with a two-stage operation amplifier, allowing to modulate the total gain of amplification according to the operating range of the specific application. Table 1 shows the ranges of total amplification gain for all the suitable sensor outputs and their corresponding values of full scale and resolution.

Table 1. Electronic board total amplification gain

	Total gain Min	Full Scale		Sensitivity	
		Max	Min	Max	Min
GYROSTAR 10	77	1 [rad/s]	5.2 [rad/s]	$9.7 \cdot 10^{-4}$ [rad/s]	$5.1 \cdot 10^{-3}$ [rad/s]
XV-3500CB	10	77	1 [rad/s]	4.5 [rad/s]	$9.7 \cdot 10^{-4}$ [rad/s]
H48C	1	4.2	1.5 g	3 g	$1.4 \cdot 10^{-3}$ g
					$2.9 \cdot 10^{-3}$ g

Moreover, a dedicated software, with a graphical user interface (GUI), has been developed for the further steps of signal processing and integration. Signal processing consists of two subsequent steps of elaboration: filtering and amplification. In the first step, in order to suppress high-frequency noise, a real-time fourth-order single-pass Butterworth low-pass filter, with a cut-off frequency of 6 Hz, is applied to all the sensor outputs (voltage measures). In the second phase, the voltage measures are amplified according to the sensor scale factor, converted into the respective physical values (angular velocities for the gyroscopes; accelerations for the accelerometers) and filtered again with the same filter.

In this application, the system calibration was performed according to existing procedures [18]. Finally, the static angles of the artificial vestibular system on the pitch and roll axes were calculated from the accelerometer outputs, while the dynamic angles were obtained from the gyroscopes outputs by means of numerical trapezoidal integration.

3 Experimental Validation

3.1 Experimental Methodology

In order to investigate the principle of a vestibular interface based on head anticipatory movements, experimental trials were set-up in which head motion were compared with the actions on a traditional input interface, during driving tasks. The experimental validation was organized in three sessions. The common set-up for the three experimental sessions consists of a number of subjects wearing the prototype vestibular interface, on top of their heads, and a binocular wearable display (I-Glasses by Video Pro 3D), for visual feedback. In all the sessions, the subjects were asked to perform a driving task,

but different devices were driven in the three sessions, with different input interfaces:

1. *driving video game*: a commercial video game was used in this session, where a car is driven by a using a gamepad by Logitech, along a rally circuit. This experiment was conceived in order to investigate the working hypothesis when the subjects are asked to drive a virtual artifact, and receive images of a simulated environment;
2. *robotic endoscope*: the navigation trials were performed by using a robotic endoscope with a 2-DOF steering tip and a bent tube simulating the exploration space. A joystick was used as input interface, and the image from the endoscope tip was sent to the subject as visual feedback. In this case, the device to be controlled is a real system, though the environment is not familiar for the subject, and so the feedback images;
3. *mobile robot*: a small mobile robotic platform with wheels and a camera on-board was set-up for this session. The subjects were asked to drive the mobile robot in a doorway passage task and the images from the on-board camera were fed back to the subjects. In this case, the robotic platform performed a real navigation in a real environment that was natural for human locomotion and familiar to the subjects.

3.2 Experimental Trials with the Driving Game

Eight subjects were involved in the experimental trials with the driving video game, and each of them was asked to perform three full laps in a circuit with 7 turns. The subjects were given a view of the circuit from inside the car and they could give the following commands: (1) steer right; (2) steer left; (3) speed up; (4) slow down.

The input interface, i.e. the gamepad, was modified so as to record the actions selected by the subjects. Specifically, the left and right steering commands were recorded. At the same time, the 6 signals generated by the vestibular interface worn by the subjects were recorded and synchronized with those coming from the gamepad by means of an audio trigger. The signals from the vestibular interface were compared with those from the input interface.

The signal corresponding to the angular velocity of the head during rotation (yaw axis) resulted to have a good correlation with the signals corresponding to right and left steering. Typical results are depicted in Fig.3 that shows the two signals as recorded in one of the trials.

If looking at the zero-crossing of the head velocity (i.e. the local minima of the head position) it is clear how they always anticipate a steering command. The time of anticipation is in average close to 0.5 sec. Also, the versus of the head rotation is coherent with the corresponding steering command, even if the amplitude of the two signals is not always proportional. Additional actions on the input interface can be observed, which are not anticipated by a head rotation. These usually correspond to adjustments of the car heading, especially after side-slips.

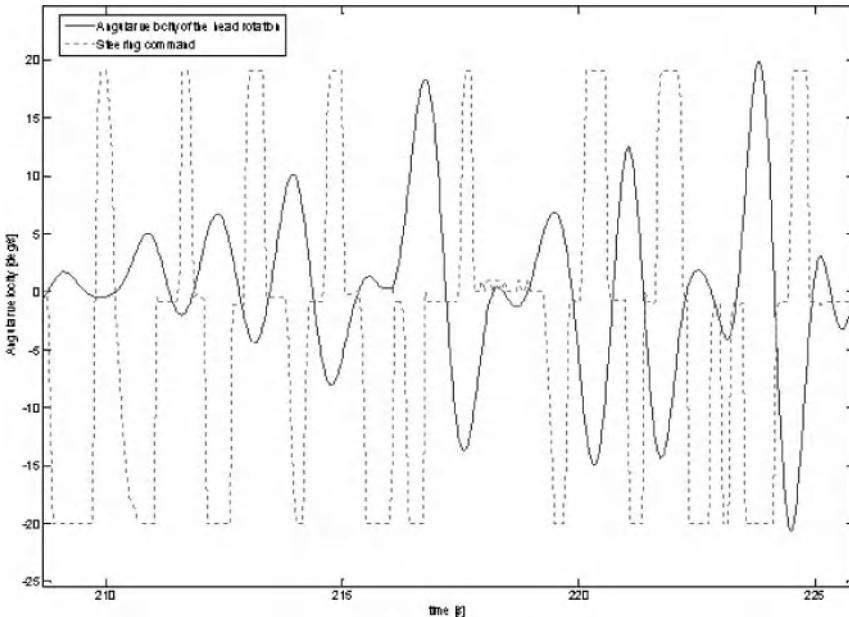


Fig. 3. Compared angular velocity of head rotation (blue solid line) and movement of the input device (red dotted line) in right-left steering, in the case of the driving game.

3.3 Experimental Trials with the Robotic Endoscope

The robotic endoscope used in the second experimental session was designed and developed for spinal neuroendoscopy, a minimally invasive technique aimed at exploring the subarachnoid space inside the spine. Due to the narrow dimensions of the lumen (i.e. the cavity filled with transparent fluid that is free from anatomical structures), its thickness ranges from 2 up to 8mm in humans, and to the presence of fragile nerve roots and blood vessels, unassisted manual neuroendoscopy is impossible in practice. For this reason, a robotic endoscope for neuroendoscopy has been proposed [19]. It consists of three main units: the end effector is a flexible multi-material multi-lumen catheter, whose external diameter is 2.7mm, housing 10 longitudinal channels: the endoscope (0.5mm of diameter, 6000 pixels of resolution) is hosted in the central one, while the steering capabilities are ensured by 3 pulling cables (hosted in three lateral channels), actuated by the Intelligent Drive Unit: it interprets the driving commands from the surgeon, tests their safety by means of a vision-based software module, the Cognitive Unit, and transfers them to a set of two stepper motors, pulling the steering cables. The Cognitive Unit processes all the information coming from the endoscopic camera and from the sensory system attached to the patient. It co-operates with the surgeon,

implementing a shared control strategy during the intervention, for example by preventing him from performing too dangerous maneuvers: a segmentation module recognizes fragile structures (e.g. blood vessels and nerve roots) in the image from the camera; a navigation module keeps trace of their position and structure even when they go out the field of view of the endoscope. A more detailed description of these two vision-related sub-systems was presented in [20]. The combined action of the Cognitive Unit and the Drive Unit is intended to overcome erroneous maneuvers of the surgeon, so as to ensure safe navigation.

The experimental task consisted in the navigation inside a curved white plastic tube, where the lumen appeared as a black circular spot in the endoscope image, which was fed back to the subjects through the wearable display. The subjects were asked to navigate towards the end of the tube, by keeping the black spot in the center of the image field. They used a joystick as input device. The signals coming from the vestibular interface and from the input joystick interface were synchronized and compared. Fig.4 shows the signal corresponding to the angular velocity of the head during rotation (along the yaw axis) and the signal corresponding to right-left steering, for one of the trials.

In this experimental session, the movements of the head during the experimental task were negligible. We envisage two main possible reasons explaining the lack of anticipatory movements of the head in this experimental scenario: first of all, the smaller dimensions of the image, as well as the narrow field of view of the endoscope are such that the subject could not have a real 'immersive' perception. It is also known that up to about 10/15 degrees the brain triggers mainly eye movements to orient in the horizontal plane [21] and therefore no head movements were needed in this restricted visual field. Secondarily, this task could have been perceived as a precision pointing task, i.e. heading towards the black spot of the lumen, rather than as a navigation task.

3.4 Experimental Trials with the Mobile Robot

A Pioneer I mobile robot by RWI was equipped with a digital video camera, as an experimental platform for the third session of experiments. A joystick was setup as the input interface for driving the robot. The image from the on-board camera was fed back to the subject through the wearable display, equipped with the vestibular interface, as in the other experimental sessions. The task asked to the subjects was to pass through a door, located at a distance of approximately 30 cm from the robot starting position, and to turn left in a corridor just after the doorway passage. A typical example of the task, with the real path and the superimposed velocity vectors during the path, is reported in Fig.5.

