

IEEE

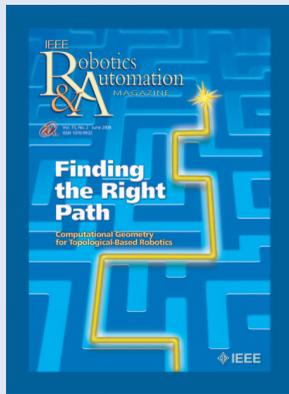
# Robotics & Automation MAGAZINE



Vol. 15, No. 2 June 2008  
ISSN 1070-9932

## Finding the Right Path

Computational Geometry  
for Topological-Based Robotics



This special issue presents state-of-the-art articles reporting on the unique aspects of topology-based approaches in robotics and the application of recent theoretical advances in computational geometry to the important robotics and path planning problems.

Background image:  
©Digital Vision

**IEEE Robotics & Automation Magazine** (ISSN 1070-9932) (IRAMEB) is published quarterly by the Institute of Electrical and Electronics Engineers, Inc. Headquarters: 3 Park Avenue, 17th Floor, New York, NY 10016-5997 USA, Telephone: +1 212 419 7900. Responsibility for the content rests upon the authors and not upon the IEEE, the Society or its members. IEEE Service Center (for orders, subscriptions, address changes): 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331 USA. Telephone: +1 732 981 0060. Individual copies: IEEE members \$20.00 (first copy only), nonmembers \$86.00 per copy. Subscription rates: Annual subscription rates included in IEEE Robotics and Automation Society member dues. Subscription rates available on request. Copyright and reprint permission: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of U.S. Copyright law for the private use of patrons (1) those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923 USA; (2) pre-1978 articles without a fee. For other copying, reprint, or republication permission, write Copyrights and Permissions Department, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854. Copyright © 2008 by the Institute of Electrical and Electronics Engineers Inc. All rights reserved. Periodicals postage paid at New York and additional mailing offices. Postmaster: Send address changes to IEEE Robotics and Automation Magazine, IEEE, 445 Hoes Lane, Piscataway, NJ 08854 USA. Canadian GST #125634188

PRINTED IN THE U.S.A.

# IEEE Robotics & Automation MAGAZINE

Vol. 15, No. 2 June 2008  
ISSN 1070-9932

## FEATURES

- 16 Capturing an Evader in a Building**  
Randomized and deterministic algorithms for mobile robots  
*By Ichiro Suzuki and Paweł Żyliński*
- 27 Ways to Tell Robots Where to Go**  
Directing autonomous robots using topological instructions  
*By David Rawlinson and Ray Jarvis*
- 37 Hybrid Control for Robot Navigation**  
A hierarchical Q-learning algorithm  
*By Chunlin Chen, Han-Xiong Li, and Daoyi Dong*
- 48 Sensor-Based Robot Motion Planning**  
A tabu search approach  
*By Ellips Masehian and Mohammad Reza Amin-Naseri*
- 58 Roadmap-Based Path Planning**  
Using the Voronoi diagram for a clearance-based shortest path  
*By Priyadarshi Bhattacharya and Marina L. Gavrilova*
- 67 Online Algorithms with Discrete Visibility**  
Exploring unknown polygonal environments  
*By Subir Kumar Ghosh, Joel Wakeman Burdick, Amitava Bhattacharya, and Sudeep Sarkar*
- 77 Probabilistic Self-Localization and Mapping**  
An asynchronous multirate approach  
*By Leopoldo Armesto, Gianluca Ippoliti, Sauro Longhi, and Josep Tornero*
- 89 Robotics Research in Australia**  
A national perspective on the needs, themes, and major groups  
*By Nick Barnes and Alexander Zelinsky*
- 96 Nanokhod Exploration Rover**  
A rugged rover suited for small, low-cost, planetary lander missions  
*By André Schiele, Jens Romstedt, Christopher Lee, Hartmut Henkel, Sabine Klinkner, Reinhold Bertrand, Rudolf Rieder, Ralf Gellert, Göstar Klingelhöfer, Bodo Bernhardt, and Harald Michaelis*
- 108 Embedded System Design for Robots**  
Design concept, system architecture, and implementation  
*By Li-Wei Wu and Juwu-Sheng Hu*
- 122 Surgical and Interventional Robotics**  
Core concepts, technology, and design  
*By Peter Kazanzides, Gabor Fichtinger, Gregory D. Hager, Allison M. Okamura, Louis L. Whitcomb, and Russell H. Taylor*