During the task, the robot was remotely operated by moving a mouse on a flat desk. The forward-backward movement of the mouse controlled the

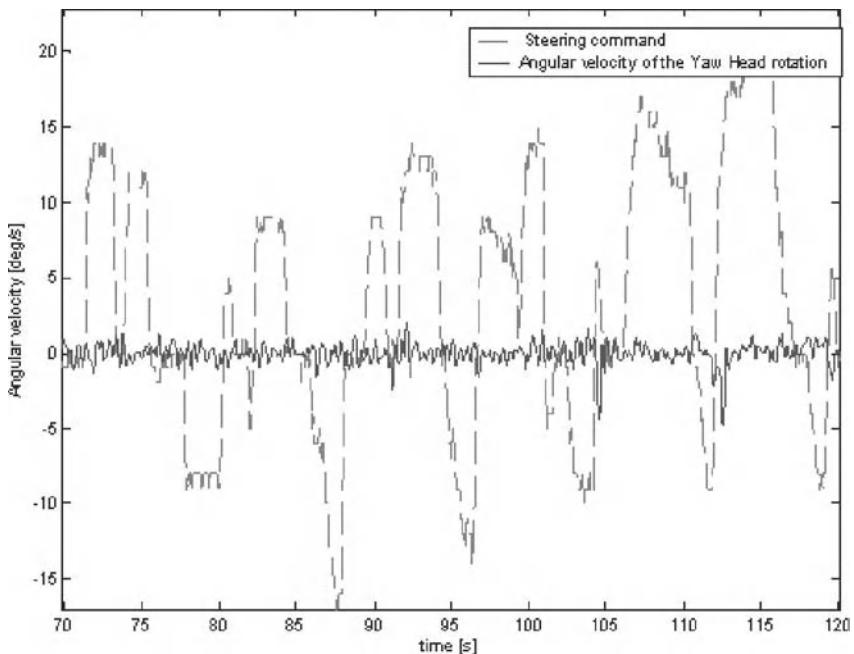


Fig. 4. Compared angular velocity of head rotation (blue solid line) and movement of the input device (red dotted line) in right-left steering, right-left steering, in the case of the robotic endoscope.

direction of motion whereas the amplitude of the movement controlled the velocity of the robot that was set between -400 mm/s (backward direction) and +400 mm/s (forward direction). A polynomial relation between mouse movements and robot velocity was implemented in order to avoid abrupt velocity variations possibly related to small movements of the input device

The steering of the robot was controlled using the left-right movements of the (joystick). In this case, in order to achieve a more reactive behavior of the robot, the central plateau of the previous control curve was avoided in favor of a sinus-like relation between the steering command and the steering velocity. Both curves are reported in Fig.6.

During each experiment, the odometric data of the robot (i.e. position, orientation and velocities) were recorded together with the data coming from the vestibular interface and from the input device. All the data were synchronized using an audio signal trigger.

The synchronized signals from the vestibular interface and from the input device are compared in Fig.7, for one of the experimental trials.

In this case, too, the signal corresponding to the angular velocity of the head in rotation (yaw axis) resulted to have the best correlations with the

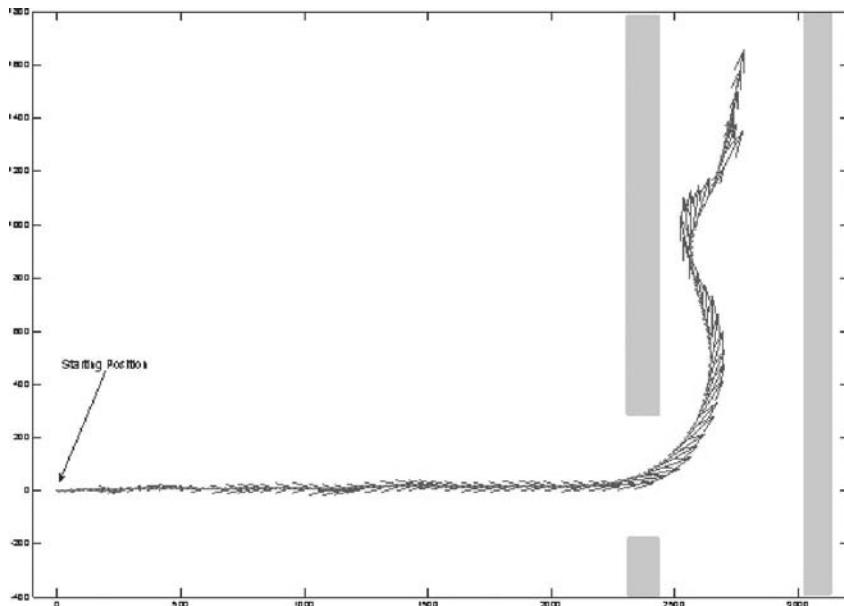


Fig. 5. Example of the robot path (*red dotted line*) with superimposed velocity vectors (*blue solid lines*) along the path.

signals corresponding to right-left steering. In this experiment the only remarkable event is the one occurring immediately after the passage of the door when a rapid steering command is issued to the robot in order to turn the corner. This happens approximately 7 to 8 seconds after the task starting. Then the steering command is kept constant for about 1 second and then another corrective steering is issued in order to adjust the alignment of the robot with the corridor. It is worth noticing that in both cases the steering command is anticipated by a coherent rapid movement of the head in the same direction that starts about 0.6 s before the steering command. Moreover, by looking at the zero-crossing of the head velocity it may be noticed that the head movement is almost completed at the very beginning of the steering command.

4 Results

This work has investigated for the first time the hypothesis that natural interfaces can be developed, based on anticipatory movements that are demonstrated to be involuntarily associated with more complex motor behaviors, in humans. This is in fact a novel principle for natural interfaces, deriving from a joint investigation by roboticists and neuroscientists, which integrates mul-

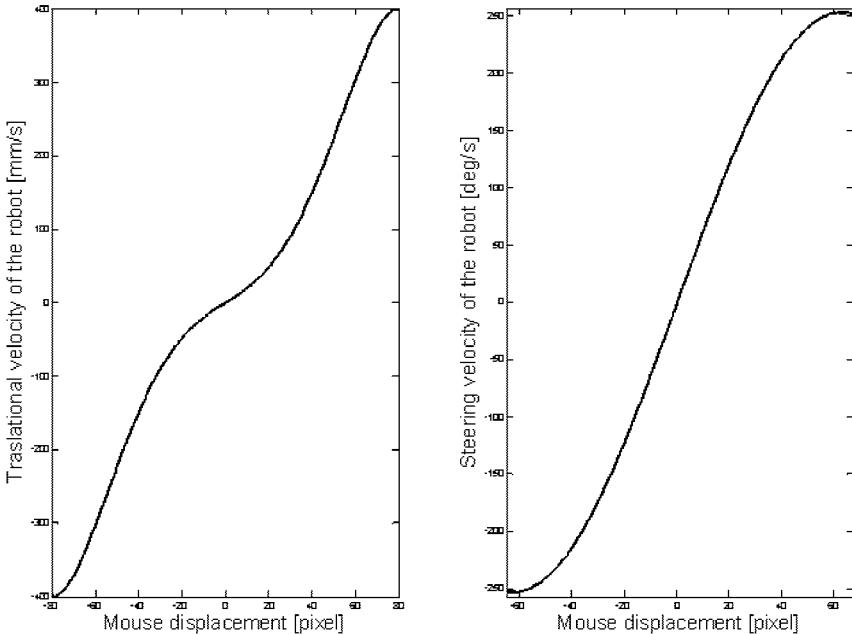


Fig. 6. Relation between joystick displacement and corresponding translational velocity (*left panel*) and steering velocity(*right panel*)

tidisciplinary expertise. The proposed approach starts from models of human sensory-motor coordination for modeling and developing interfacing mechanisms that exploit them at the best to obtain natural perception and control. Preliminary experiments have been conducted for the case of the head anticipatory movements associated with steering, during locomotion.

The results obtained with three different experimental set-ups show that anticipatory head movements occur even when the person is driving a device, like those used in sessions 1 and 3 of the experiments, instead of being walking. Actually, for one of the experimental scenario this result was not obtained. A comparative analysis of the three cases induces to think that a critical role is played by the perception that the person can have of navigation/locomotion. This was in fact reduced in the second scenario, due to the smaller dimensions of the image, the narrow field of view and the reduced steering angles. This interpretation may be further confirmed by some preliminary results that we obtained when the feedback was given to the user by mean of a traditional monitor rather than using a wearable display. In these cases, we were not able to detect any significant movement of the head in none of the above reported scenarios. This circumstance was in fact perceived by the user not as a fully-immersive navigation but rather as a precision pointing task of an external artifact. Experimental results also show that head movements always

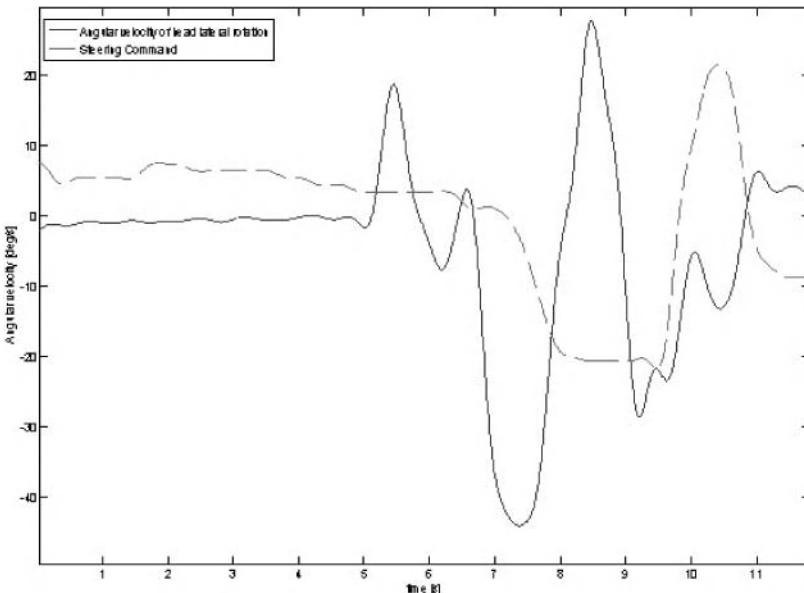


Fig. 7. Relation between joystick displacement and corresponding translational velocity (*left panel*) and steering velocity(*right panel*)

anticipate commands to the input device: though steering commands may be issued even without an anticipatory head movement, when a head movement is detected a related action on the input device can always be detected as well. Also, the head movements occurred visibly in advance with respect to the steering command, e.g. up to 0.5 sec, which is a significant anticipation in case it is used for controlling a robotic device.

In conclusion, the results obtained with this first experimental work indicate that the proposed hypothesis is valid and that a further research effort is worthwhile in the direction of using this novel principle to develop natural interfaces, which in fact can be very useful in many tasks, with different devices.

Further developments will concern the use of the signals from the vestibular interface for controlling a robotic device, thus realizing a real interfacing mechanism. An evaluation of the improvement of the control of the device, as well as of the perceived friendliness and easiness of use of the interface will be then possible and needed.

Acknowledgments

Part of the work described in this paper has been supported by the EU within the NEUROBOTICS Project (The fusion of NEUROscience and roBOTICS,

IST-FET Project 2003-001917), and the MiNOSC Project (MicroNeuroendoscopy of Spinal Cord, IST-RD Project QLG5-CT-2001-02150).

The first development of the artificial vestibular system was conducted with the support of the Humanoid Robotics Institute (HRI) of the Waseda University, at RoboCasa, a joint laboratory supported by the Italian Ministry of Foreign Affairs, General Directorate for Cultural Promotion and Cooperation.

References

1. Y. Hirata (2003) Proc. of the Int. Symposium on Microelectronics, Boston, MA, USA, November 16-20, 2003, 560–565
2. A. Knoll, C. Bekey, T.C. Henderson (2001) Robotics and autonomous systems 37(2-3):73-80
3. M.J. Johnson, E. Guglielmelli, C. Suppo, A. Pisetta, F. Leoni, G.A. Di Lauro, M.C. Carrozza, C. Laschi, P. Dario (2003) A fetch-and-carry robot appliance for elderly and disabled persons, In: Assistive Technology - Shaping the Future, G.M. Craddock, L.P. McCormack, R.B. Reilly, H.T.P. Knops (Ed.s), Assistive Technology Research Series, IOS Press
4. P. Dario, Sandini G., Aebischer P. (Eds) (1993), Robots and Biological Systems: Towards a New Bionics?, NATO ASI Series, Springer-Verlag
5. M. Gasson, B. Hutt, I. Goodhew, P. Kyberd, K. Warwick (2002), Proc. of the IEEE International Workshop on Robot and Human Interactive Communication, Berlin
6. D. Perzanowski, A.C. Schultz, W. Adams, E. Marsh, M. Bugajska (2001) IEEE Intelligent Systems 16(1):16-21
7. J.M. Carmena, M.A. Lebedev, R.E. Crist, J.E. O'Doherty, D.M. Santucci, D.F. Dimitrov, P.G. Patil, C.S. Henriquez, M.A.L. Nicolelis (2003) PLoS Biology 1(2):193-208
8. X. Navarro, T.B. Krueger, N. Lago, S. Micera, P. Dario, T. Stieglitz (2005) Journal of the Peripheral Nervous System 10(3):229–258
9. M. Land, N. Mennie, J. Rusted (1999) Perception 28(11):1311–1328
10. R.S. Johansson, G.B. Westling, A. Backstrom, J.R. Flanagan (2001) Journal of Neuroscience, 21:6917-6932
11. A. Berthoz (2000) The Brain's Sense of Movement: Perspectives in Cognitive Neuroscience, Harvard University Press
12. X. Lu, J. Ashe (2005) Neuron 45(6):967–973
13. J.C. Van der Heide, B. Otten, L.A. van Eykern, M. Hadders-Algra (2003) Experimental Brain Research 151(1):32–45
14. E. Kowler, A.J. Martins, M. Pavel (1984) Vision Research 24:197–210
15. S.G. Wells, G.R. Barnes (1998) Experimental Brain Research 120:129–33
16. R. Grasso, P. Prvost, Y.P. Ivanenko, A. Berthoz (1998) Neuroscience Letters 253:115–118
17. F. Patanè, C. Laschi, H. Miwa, E. Guglielmelli, P. Dario, A. Takanishi (2004) Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan
18. F. Ferraris, U. Grimaldi, M. Parvis (1995) Sensors and Materials 7(5):311–330

19. L. Ascari, C. Stefanini, A. Menciassi, S. Sahoo, P. Rabischong, P. Dario (2003) Proc. of the IEEE International Conference on Robotics and Automation, Taipei
20. L. Ascari, U. Bertocchi, C. Laschi, C. Stefanini, A. Starita, P. Dario (2004) Proc. of the IEEE International Conference on Robotics and Automation, New Orleans, USA
21. C. André-Deshays, A. Berthoz, M. Revel (1988) Experimental Brain Research 69:399–406
22. R. Grasso, C. Assaiante, P. Prevost, A. Berthoz (1998) Neuroscience and biobehavioral reviews 22:533–539
23. R. Grasso, S. Glasauer, Y. Takei, A. Berthoz (1996) Neuroreport 7(6):1170–1174
24. R. Grasso R, Y.P. Ivanenko, J. McIntyre, I. Viaud-Delmon, A. Berthoz (2000) Neuroreport 11(4):775–778
25. R. Grasso, P. Prevost, Y.P. Ivanenko, A. Berthoz (1998) Neuroscience Letters 253:115–118
26. Y. Takei, R. Grasso, M.A. Amorim, A. Berthoz (1997) Experimental Brain Research 115(2):361–368
27. M. Zugaro, A. Berthoz, S. Wiener (2002) Hippocampus 12:481–486

How Social Robots Will Help Us to Diagnose, Treat, and Understand Autism

Brian Scassellati

Yale University, 51 Prospect Street, New Haven, CT, 06520, scaz@cs.yale.edu

Autism is a pervasive developmental disorder that is characterized by social and communicative impairments. Social robots recognize and respond to human social cues with appropriate behaviors. Social robots, and the technology used in their construction, can be unique tools in the study of autism. Based on three years of integration and immersion with a clinical research group, this paper discusses how social robots will make an impact on the ways in which we diagnose, treat, and understand autism.

1 Introduction

For the past three years, our robotics group has been immersed in one of the premiere clinical research groups studying autism, led by Ami Klin and Fred Volkmar at the Yale Child Study Center. This paper outlines our initial attempts to apply technology from social robotics to the unique clinical problems of autism.

Section 2 provides an introduction to autism which highlights some of the difficulties with current diagnostic standards and research techniques. Section 3 describes attempts to use robots as therapeutic aids and discusses the as yet unfulfilled promise of these methods. Section 4 describes how diagnosis can be improved through the use of both passive social cue measurement and interactions with a social robot to provide quantitative, objective measurements of social response. Section 5 speculates on how the use of social robots in autism research might lead to a greater understanding of the disorder.

2 What We Know About Autism

Autism was first identified in 1943 by Kanner who emphasized that this congenital condition was characterized by an inability to relate to other people from the first days of life. Over the past 6 decades considerable work has been

done to refine the concept and identify important aspects of the condition. Current research suggests that 1 in every 300 children will be diagnosed with the broadly-defined autism spectrum disorder (ASD), but studies have found prevalence rates that vary between 1 in every 500 to 1 in every 166. For comparison, 1 in every 800 children is born with Down syndrome, 1 in every 450 will have juvenile diabetes, and 1 in every 333 will develop cancer by the age of 20. Furthermore, the rate of diagnosis increased six-fold between 1994 and 2003. It is unclear how much of this increase is a result of changes in the diagnostic criteria, increases in awareness, or a true increase in prevalence. Early intervention is critical to enabling a positive long-term outcome, but even with early intervention, many individuals will need high levels of support and care throughout their lives [fDCP06].

The social disability in autism is a profound one affecting a person's capacity for understanding other people and their feelings, and for establishing reciprocal relationships. To date, autism remains a behaviorally specified disorder [VLB⁺04]; there is no blood test, no genetic screening, and no functional imaging test that can diagnose autism. Diagnosis relies on the clinician's intuitive feel for the child's social skills including eye-to-eye gaze, facial expression, body postures, and gestures. These observational judgments are then quantified according to standardized protocols that are both imprecise and subjective (e.g. [SBC84, Mul95]). The broad disagreement of clinicians on individual diagnoses creates difficulties both for selecting appropriate treatment for individuals and for reporting the results of population-based studies [KLCV00, VCK05].

The need for improved characterization of the core social disorder in autism that underlies the broad spectrum of syndrome manifestations has been highlighted by genetic and neuro-functional research [VLB⁺04, SR05]. It is clear that autism is a brain-based disorder with a strong genetic basis. Approximately 25% of children with autism develop seizures and the recurrence risk for siblings is between 2 and 10% (a 50–100 fold increase over the general population). Genetic studies have underscored the importance of understanding both the broader phenotype of autism and the remarkable heterogeneity in syndrome expression. However, the causes and etiology of the disorder are still unknown [VLB⁺04]. A more precise characterization and quantification of social dysfunction is required to direct neurobiological research in autism is still lacking [BPR96, KJS⁺02a].

3 Robots Provide Motivation and Engagement in Therapy

A few projects world-wide seek to include robots as part of the therapeutic regimen for individuals with autism [WD99, MTT02, Dau00, KNY]. Each of these studies has demonstrated that robots generate a high degree of motivation and engagement in subjects, including subjects who are unlikely or unwilling to interact socially with human therapists. The great hope of this

line of research is the development of a "social crutch," a robot that motivates and engages children, teaches them social skills incrementally, and assists in the transfer of this knowledge to interactions with humans. Since the behavior of a robot can be decomposed arbitrarily, turning off some behaviors while leaving others intact, we can selectively construct complex social abilities through layers of social responses, sometimes in combinations that cannot be performed by humans. This layering of response allows the therapist to focus on single behaviors while ignoring all other social factors or maintaining their response at a constant. This type of isolation of cues and responses is difficult to train human therapists to perform. The as yet unfulfilled promise of this line of research is that learning skills with a robot will be simpler because of the ability to isolate particular responses, thus allowing a unique form of incremental therapy. In a different domain, but using a similar principle, we have preliminary data suggesting that computerized face perception training leads to therapeutic benefits for individuals with autism [SK05].