## DEPARTMENTS

- |    |                        |     |                        |
|----|------------------------|-----|------------------------|
| 3  | From the Editor's Desk | 12  | TC Spotlight           |
| 4  | President's Message    | 14  | Education              |
| 6  | From the Guest Editors | 131 | Industry/Research News |
| 8  | Position               | 132 | Regional               |
| 10 | Society News           | 134 | Calendar               |

Digital Object Identifier 10.1109/MRA.2008.922387

*A Publication of the*  
 IEEE ROBOTICS AND AUTOMATION SOCIETY  
 Volume 15, No. 2 June 2008 ISSN 1070-9932

**EDITORIAL BOARD***Editor-in-Chief*

Prof. Stefano Stramigioli  
 IMPACT Institute  
 University of Twente  
 P.O. Box 217  
 7500AE Enschede  
 The Netherlands  
 Phone: +31 53 4892794  
 Fax: +31 53 4892223  
 Email: S.Stramigioli@ieee.org  
<http://www.ce.utwente.nl/smi/>

*Editor-in-Chief Emeritus*

Kimon P. Valavanis, University of South Florida (USA)

*Associate Editors*

Eugenio Guglielmelli Campus Bio-Medico University, Roma (Italy)  
 Peter W.L. Xu Massey University, Auckland (New Zealand)  
 Cristian Secchi University of Modena and Reggio Emilia (Italy)  
 Magnus Egerstedt Georgia Institute of Technology (USA)  
 Zhi-Dang Wang Chiba Institute of Technology (Japan)  
 Jonathan Roberts CSIRO (Australia)  
 Pedro J. Sanz Jaume-I University (Spain)

*Liaisons*  
 Frank C. Park Seoul National University, Korea (T-RO)  
 Michael Yu Wang Chinese University of Hong Kong (T-ASE)  
 Kostas Kyriakopoulos National Technical University of Athens (Greece) (EURON)

**IEEE RAS Vice-President for Publications**

Peter Luh, University of Connecticut (USA)

**RAM homepage:** <http://www.ieee-ras.org/ram>

**Robotics and Automation Society Administrator**

Rosalyn Graham Snyder  
 (r.g.snyder@ieee.org)

**Advertising Sales**

Susan Schneiderman  
 Business Development Manager  
 Tel: +1 732 562 3946 Fax: +1 732 981 1855  
 ss.ieeemedia@ieee.org

**IEEE Periodicals Magazines Department**

Debby Nowicki, Managing Editor  
 (d.nowicki@ieee.org)  
 Janet Dudar, Art Director  
 Gail A. Schnitzer, Assistant Art Director  
 Felicia Spagnoli, Advertising Production Manager  
 Peter M. Tuohy, Production Director  
 Dawn M. Melley, Editorial Director  
 Fran Zappulla, Staff Director, Publishing Operations

**IEEE-RAS Membership****and Subscription Information:**

+1 800 678 IEEE (4333); Fax: +1 732 463 3657  
<http://www.ieee.org>

**Let Your Voice Be Heard**

**H**ere we are again with a great issue on computational geometry in path planning, which is guest edited by Marina Gavrilova, Jorge Cortes, and Raymond Jarvis.

I invite you once again to visit <http://wiki.ieee-ras.org> and use our Society Wiki to comment and express your opinions on the contributions of the position papers of Herman Bruyninckx on robotics software as well as the position paper in this issue contributed by Wolfram Burgard on probabilistic approaches to robotics navigation. We hope this will become a forum for interesting discussions, which we may partially publish in the magazine.



We have the first multimedia attachments! Via *IEEE Xplore*, you can download multimedia extensions related to papers that appeared in the magazine. We invite you, as an author, to contribute

videos with the final version of your accepted paper as well, so that the video record of your work will be archived along with the paper.

In this issue, we will start a new series of tutorials on medical robotics contributed by various of our esteemed colleagues from Johns Hopkins University.

Last, but not least, I would like to take this occasion to invite you all to express your vote in the upcoming Administrative Committee (AdCom) elections as that is your chance to let your voice be heard within the society. You will again be able to vote for your candidates electronically.

I hope you will enjoy the issue!