However, the design criteria for what makes individuals with autism likely to respond to these devices are not understood. The robots used in these studies include four-wheeled rovers, anthropomorphic robotic dolls, a spherical robot ball with eyes, and an expressive snowman-like device. These robots show a wide range of anthropomorphic characteristics, behavioral repertoires, aesthetics, and sensory and interactive capabilities. While there are many studies of the effects of these interaction variables on typical adults, very little is known about how individuals with autism respond to these design dimensions. While we have many expectations for why children with autism respond so positively to these robots, we have no direct experimental data that provide an analysis of the design criteria that are important to producing this response. We would expect that one reason that children (both autistic and typically developing) would respond so positively to robots (as seen in the studies mentioned above) is that the robots offer simple, contingent, predictable responses to the child's actions. The preference for things that interact with us, that respond directly to our actions, is well known for typical adults and children. However, our expectations derived from studies with typical adults and children often do not carry over to adults and children with autism.

As an example of how our expectations regarding these design criteria are misplaced, we conducted a simple pilot study that looked at the effects of social contingency. Using an extremely simple commercial robot called ESRA (see Figure 1) which generates a small set of facial expressions using five servos, we compared children's attentiveness to the robot in two experimental conditions. In the non-contingent condition, ESRA was programmed to perform a short script which included both a set of actions and an accompanying audio file that was played from speakers hidden near the robot. The robot had no sensory capabilities and did not respond to anything that the child did. In the contingent condition, the robot performed behaviors from this same repertoire, but the initiation of these behaviors was triggered by an experi-

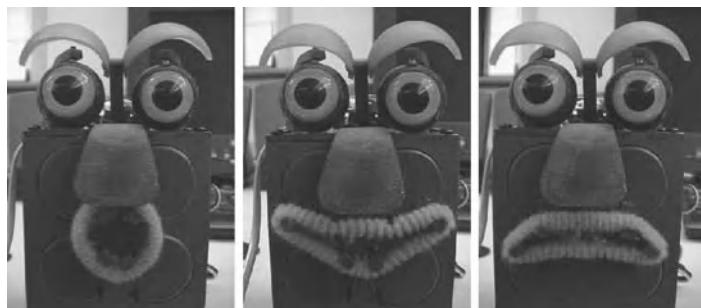


Fig. 1. Three facial expressions from the ESRA robot.

menter sitting behind a one-way mirror. The experimenter triggered behaviors that they deemed to be socially appropriate based on the actions of the child. 13 subjects (mean age 3.4 years) including 7 children with autism spectrum disorders and 6 typically developing children were positioned across a table from ESRA for a period of 3-5 minutes. Even with the extremely limited capabilities of ESRA, the robot was well tolerated by all of the children and many of them (including many of those within the autism spectrum) seemed to thoroughly enjoy the session.

In the contingent condition, typical children were universally engaged with the robot, and often spent the entire session touching the robot, vocalizing at the robot, and smiling at the robot. In the non-contingent condition, typically developing children were initially attracted to the robot but tended to lose interest quickly, preferring instead to attend to other (non-robotic) toys that were in the room. In contrast, the children with autism did not differ significantly in their interactions between these two experimental conditions. They tended to spend almost all of the session attending to the robot, regardless of whether or not it was responding contingently to them. In both conditions, children with autism often generated behavior similar to their typically developing peers, including smiling at the robot, making eye contact, and vocalizing to the robot. For many of the children with autism in this pilot study, these positive proto-social behaviors are rarely seen in a naturalistic context.

These results are only preliminary, but they point out that the severe social deficits that accompany this disorder do not respond in accordance with the interaction dynamics that have been observed with typical adults by research in human-robot interaction. These results should also not be interpreted to show that children with autism fail to respond to any form of social contingency. Because of the very wide range of functional capacities of children and adults who receive the diagnosis of autism (see the following section), these types of generalizations are notoriously dangerous. This simple study does demonstrate that further detailed study of these design variables are necessary to begin to delineate the factors that cause this remarkable response from children with autism.

4 Quantitative Objective Metrics for Diagnosis

Many of the diagnostic problems associated with autism would be alleviated by the introduction of quantitative, objective measurements of social response. We believe that this can be accomplished through two methods: through passive observation of the child at play or in interactions with caregivers and clinicians, and through structured interactions with robots that are able to create standardized social "presses" designed to elicit particular social responses. While the information gathered from both passive and interactive systems will not replace the expert judgment of a trained clinician, providing high-reliability quantitative measurements will provide a unique window into the way in which children with autism attempt to process naturalistic social situations. These metrics provide both an opportunity to compare populations of individuals in a standardized manner and the possibility of tracking the progress of a single individual across time. Because some of the social cues that we measure (gaze direction in particular) are recorded in greater detail and at an earlier age than can occur in typical clinical evaluations, one possible outcome of this work is a performance-based screening technique capable of detecting vulnerability for autism in infants and toddlers.

4.1 Passive Sensing

Passive sensors record information on social response without directly engaging in interactions. In many cases, the perceptual systems of a social robot can act as a passive social cue sensor. To evaluate the usefulness of this idea, we have outfitted some of our clinical evaluation rooms with cameras and microphones and software similar to that used on the social robots Nico, Cog, and Kismet [Sca03, Sca01, BEF⁺⁰⁰]. Most of these passive sensors record and interpret data while the subjects are actively engaged in standard clinical evaluations and do not require any specific protocol to be employed. Currently, three cue recognition systems have been developed: (1) detecting gaze direction, (2) tracking the position of individuals as they move throughout a room, and (3) measuring aspects of prosody from human voices.

Gaze direction and focus of attention

For several years, we have used commercial eye-tracking systems which require subjects to wear a baseball cap with an inertial tracking system and camera/eyepiece assembly which allows us to record close-up images of one eye. In addition to this commercial system, we have developed computational systems that give much less accurate recordings but do not require the subject to be instrumented. When viewing naturalistic social scenes, adolescents and adults with autism display gaze patterns which differ significantly between control populations (see Figure 2) [KJS^{+02a}, KJSV03, KJS^{+02b}]. Fixation time variables predicted level of social competence (e.g., at an average

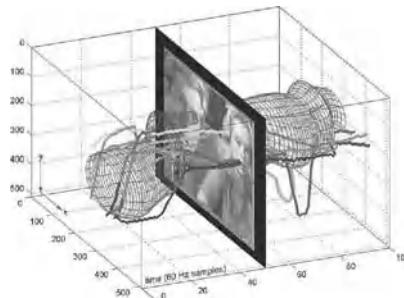


Fig. 2. Gaze patterns differ significantly between typical adolescents and adolescents with autism. This spatio-temporal plot shows 10 scan paths of individuals with autism (red lines) and the bounding volume (in blue) for 10 typical individuals. The typical group shows very similar, structured gaze patterns. The group with autism shows less structure, but is far from random. (Figure adapted from [KJS⁺02a]).

$r=.63$). This was the first experimental measure to successfully predict level of social competence in real life for individuals with autism. Visual fixation data related to viewing of naturalistic scenes of caregivers' approaches reveals markedly different patterns. Toddlers with autism fixate more on the mouth region rather than on eye regions of faces. Combined with experiments probing these children's capacity for mentally representing human action, it has been suggested that these children are treating human faces as physical contingencies rather than social objects (they fixate on mouths because of the physical contingency between sounds and lip movements). Although visual fixation on regions of interest are sensitive measures of social dysfunction, moment-by-moment scan-paths are even more sensitive and offer further insight into the underlying dysfunction (see section 5 for an example) [KJS⁺02a].

Position tracking

Some of the most basic information on social response can be derived from the relative positioning of individuals. How close a child stands in relation to an adult, how often the child approaches an adult, how much time is spent near an adult, and whether or not the child responds when an adult approaches are a few of the relatively simple statistics that can be derived from positional information. These social cues, especially the concept of "personal space," are often deficient in individuals with autism and are part of the diagnostic criteria [VLB⁺04].

Using a pair of calibrated stereo cameras and a computational vision system developed in part by our team, we have been able to successfully track the position of individuals as they move about in our clinical space. Computed disparity information is used in conjunction with information on color, direction of motion, and background pixels to segment the moving objects in the scene. A multi-target tracking system (similar to the system developed in [Sca02])



Fig. 3. Two tracking images from the left camera of a calibrated stereo cameras rig. Individuals are tracked as they move throughout one of our clinical evaluation rooms during an interview session.

is then used to predict relative motion and identify motion trajectories of individuals. Figure 3 shows two images obtained during a standard diagnostic interview. Note that the recording and computation performed by this system impact the diagnostic interview no more than other video recording devices would.

Our initial experiments with this technique were able to successfully track the positions of toddlers during a standard behavioral assessment. This included instances when individuals left the field of view, were occluded completely by objects or other individuals, and changed postures dramatically (moving from a standing position to crouched in a corner to lying down horizontally). However, the range of motion of these children during the assessment is limited; in order to allow the completion of the evaluation, both the parent and the experimenter act to try to keep the child on-task at the table. We are currently deploying this system in a larger space that is used for social skills training sessions for adolescents with autism. We anticipate that the data obtained in this environment will be more indicative of natural social response.

Vocal prosody

Individuals with autism often have difficulty both generating and recognizing vocal prosody and intonation [SPM⁺01]. (Simply put, prosody refers to not what is said, but how it is said.) There are no standardized measures of prosody in the clinical literature [Pau05], and the only research instrument available [SKR90] is very laborious and thus seldom used in diagnostic evaluation or experimental studies.

We recently constructed a multi-stage Bayesian classifier capable of distinguishing between five categories of prosodic speech (prohibition, approval, soothing, attentional bids, and neutral utterances) with an accuracy of more than 75% on a difficult set of vocal samples taken from typical adults (both

male and female). In comparison, human judges were able to correctly classify utterances 90% of the time within this data set [RMS04]. To develop this technique to the point where it can be used as a diagnostic tool in the clinic will require us to develop two different forms of classifier based on our initial system design. First, we would like to have a very selective system with a low false-positive rate that can be used continuously on microphone arrays in our clinical evaluation rooms. This system would mark portions of the recorded audio/video streams when extreme prosodic utterances occurred. Second, a system that can be used under more controlled conditions (during experimental protocols) would be developed that was more sensitive to prosodic cues but would suffer from higher rates of both false positives and false negatives. Both of these systems can be obtained by altering a small set of parameters in our initial multi-stage classifier design, but these systems have yet to be evaluated in the clinic.