Stefano Stramigioli  
 S.Stramigioli@ieee.org

**Coming Attractions!**

**Adaptable Compliance or  
Variable Stiffness Actuators**

**OFFICERS**

- President: Bruno Siciliano, *Università di Napoli Federico II (Italy)*
- President-Elect: Kazuhiro Kosuge, *Tohoku University (Japan)*
- Past President: Richard A. Volz, *Texas A&M University (USA)*
- Founding President: George Saridis, *Rensselaer Polytechnic University (USA)*
- Vice President, Publications Activities: Peter Luh, *University of Connecticut (USA)*
- Vice President, Conference Activities: John Hollerbach, *University of Utah (USA)*
- Vice President, Financial Activities: Ian Walker, *Clemson University (USA)*
- Vice President, Industrial Activities: Alexander Zelinsky, *CSIRO (Australia)*
- Vice President, Member Activities: Alícia Casals, *Universidad Politècnica de Catalunya (Spain)*
- Vice President, Technical Activities: Kenneth Goldberg, *University of California-Berkeley (USA)*
- Treasurer: Xiaoping Yun, *Naval Postgraduate School (USA)*
- Secretary: Frank Park, *Seoul National University (Korea)*
- IEEE Division X Director: William X. Gruver, *Simon Fraser University (Canada)*
- Editor-in-Chief, *IEEE Trans. on Automation Science & Engineering*: Nukala Viswanadham, *Indian School of Business (India)*
- Editor-in-Chief, *IEEE Transactions on Robotics*: Alessandro De Luca, *Università di Roma "La Sapienza" (Italy)*
- Editor-in-Chief, *IEEE Robotics & Automation Magazine*: Stefano Stramigioli, *University of Twente (The Netherlands)*
- IEEE International Conference on Robotics and Automation
  - General Chair, ICRA 2008: Maja J. Mataric, *University of Southern California (USA)*
  - General Co-Chair, ICRA 2008: Paul Schenker, *JPL (USA)*
  - Program Chairs, ICRA 2008: Stefan Schaal and Gaurav Sukhatme, *University of Southern California (USA)*
  - General Chair, ICRA 2009: Kazuhiro Kosuge, *Tohoku University (Japan)*
  - Program Chair, ICRA 2009: Katsushi Ikeuchi, *University of Tokyo (Japan)*

**ADMINISTRATIVE COMMITTEE**

- Terms ending in 2008
  - Aude Billard, *EPFL (Switzerland)*
  - Pierre Dupont, *Boston University (USA)*
  - Hideki Hashimoto, *University of Tokyo (Japan)*
  - Seth Hutchinson, *University of Illinois (USA)*
  - Katasushi Ikeuchi, *University of Tokyo (Japan)*
  - Kevin Lynch, *Northwestern University (USA)*
- Terms ending in 2009
  - Hajime Asama, *University of Tokyo (Japan)*
  - Rüdiger Dillmann, *Universität Karlsruhe (Germany)*
  - Toshio Fukuda, *Nagoya University (Japan)*
  - Vijay Kumar, *University of Pennsylvania (USA)*
  - Jean-Paul Laumond, *LAAS-CNRS (France)*
  - Roland Siegwart, *ETHZ (Switzerland)*
- Terms ending in 2010
  - Peter Corke, *CSIRO (Australia)*
  - Alessandro De Luca, *Università di Roma "La Sapienza" (Italy)*
  - Lynne Parker, *University of Tennessee-Knoxville (USA)*
  - Stefano Stramigioli, *University of Twente (The Netherlands)*
  - Shigeki Sugano, *Waseda University (Japan)*
  - Satoshi Tadokoro, *Tohoku University (Japan)*

**The Commitment and the Reward**

Around the time this issue goes to press, our flagship conference, the IEEE International Conference on Robotics and Automation (ICRA), will be taking place in Pasadena. The technical program looks very promising, featuring three keynotes, 110 sessions, 23 workshops



and tutorials, and a number of special events including our spring AdCom meeting, which is to be held in the middle of the conference to favor members' attendance, the Women in Robotics lunch, the Graduates of Last Decade (GOLD) reception, and a student party, a new initiative at our conference.

I am writing this column while enjoying a pleasant week of family holiday skiing on the Dolomiti di Brenta in North-eastern Italy. I very much looked forward to the Easter break after a hectic start of the new year and my presidential term. I knew already

from my predecessor Dick Volz about the kind of devotion and the amount of time required of such a demanding societal commitment. Yet, the flow of e-mails and telecons during these first three months, concerning, especially, the appointments of volunteers on the various boards and standing committees, has been simply overwhelming. The reward for such an intensive commitment has been that all the various activities and programs of our Society are on track and a number of new initiatives are being pursued. Here are the highlights from our Boards, and I would like to thank our six vice presidents—an outstanding group to work with (see the masthead on the side).