4.2 Interactive Social Cue Measurement

While there is a vast array of information that can be obtained by passive sensing technologies, the use of interactive robots provides unique opportunities for examining social responses in a level of detail that has not previously been available. These advantages include the following:

1. By generating a social press designed to elicit a particular social response from the subject, the interactive system can selectively probe for information on low-occurrence social behaviors or on behaviors that may not easily emerge in diagnostic sessions in the clinic.
2. The robot provides a repeatable, standardized stimulus and recording methodology. Because both the production and recognition are free from subjective bias, the process of comparing data on social responses between individuals or for a single individual across time will be greatly simplified. As a result, the interactive system may prove to be a useful evaluation tool in measuring the success of therapeutic programs and may provide a standard for reporting social abilities within the autism literature.
3. Because a robotic system can generate social cues and record measurements autonomously, simple interactive toys can be designed to collect data outside of the clinic, effectively increasing both the quantity and quality of data that a clinician can obtain without extensive field work.

We have developed one simple device, called Playtest (see Figure 4), for determining auditory preferences that can be used in the clinic or in the home. When a button is pressed, the device plays one of two audio clips, produces a series of flashing lights to entice attention, and records the time, date, button pressed and audio clip played to non-volatile memory. This device can be sent home with a family to collect information on typical play patterns. This method has been shown to have important diagnostic value [KLi91] since it

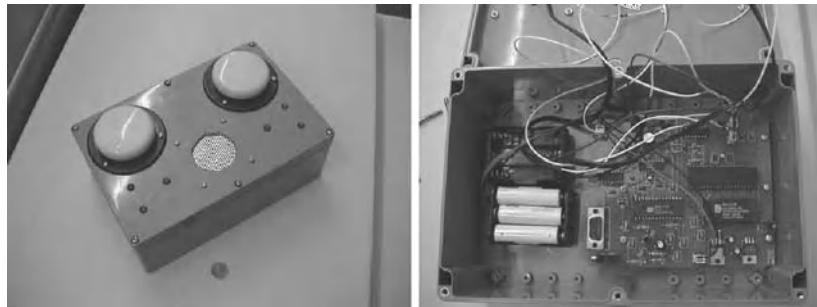


Fig. 4. External view (left) and internal view (right) of the Playtest device for measuring auditory preferences in the home. See section 4.2 for a description.

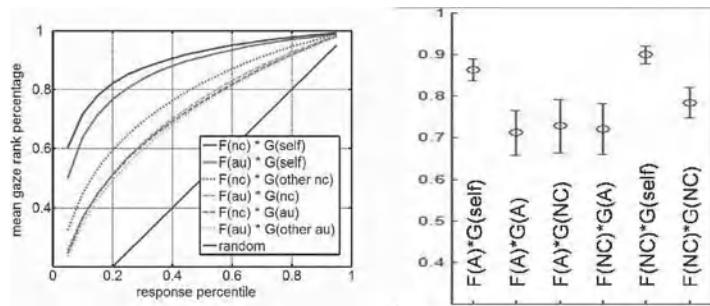


Fig. 5. Results of linear discriminant analysis of autistic (au) and normal (nc) gaze patterns. Linear filters $F(x)$ are trained to reproduce the gaze pattern $G(x)$ of each individual x . Filters can then be applied to predict the gaze patterns of any other individual. For example, $F(A)*G(self)$ indicates a filter trained on an individual with autism is tested on that same individual while $F(NC)*G(A)$ indicates a filter trained on a control individual is tested on data from an individual with autism. At left, the mean performance of this data (y-axis) is a function of the response percentile of individual pairings. At right, significant differences (all $p < 0.01$ for a two-tailed t-test) are seen between the following classes: (1) $F(NC)*G(self)$, (2) $F(A)*G(self)$, (3) $F(NC)*G(NC)$, and (4) the three other conditions. See section 5 for a discussion.

can measure listening preferences to speech sounds, abnormalities of which are among the most robust predictors of subsequent diagnosis of autism [Lor95].

5 Robots as Tools of Understanding

The fine-grained analysis of social capabilities that result from work on therapeutic and diagnostic applications have the potential to enhance our understanding of autistic disorders. We have already encountered one example

of this potential in our pilot studies of gaze detection. Based on our earlier observations on the differences in gaze direction between typically developing individuals and individuals with autism and in response to our need to characterize potential looking patterns for a robot, we have begun to generate predictive models that show not only the focus of an individual's gaze but also provides an explanation of why they choose to look at particular locations. A simple classifier (a linear discriminant) was trained to replicate the gaze patterns of a particular individual (see Figure 5). The performance of this predictor for a single frame is evaluated by having the filter rank-order each location in the image and selecting the rank of the location actually chosen by a particular individual. Thus, random performance across a sequence of images results in a median rank score of 50th percentile, while perfect performance would result in a median rank score of 1.0 (100th percentile). Trained filters predict the gaze location of the individual they were trained upon with good accuracy (median rank scores of 90th -92nd percentile). By applying a filter trained on one individual to predict the data of a second individual, we can evaluate the similarity of the underlying visual search methods used by each individual. In a pilot experiment with this technique, typically developing individuals were found to all use similar strategies (median rank score in the 86th percentile). Significantly, autistic individuals failed to show similar visual search strategies both among other individuals with autism (73rd percentile) and among the typically developing population (72nd percentile). Filters trained on our control population were similarly unsuccessful at predicting the gaze patterns of individuals with autism (71st percentile). These preliminary results suggest that while our control population all used some of the same visual search strategies, individuals with autism were both not consistently using the same strategies as the control population nor were they using the strategies that other individuals with autism used.

Acknowledgments

This work would not be possible without the patience and innovation of Ami Klin, Fred Volkmar, Kasia Chawarska, Warren Jones, and their entire team at the Child Study Center. Ganghua Sun, Fred Shic, Liz Darbie, Avi Robinson-Mosher, Jim Logan, and Reuben Grinberg contributed to some of the preliminary findings reported here. Support for this work was provided by a National Science Foundation CAREER award (#0238334). Some parts of the architecture used in this work was constructed under NSF grants #0205542 (ITR: A Framework for Rapid Development of Reliable Robotics Software) and #0209122 (ITR: Dance, a Programming Language for the Control of Humanoid Robots) and from the DARPA CALO project (BAA 02-21). Support for development of interactive devices for autism diagnosis was provided by the Doris Duke Charitable Foundation.

References

- [BEF⁺00] Cynthia Breazeal, Aaron Edsinger, Paul Fitzpatrick, Brian Scassellati, and Paulina Varchavskiaia. Social constraints on animate vision. *IEEE Intelligent Systems*, July/August 2000.
- [BPR96] A. Bailey, W. Phillips, and W. Rutter. Autism: Towards an integration of clinical, genetic, neuropsychological, and neurobiological perspectives. *Journal of Child Psychology and Psychiatry*, 37:89–126, 1996.
- [Dau00] K. Dautenhahn. Design issues on interactive environments for children with autism. In *Proceedings International Conference on Disability, Virtual Reality and Associated Technologies (ICDVRAT)*, pages 153–161, 2000.
- [fDCP06] Centers for Disease Control and Prevention.
<http://www.cdc.gov/ncbddd/dd/dautism.htm>, 2006.
- [KJS⁺02a] A. Klin, W. Jones, R. Schultz, F. Volkmar, and D. Cohen. Defining and quantifying the social phenotype in autism. *American Journal of Psychiatry*, 159(6):895–908, 2002.
- [KJS⁺02b] A. Klin, W. Jones, R. Schultz, F.R. Volkmar, and D.J. Cohen. Visual fixation patterns during viewing of naturalistic social situations as predictors of social competence in individuals with autism. *Archives of General Psychiatry*, 59(9):809–816, 2002.
- [KJSV03] A. Klin, W. Jones, R.T. Schultz, and F.R. Volkmar. The enactive mind - from actions to cognition: Lessons from autism. *Philosophical Transactions of the Royal Society, Biological Sciences*, 358:345–360, 2003.
- [KLCV00] A. Klin, J. Lang, D.V. Cicchetti, and F.R. Volkmar. Interrater reliability of clinical diagnosis and dsm-iv criteria for autistic disorder: Results of the dsm-iv autism field trial. *Journal of Autism and Developmental Disorders*, 30(2):163–167, 2000.
- [Kli91] A. Klin. Young autistic children's listening preferences in regard to speech: A possible characterization of the symptom of social withdrawal. *Journal of Autism and Developmental Disorders*, 21(1):29–42, 1991.
- [KNY] H. Kozima, C. Nakagawa, and H. Yano. Designing a robot for spatio-temporal contingency-detection game. In *International Workshop on Robotic and Virtual Agents in Autism Therapy YEAR = 2000*.
- [Lor95] C. Lord. Follow-up of two-year olds referred for possible autism. *Journal of Child Psychology and Psychiatry*, 46(8):1365–1382, 1995.
- [MT'T02] F. Michaud and C. Thberge-Turmel. Mobile robotic toys and autism. In Kerstin Dautenhahn, Alan Bond, Lola Canamero, and Bruce Edmonds, editors, *Socially Intelligent Agents - Creating Relationships with Computers and Robots*, pages 125–132. Kluwer Academic Publishers, 2002.
- [Mul95] E.M. Mullen. *Mullen Scales of Early Learning: AGS Edition*. American Guidance Service, Circle Pines, MN, 1995.
- [Pau05] R. Paul. Communicative competence in individuals with autism. In F.R. Volkmar, R. Paul, A. Klin, and D.J. Cohen, editors, *Handbook of Autism and Pervasive Developmental Disorders*. Wiley, New York, 3rd edition edition, 2005.
- [RMS04] A. Robinson-Mosher and B. Scassellati. Prosody recognition in male infant-directed speech. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