**Conference Activities Board (CAB)**

The conference portfolio of the IEEE Robotics and Automation Society (RAS) is getting even more complicated due to the increased number of sponsored conferences and, consequently, the paperwork and finances involved with each conference. The RAS finances now depend more on conferences (surpluses plus *IEEE Xplore* returns) than on any other factor. To manage the conference portfolio, the following steps are being considered:

- ◆ Previously, the meetings chair and the CAB treasurer handled the bulk of conference management. Since more assistance is required, the position of associate vice president of CAB (Shigeki Sugano) has been recast to include the responsibility of managing the conference portfolio. Additional positions to assist the CAB treasurer (Yi Guo) may be created.
- ◆ We are considering hiring an administrative assistant for the Conference Board, to help with all the reporting and financial details and to unburden Society volunteers.
- ◆ The timely closing of conferences and the prompt posting of proceedings on *IEEE Xplore* has a big financial impact. We are considering requiring the assistance of PaperPlaza to expedite proceedings' posting.
- ◆ Of the many sponsored conferences, only ICRA is scrutinized closely in terms of its intellectual impact and operations. Given the RAS exposure to other sponsored conferences, the Steering Committee for Technical Programs (SCTP) chaired by Vijay Kumar (associate vice president of CAB) will begin to scrutinize them similarly.

### Financial Activities Board (FAB)

RAS has budgeted the 2008 funds for an initiatives competition, for which proposals were solicited earlier this year. Sixteen proposals were received, which have been reviewed and evaluated by the proper boards, and these are being examined by FAB concerning their financial implications. FAB is also engaged in the preparation of the RAS budget for 2009 to keep us on a good path financially and to make sure that all financial activities flow smoothly.

### Industrial Activities Board (IAB)

The annual event between our Society and the International Federation of Robotics (IFR), the Joint Forum on Innovation and Entrepreneurship in Robotics and Automation (IERA), takes place in Munich, Germany, on 11 June 2008. The collocation with Robotik 2008, Germany's largest robotics conference, and Automatica 2008, the International Fair for Automation, constitutes an excellent opportunity to tighten the relationship with people from the industrial community. As in the previous editions of the forum, the Invention and Entrepreneurship Award for outstanding achievements in commercializing innovative robot and automation technology will be presented. IAB is also looking into streamlining its standardization and road-mapping activities, and a bimonthly hookup is being organized.

### Member Activities Board (MAB)

The local and student chapter leaders meet at ICRA during a dedicated workshop, organized by the Student Activities

Committee, for discussing, coordinating, and exchanging ideas at the chapter level. A series of scholarships have been arranged to support the travel of student and local chapter officers for attending this workshop.

### Publications Activities Board (PAB)

The new *IEEE Transactions on Haptics* will be published semi-annually with the first issue to be premiered this September. It will address the science, technology, and applications associated with information acquisition and object manipulation through touch. Haptic interactions relevant to this journal include all aspects of manual exploration and manipulation by humans and machines as well as the interactions between the two, which are performed in real, virtual, teleoperated, or networked environments. This new journal is cosponsored by our Society (45%), the IEEE Computer Society (45%), and the IEEE Consumer Electronics Society (10%). It is also technically cosponsored by the IEEE Engineering in Medicine and Biology Society. The editor-in-chief is Prof. J. Edward Colgate of Northwestern University, and the Editorial Board has been functional since November 2007. Please visit <http://www.computer.org/th> for more information.

### Technical Activities Board (TAB)

TAB is working on several new projects, which are as follows:

- ◆ a triennial comprehensive review of each technical committee
- ◆ an online database for all Distinguished Lectures
- ◆ a new format for the GOLD event at ICRA and IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

TAB is also studying new ways to present research milestones in robotics and automation, e.g., an online graphical interface for RAS best papers indexed by year and topic and linked to *IEEE Xplore*.

The sum of these activities is a testimony to the hard work and the spirit of initiative of our team of volunteers, who participate in helping our Society to continue to evolve and improve. We would welcome inputs from all our members. Please send your comments to me, any of the officers, or our Activities Coordinator Rosalyn Snyder. Contact information for RAS officers and committee chairs can be obtained at <http://www.ieee-ras.org>.