- [SBC84] S.S. Sparrow, D. Balla, and D. Cicchetti. *Vineland Adaptive Behavior Scales, Expanded Edition*. American Guidance Service, Circle Pines, MN, 1984.
- [Sca01] Brian Scassellati. *Foundations for a Theory of Mind for a Humanoid Robot*. PhD thesis, Massachusetts Institute of Technology, 2001.
- [Sca02] Brian Scassellati. Theory of mind for a humanoid robot. *Autonomous Robots*, 12(1):13–24, 2002.
- [Sca03] B. Scassellati. Investigating models of social development using a humanoid robot. In *Proceedings of the International Joint Conference on Neural Networks*, 2003. CD-ROM.
- [SK05] R.T. Schultz and C. Kleinman. Personal communication. 2005.
- [SKR90] L. Shriberg, J. Kwiatkowski, and C. Rasmussen. *Prosody-Voice Screening Profile*. Communication Skillbuilders, Tuscon, AZ, 1990.
- [SPM⁺01] L.D. Shriberg, R. Paul, J.L. McSweeny, A. Klin, D.J. Cohen, and F.R. Volkmar. Speech and prosody characteristics of adolescents and adults with high functioning autism and asperger syndrome. *Journal of Speech, Language, and Hearing Research*, 44:1097–1115, 2001.
- [SR05] R.T. Schultz and D. Robins. Functional neuroimaging studies of autism. In F.R. Volkmar, R. Paul, A. Klin, and D.J. Cohen, editors, *Handbook of Autism and Pervasive Developmental Disorders*. Wiley, New York, 3rd edition edition, 2005.
- [VCK05] F.R. Volkmar, K. Chawarska, and A. Klin. Autism in infancy and early childhood. *Annual Review of Psychology*, 56:315–36, 2005.
- [VLB⁺04] F.R. Volkmar, C. Lord, A. Bailey, R.T. Schultz, and A. Klin. Autism and pervasive developmental disorders. *Journal of Child Psychology and Psychiatry*, 45(1):1–36, 2004.
- [WD99] I.P. Werry and K. Dautenhahn. Applying mobile robot technology to the rehabilitation of autistic children. In *Proceedings of SIRS '99, Symposium on Intelligent Robotics Systems*, 1999.

Invited Overview Talk

Robotics Science (Panel Discussion)

Expo 2005 Robotics Project

Hirohisa Hirukawa¹ and Hirochika Inoue¹

National Institute of Advanced Industrial Science and Technology (AIST),
1-1-1 Umezno, Tsukuba 305-8568 Japan,
{hiro.hirukawa,h.inoue}@aist.go.jp

Abstract. This paper overviews a robotics project at the Expo 2005. The project consists of long term experimental evaluation of practical robots at the Expo site simulating the society in the future and short term demonstration of prototype robots. The long term evaluation can let robots advance from the demonstration level to the practical use one, and the short term demonstration from the single shot experiment level to the demonstration one.

1 Introduction

The 2005 World Exposition (Expo 2005) has taken place at Aichi, Japan from Mar. 2005 for half a year. Though “Nature’s wisdom” is the message of Expo 2005, Expo 2005 has shown many kinds of advanced technologies as Human’s wisdom. Robots are one of major display at many pavilions. For example, music playing robots at Toyota pavilion is one of most popular attractions at Expo 2005.

New Energy and Industrial Technology Development Organization (NEDO for short), a funding agency of the Japanese Government, runs a robotics project at Expo 2005. The project consists of long term experimental evaluation of practical robots and short term demonstration of prototype robots, where the practical robots are expected to be used in the society before 2010 and the prototype ones before 2020.

The applications of the practical robots are the cleaning and security of the Expo site, autonomous wheelchairs, clerks at the information desks, and children sitters. The practical robots are used every day for the whole period of Expo 2005, i.e. 185 days. The Expo site is supposed to be a simulated society in the near future. The long term evaluation could let the robots advance from the demonstration level to the practical use one.

Sixty-five kinds of prototype robots have been developed, whose applications diverge widely. The demonstrations of the prototype robots had been

shown for two weeks in a simulated town of 2020 at the Expo site, where people and robots live together. Most of the prototype robots were developed by universities and the demonstration of two weeks let the robots advance from the single shot experiment level to the demonstration level, where the single shot experiment level means the level of a robot which can work properly only once for taking a champion video segment.

This paper overviews Expo 2005 Robotics Project of NEDO. Section 2 presents the details of the practical robots, and section 3 those of the prototype robots. Section 4 concludes the paper.

2 Long Term Evaluation of the Practical Robots

2.1 Mobile Robots

The first category of the practical robots include mobile robots that consist of cleaning robots, security robots and an autonomous wheelchair. The key technology of the robots is autonomous navigation in the open air. The navigation is implemented by combinations of dead reckoning, laser measurement sensor, RTK-GPS and RF-ID. More details of the robots are described in the following.

Cleaning robots

The missions of the cleaning robots include the cleaning of the main pedestrian loop of the Expo site, called Global Loop, whose diameter is about 1 km. The Global Loop has many curves and slopes as well as many obstacles like benches and bending machines, and the robots can not see enough number of the GPS satellites at some places. Though the cleaning is done at night to avoid a crowd of visitors, a small number of employees may walk on the Global Loop. A map of the Global Loop is shown in Fig.1 and a photograph of the Global Loop in the daytime in Fig.2.

The cleaning robots consist of Subaru Robohiter RS1 developed by Fuji Heavy Industries Ltd. and SuiPPi by Matsushita Electric Works Ltd., whose picture are shown in Fig.3. The size of RS1 is $1,080mmW \times 1,600mmD \times 1,160mmH$ and the weight is about 360 kg. RS1 can clean about $3,600m^2$ per an hour when it travels at $3km/h$, and continue to work more than 3 hours. The size of SuiPPi is a bit larger than that of RS1, that is, $1,200mmW \times 1,513mmD \times 1,233mmH$ and the weight is about 500 kg. SuiPPi does not use GPS, and its localization is calibrated by a laser measurement sensor. The required infrastructure for the navigation of the robots are two RTK-GPS stations at the Expo site and reflectors for the laser along the Global Loop. The reflector is mounted every 20 meters on the loop, which is shown at the lower-left corner of Fig.4.



Fig. 1. Map of the Global Loop



Fig. 2. A photograph of the Global Loop

RS1 and SuiPPi had been successfully applied to the cleaning of the Global Loop for half a year. Though the robots are designed safe enough to be operated at the Expo site, it was not possible for the robots to meet all the related regulations for the safety. Therefore the robots had been operated under the supervision of a human. A photograph of RS1 working at night is shown in Fig.5. Fuji Heavy Industries Ltd. also developed Subaru Robohitter T1 that can exchange garbage cans.



Fig. 3. Cleaning robots RS1[Fuji Heavy Industries] and SuiPPi[Matsushita Electric Works] ©NEDO



Fig. 4. Reflector for the laser on the Global Loop

Security robots

Though security robots are expected to watch the Expo site, report suspicious events and remove them if possible, the current abilities of security robots are still limited. For example, it is very difficult to judge if a person is suspicious and if an object is unattended. Instead, ALSOK GuardRobo i, 190kg weight, developed by Sohgo Security Services Co.Ltd. can go around and send live pictures to a security center, find a human around the robot when nobody should be there, and report an object exists where nothing should be there. Mujiro/Ligurio, about 300 kg weight, developed by tmsuk Co.Ltd. can pick up an object by two manipulators via teleoperation. Photographs of the security robots are shown in Fig.6.

Autonomous wheelchair

An autonomous wheelchair is an enhanced electronic wheelchair with navigation ability. TAO Aicle, 40 kg weight, developed by Aisin Seiki Co.Ltd.,



Fig. 5. RS1 working on the Global Loop in the night

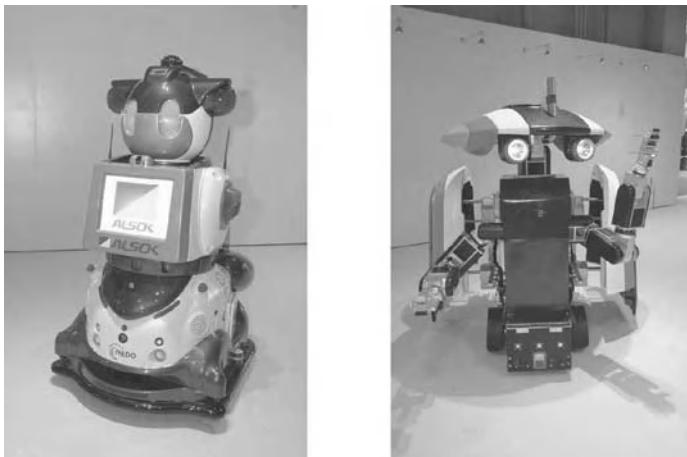


Fig. 6. ALSOK GuardRobo i[Sohgo Security Services] and Mujiro-Ligurio[tmsuk] ©NEDO

Fujitsu Ltd. and AIST can travel autonomously while avoiding obstacles[4]. The maximum speed of TAO Aicle is 2 km/hour and the maximum operating time is 3 hours. Each wheel is driven by a DC motor of 90 W, and the maximum weight of the passenger is 100 kg. The navigation of TAO Aicle depends on RTK-GPS and RF-ID embedded in the working environment. The wheelchair can know its position from the RF-IDs which are embedded in the floor. The RF-IDs are put with the interval of 15cm in a belt, and the belts are arranged every four meters. The visitors to the Expo could try to use TAO Aicle in the experimental environment. A photograph of TAO Aicle in the environment is shown in Fig.7. TAO Aicle can communicate with a traffic



Fig. 7. Autonomous wheelchair TAO Aicle[Aisin, Fujitsu, AIST] ©NEDO

signal via a wireless LAN, and stops when a traffic signal is red. Though the navigation of TAO Aicle should depend on the infrastructure significantly, the requirements for it can be considered realistic in the near future and the long term experiments at the Expo have proved that it can be operated reliably when it is available.

The user interface of TAO Aicle is implemented by a PDA and is designed to be friendly even to senior people. For example, the user can specify the destination by touching its panel.

2.2 Interactive Robots

The second category of the practical robots includes interactive robots that consist of clerk robots at the information desks in the Expo site, receptionist robots, and children sitter robots. The key technology of the robots is the understanding of spoken languages. The clerk robots can understand spoken Chinese, English, Japanese and Korean in a specific domain. The four languages were chosen since the Expo expects most visitors from the countries in which the languages are spoken.

Clerk robot at an information desk

The mission of the clerk robots is to answer questions about the Expo. The information desks are located near three gates of the Expo site, and the robots can understand spoken Chinese, English, Japanese and Korean in a noisy



Fig. 8. Clerk robot Actroid [Kokoro and Advanced Media] c

environment. Usually the information desks are surrounded by several tens of people and a cloud of people may walk around them.

Actroid developed by Kokoro Co.Ltd. and Advanced Media, Inc. has served as the clerk robot. Actroid is a concatenation of an actor and an andoroid, and is supposed to stand for a robot actor whose picture is shown in Fig.8. As you can see from the picture, Actroid has a realistic appearance of a human female. The behavior of Actroid are also designed to emulate those of a human. The appearance and behavior can make the visitors feel to interact with a real human. In fact, most visitors seem to be very happy when they asked questions to Actroid. The hardware and the behavior of Actroid were developed by Kokoro.

It is very difficult to understand the questions when Actroid does not know which language is spoken. So we ask the user of Actroid to start from hello or the equivalent in four languages. Then Actroid can be ready for the selected language. The noise and echo in the working environment of Actroid are reduced by a cancellation technique to realize the understanding. More than twenty responses are prepared to single question to realize a natural discourse. The speech recognition software was developed by Advanced Media. Actroid has been operated for the whole period of the Expo, and a huge number of people have enjoyed it.

Receptionist robot

The missions of the receptionist robot are to call a person specified by a visitor at a reception and to answer questions about a daily life like today's weather or hot topics. The receptionist robot can also understand spoken Chinese,

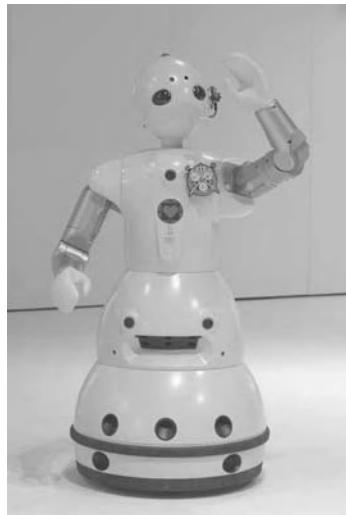


Fig. 9. Receptionist robot wakamaru [Mitsubishi Heavy Industries] ©NEDO

English, Japanese and Korean in a specific domain. The receptionist robot is wakamaru developed by Mitsubishi Heavy Industries Ltd., whose picture is shown in Fig.9. wakamaru has 1 meter height, 30 kg weight and 13 d.o.f. The robot can be operated for 2 hours by battery, and can go to a charge station autonomously.

wakamaru can also find how many human faces are around it and where they are. Then wakamaru can look at the speaking person and interact with the person more friendly.

wakamaru may hit a nearby user because it has two arms to express his feeling. So the tips of its arms are covered by soft material. A hand of a user may be caught by one arm and the body too. The related joints of wakamaru has mechanical stoppers to avoid the accident. Such safety design is very important to run wakamaru for 185 days in a cloud of people.

Childcare robot

The missions of the childcare robot is to interact with a child and make him/her enjoy. The childcare robot is PaPeRo developed by NEC Corporation[1, 2], whose picture is shown in Fig.10. PaPeRo has three kinds of ability to sit children. PaPeRo can recognize a human face when it is registered in advance. Ten faces can be registered at most. PaPeRo can understand Japanese spoken by children, which had been considered more difficult than that by Adults. PaPeRo is equipped with noise canceller software so that it can recognize voice even in very noisy environment like exhibition rooms. When PaPeRo fails to understand some words, PaPeRo analyzes why the understanding failed and suggests the user to speak in a better way. For example, PaPeRo may say



Fig. 10. Childcare robot PaPeRo [NEC] ©NEDO

“Please speak in a smaller voice”. The last ability of PaPeRo is to provide entertainment. PaPePo can dance when requested, and react when touched. The robot can provide quiz to a kid and teach him/her how to give greetings.

PaPeRo has been used to play with children, and the performance of the robot has been evaluated from various viewpoints. A photograph of PaPeRo playing with a child is shown in Fig.11.



Fig. 11. PaPeRo playing with a child

2.3 Biped Robot

The last category of the practical robots is a biped robot. The biped robot is not a humanoid but a biped dinosaur robot, developed by AIST and NEDO. Two types of the dinosaur robot were developed, one is Tyrannosaurus Rex and another Parasaurolophus, whose pictures are shown in Fig.12 and Fig.13 respectively. The length of the robots is about 3.5 meters and the weight is about 86 kg. The structure of the dinosaur robot is an endoskeleton type, which is covered by soft material. The soft cover can reduce the impact when the robot fall down, and realize a realistic appearance as well.

The biped walking is implemented by the software used for humanoid robot HRP-2. The dinosaur robot has shown its demonstration of fifteen minutes more than 1,500 times and fell down four times so far. The long term operation of the robot is also a very nice evaluation of biped robot technologies.



Fig. 12. Tyrannosaurus Rex [AIST and NEDO] ©AIST and NEDO



Fig. 13. Parasaurolophus [AIST and NEDO] ©AIST and NEDO

3 Short Term Demonstration of the Prototype Robots

Sixty-five kinds of prototype robots were developed and embedded in a simulated town in 2020. The message of the demonstration is “We live together with robots in 2020”. A perspective picture of the simulated town is shown in Fig.14 and a snapshot of the real counterpart is in Fig.15.



Fig. 14. Simulated town in 2020



Fig. 15. Snapshot of the simulated town in 2020

The prototype robots can be categorized into service robots, medical and welfare robots, outdoor robots, robots for special environments, partner robots, and humanoid robots. The list of the selected prototype robots is shown in Table 1.

Table 1. Selected List of the Prototype Robots for Expo 2005**Service robots**

Life Pod (Fuji Electric Sys.)	a smart vending machine for security services
COOPER (Yoshikawa Kikai et al.)	a robot caricaturist
Picture Robot (Gifu Ceramics RI et al.)	a ceramics painting robot
TELESarPHONE (U of Tokyo et al.)	a robot making people feel in a remote place
EMIEW (Hitachi)	a robot that acts as a work mate
Momochi (Kyushu U. et al.)	a contents-driven companion robot
SmartPal (Yaskawa Electric)	an autonomous mobile robot with dual arms
Cyber Assist Meister Robot (Saitama U.)	an interactive robot
Power Effector (Ritsumeikan U.)	a power effector directly operated by a human
ApriAlpha (Toshiba)	a robot that recognizes spoken language

Medical and Welfare Robots

MM-1 (NHK Eng. Services et al.)	a microsurgery robotic system
Surgery Robot (Nagoya U. et al.)	a surgical robot for remote micro surgery
EVE (Nagoya U.)	a high-precision patient robot
HAL (U. of Tsukuba)	a robotic power suit
Optical-tongue Robot (NEC Sys. Tech. et al.)	a robot that analyses taste of foods

Outdoor Robots

WallWalker (Miraikikai et al.)	a wall cleaning robot
MOIRA2 (Kobe U.)	a mobile inspection robot for rescue activity
IMR-Type 1 (IHI)	a leg-wheeled mobile robot
Dr.Impact (Gifu U. et al.)	a pipe inspection robot
Batting Robot (Hiroshima U.)	a high-speed batting robot
WOODY-1 (Waseda U.)	a robot woodcutter

Robots for Special Environments

ACM-R5 (TITech et al.)	an amphibious snake-like robot
Kinshachi Robot (Ryomei Eng.)	a real fish robot
UMRS-NBCT (Int. Rescue Sys.)	a mobile robot for NBC pollution

Partner Robots

DAGANE (Business Design Lab. et al.)	a verbal and nonverbal communication robot
Repliee Q1expo (Osaka U. et al.)	an android that looks like a human
InterAnimal (Okayama U. et al.)	an interactive animal-like robot
Robovie & wakamaru (Yoshimoto et al.)	comedian robot-duo
Dress-up Robot (Future U. Hakodate et al.)	an authoring robot
PBDR (Tohoku U. et al.)	a parner ballroom dance robot

Humanoid Robots

HRP-2 with Human Supervision (AIST)	a humanoid robot that can investigate objects
HRP-2 that interacts people (NAIST)	a humanoid robot that uses spoken dialog
WIND Robot System (Chiba IT et al.)	a small humanoid robot driven by a SIP
UT-μ2: magnum (U of Tokyo)	an animatronic humanoid robot
WABIAN-2 (Waseda U.)	a biped humanoid robot

One of the service robots is SmartPal developed by Yaskawa Electric Corporation, which can serve drinks at a cafe, and one of outdoor robots is WOODY-1 developed by WABOT-HOUSE which can cut branches of a wood while climbing it. Pictures of SmartPal and WOODY-1 are shown Fig.16.

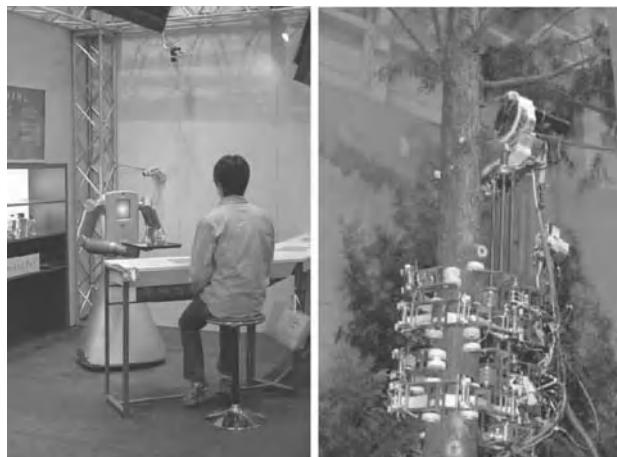


Fig. 16. SmartPal [Yaskawa Electric] and WOODY-1 [WABOT-HOUSE] ©NEDO

Figure 17 shows dance parter robot PBDR [Tohoku University], HRP-2 [AIST], Nagara [Gifu], Robovie and Wakamaru [ATR].



Fig. 17. PBDR, HRP-2, Nagara, Robovie and wakamaru ©NEDO

The demonstrations of the prototype robots had been shown for two weeks, and attracted more than 100,000 visitors during the exhibition. It was amazing that the robots developed by laboratories of universities can show the

demonstrations for two weeks without serious problems, and the exhibition offered a nice opportunity to train the robots.

4 Conclusions

The Expo site was a simulated society in 2010 to evaluate the practical robots for 185 days, and could let the robots advance from the demonstration level to the practical use one.

The exhibition site of the prototype robots was a simulated town in 2020 in which we live together with robots. The demonstration of two weeks let the robots advance from the single shot experiment level to the demonstration level.

From the results, Expo 2005 could train the robotics in Japan significantly, and we believe it can promote more applications of the robots to the society.

Acknowledgments

Expo 2005 Robotics Project was sponsored by New Energy and Industrial Technology Development Organization (NEDO). The authors also thanks all the contributors to the project for their excellent works, and Japan Association for the 2005 World Exposition for the planning and management of the project.

References

1. Yoshihiro Fujita: Prototype Personal Robot "PaPeRo" - A Partner-type Personal Robot -, Journal of Robotics & Mechatronics, Vol.14, No.1, pp.60-63, 2002.
2. Toshihiro Nishizawa, Shin'ichi Ohnaka, Yoshihiro Fujita: Safety Design for Childcare Robot, PaPeRo, IARP/IEEE-RA/EURON The 3rd announcement: The 4th Workshop on Technical Challenges for Dependable Robots in Human Environments, 2005.
3. H.Hirukawa, S.Kajita, F.Kanehiro, K.Kaneko and T.Isozumi, The Human-size Humanoid Robot that can Walk, Lie down and Get up, Robotics Research, pp.375-384, 2005.
4. O.Matsumoto, K.Komoriya, K.Tanie, K.Toda, S.Goto, T.Hatase, Y.Wakamatsu, H.Nishimura, H.Yamamoto, K.Sato and Y.Funakoshi, Development and Safety Issues of Intelligent Wheelchair TAO Aicle, The 4th Workshop on Technical Challenges for Dependable Robots in Human Environments, 2005.

Position Statement: Robotics Science

Ruzena Bajcsy¹

EECS Department, UC Berkeley, CA bajcsy@eecs.berkeley.edu

1 Introduction

Robotics as a subject of inquiry has had from its beginning an identity problem. Questions such as:

Is Robotics a science or engineering? Is it an application of certain discipline or does it have a core of problems, tools, methodologies which are unique to robotics?

Is robotics a multidiscipline or are there enough unique problems, methodologies, theories to stand as a single discipline on its own right?

Some researchers who are more pragmatic say: does it matter?

My answer is that indeed it matters, especially when one competes for national and international resources of support. It does matter, when one is concerned what training is needed to produce the professionals who could pursue and advance the field of robotics. It does matter, when one needs to clearly define what is robotics as a discipline, what are its objectives, what is the basic knowledge upon which robotics is build on and what are the criteria of success.

2 What Is Robotics?

Robotics has both the analytical and synthetic component, hence is both science and engineering, just like Computer Science is.

The difference between biology however is that we first must build artifacts (complex artifacts. Biology analyses the living nature) which then we analyze their behavior, their interaction with other artifacts and the environments including humans.

We use all the observations and data analysis as our counterparts in psychology, and sociology use. Think of robot behavior in any environment, or a swarm of robots interacting with each other.

The synthetic part of robotics brings us closer to the engineering discipline though robots are typically more complex than most of engineering artifacts. During the design process we have to employ all the engineering tools and methodologies in order to achieve the desired performance and behavior. The difficulty here for robotics is the complexity of interaction amongst the components within robots, the uncertainty and unpredictability of the environment in which they live, which leads to at best nondeterministic performance within some bounds.

So why I believe robotics is science?

Because it has to address, develop theoretical foundation of interactive complex physical and dynamic systems.

Just as chemists cannot claim that they understand a complex molecule until the can synthesize it, we cannot rest until we have foundation (tools, theories, methodologies) that will enable us to design (synthesize) complex robots with predictable behavior and guaranteed performance in a given environment within given bounds.

Let us remember that in the living world animals are adjusted (their body, perceptual and mobile capabilities) to their environment. Yet they are adaptive within some bounds. In turn we have to adhere to good and proven analytical methodologies to verify the predicted behavior of the robots.

3 Robotic System Science

It has been established for some times that Robots are made of physical components (sensors, motors, manipulators, hands legs, wheels, and of course computers).

Robotics science is also segmented into sub disciplines that utilizes the intellectual power from:

Perception, control, action and planning, Kinematics, dynamics, mobility, mechanisms of adaptation and learning, knowledge organization, behaviors and decision making: cooperative and competitive, and so on.

It is only natural that researchers feel more comfortable to study each of these sub-disciplines in isolation, especially if one accepts that each of these sub-disciplines is intellectually demanding. Nevertheless, I believe that one cannot make true progress in robotic science if one does not consider the system as the whole.

This is of course extremely demanding both intellectually but also materially, it requires a larger group of people with different skills, a proper infrastructure and it requires long term sustained funding.

The good news is that progress is being made both at the theoretical level as well as at the technological level.

At the technological level, we are benefiting from the miniaturization of computers, sensors, actuators, from new materials which are lighter, sturdier, more flexible and less energy hungry.

At the theoretical level, we have made great advances in control, non-linear control, hybrid control, distributed control, adaptive control, modeling complex systems, understanding multidimensional signals and geometry of space, data reduction without much loss of information, modeling uncertainty and making decision under uncertainty.

Finally, great progress has been in the development of learning mechanisms.

4 Conclusion and What Needs to Be Done

There are several implications following from the above analysis:

We need good models of the task that the robotic system is expected to perform.

We need models of the environment and context in which the task must be accomplished.

The robotic systems must be adaptive to unexpected changes though the variations must be bounded Under these conditions we must have theories and methodologies that guarantee performance.

If we take lessons from biology, we do not have universal living organism but rather organism that are adapted to their environments to accomplish task of survival. Different environments provide constraints on design and functionality of the organism.

Hence our aim should be understand these constraints and design robotic systems in a systematic way so that they can exist and perform the given task. I believe this is possible.

Author Index

- Abbott, Jake J. 49
Albu-Schäffer, Alim 5
Ascari, Luca 537
- Bajcsy, Ruzena 583
Bell, D.J. 163
Berthoz, Alain 537
Bertocchi, Ulisse 537
Bicchi, Antonio Bicchi 3
Biesiadecki, Jeffrey J. 254
Bingham, Brian 416
Blake, A. 295
Bolles, Bob 283
Bowling, Michael 190
Bradley, Albert M. 416
Brooks, Ari D. 34
Burgard, Wolfram 453
- Chestnutt, J. 103
Christensen, Henrik I. 187
Ciaravella, Gaetano 537
Criminisi, A. 295
Cross, G. 295
- Dario, Paolo 451, 537
Delage, Erick 305
Desai, Jaydev P. 34
Diolati, Nicola 22
Dissanayak, Gaminu 203
Donald, Bruce R. 337
Dong, L.X. 163
Durrant-Whyte, Hugh 401
- Eustice, Ryan 214, 430
- Ferguson, Dave 239
Fox, Dieter 487
- Ghodsi, Ali 190
Goldberg, Ken 510
Gordon, Geoffrey J. 69
- Hagita, Norihiro 525
Hing, James T. 34
Hirukawa, Hirohisa 101, 567
Hirzinger, Gerd 5
- Hollis, Ralph 327
Hsu, David 83
Huang, Shoudong 203
- Inoue, Hirochika 567
Ishiguro, Hiroshi 118, 523, 525
Itoh, Kazuko 357
- Jakuba, Michael 416
Jarvis, Ray 509
- Kagami, S. 103
Kaneko, Makoto 523
Kantor, George 327
Kautz, Henry 487
Kazerooni, H. 373
Kelly, Alonzo 237
Kogure, Kiyoshi 525
Kolgomorov, V. 295
Kuffner, J. 103
Kuniyoshi, Yasuo 473
Kurniawati, Hanna 83
- Laschi, Cecilia 537
Latombe, Jean-Claude 83
Lauwers, Tom 327
Lee, Honglak 305
Leger, Chris 254
Leonard, John 214
Levey, Christopher G. 337
Liao, Lin 487
- Maimone, Mark M. 254
Maini, Eliseo Stefano 537
Marayong, Panadda 49
Martínez Mozos, Óscar 453
Matthies, Larry 285
McGray, Craig D. 337
Merlet, Jean-Pierre 143, 175
Michel, P. 103
Milstein, Adam 190
Miyashita, Takahiro 525
- Nakamura, Yoshihiko 3, 128
Nebot, Eduardo M. 268
Nelson, B.J. 163

- Newman, Paul 187
Ng, Andrew Y. 305
Niemeyer, Günter 22
Nishiwaki, K. 103

Ogura, Yu 357
Okamura, Allison M. 49
Ott, Christian 5

Paprotny, Igor 337
Patane', Francesco 537
Pinneau, Joelle 69
Pizarro, Oscar 430
Plotnik, Aaron 402

Rock, Stephen 402
Roman, Christopher 430
Rother, C. 295
Rottmann, Axel 453
Roy, Nicholas 67
Rus, Daniela 337

Sangawa, Shinji 473
Sato, Tomomasa 509
Scassellati, Brian 552
Shirai, Yoshiaki 283
Sibley, Gabe 285
Siegwart, Roland 67
Singh, Hanumat 430

Song, Dezhen 510
Stachniss, Cyrill 453
Stefanini, Cesare 537
Steger, R. 373
Stentz, Anthony 239
Stilman, M. 103
Subramanian, A. 163
Sukhatme, Gaurav 285
Suzuki, Shinsuke 473

Tajika, Taichi 525
Takanishi, Atsuo 357
Takano, Wataru 128
Tanner, Neal 22
Thompson, S. 103
Thorpe, Chuck 237
Tomlin, Claire J. 325
Tsumaki, Yuichi 145

Uchiyama, Masaru 145

Walter, Matthew 214
Wang, Zhan 203
Whitcomb, Louis L. 401
Wilkinson, Dana 190

Yamane, Katsu 128
Yoerger, Dana R. 416
Yoon, Woo-Keun 145

Printing: Krips bv, Meppel
Binding: Stürtz, Würzburg