Bruno Siciliano  
siciliano@ieee.org  
RAS President 2008-2009

## Computational Geometry in Navigation and Path Planning

Marina Gavrilova, Jorge Cortes, and Raymond Jarvis

This special issue on computational geometry in navigation and path planning presents a collection of state-of-the-art articles reporting on the unique aspects of topology-based approaches in robotics. This issue presents the applications of computational geometry in obstacle avoidance, path planning with minimum clearance, autonomous mobile robot navigation, exploration of unknown polygonal environments, and it explores the practical relevance to the geographical information systems, mobile networks, risk avoidance, security, games, and online system design. The six contributions carefully selected for this special issue are connected by a common theme: the application of recent theoretical advances in computational geometry to the important robotics and path planning problems.

The article by Ichiro Suzuki and Paweł Żyliński presents a number of new approaches to capturing an evader in a constrained space. The authors provide several new strategies for a group of mobile robots on a three-dimensional grid such as a randomized algorithm for detecting an evader by one robot having the same maximum speed as the evader, a randomized algorithm for capturing an evader using two or more robots who can move slightly faster than the evader, and a deterministic algorithm for capturing an evader under specific constraints. The article critically compares all the proposed algorithms on their efficiency and robustness and makes a unique contribution to the areas of pursuit game design, randomized algorithms, and mobile robot planning.

The article by David Rawlinson and Ray Jarvis explores a challenging problem of teaching a robot to navigate directly to goals in unfamiliar or unknown environments. Language allows people to exploit other people's knowledge in such situations, by transferring all necessary information from one person to the other, whereas what means can assist robots in doing the same is the question asked by this article. The proposed solution directs an autonomous robot using efficient and universal topological instructions that can be incrementally interpreted by a moving robot that does not initially have its own map of the environment. Many real-world experiments featuring autonomous exploration and mapping led to

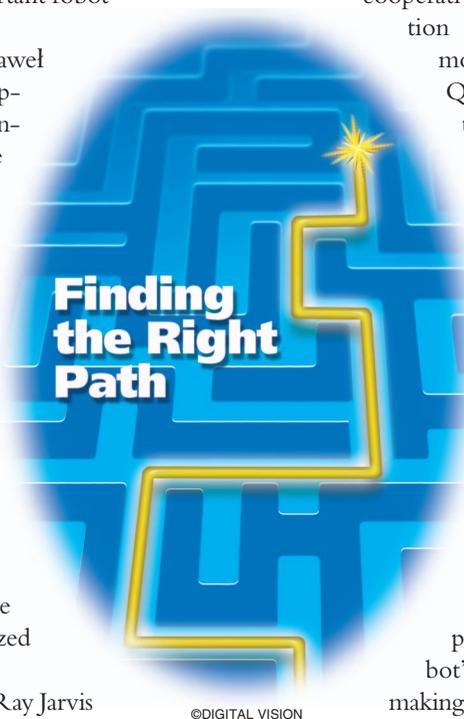
the remarkable conclusion that for this type of navigation, better object recognition capabilities are more important than better mapping capabilities.

Another approach to autonomous robot navigation is presented in the article by Chunlin Chen, Han-Xiong Li, and Daoyi Dong. In this work, the authors present a control method based on hierarchical Q-learning for mobile robot navigation in unknown environments. Hybrid control here refers to the integration of reactive control for local navigation and deliberative control for global navigation. The environment is represented by grid-topological maps that are constructed online during the process of learning to achieve the cooperative optimization of global and local navigation control. The navigation controls of the mobile robots are implemented by extending Q-learning to a hierarchical setting based on the model of a hybrid Markov decision process. The simulated and real experimentation shows that the proposed approach is capable of optimizing global navigation and avoiding the local minimum trap, so that the method works well in unknown dynamic environments and can be utilized as an effective integrated control scheme for navigation.

The article by Ellips Masehian and Mohammad Reza Amin-Naseri is devoted to the problem of sensor-based motion planning and proposes an approach based on the tabu search method to resolve it. The online motion planner presented in this article incorporates the robot's sensory data into the intelligent decision-making process guided by the tabu search technique.

The information from the environment is collected by performing a visibility scan, and the distance from the new robot location to the surrounding obstacles is computed by means of its radial rangefinder sensor readings. The proposed method is compared extensively with other offline and online techniques, such as the potential fields, distance transform, and the generalized Voronoi diagram methods.

The article by Priyadarshi Bhattacharya and Marina L. Gavrilova presents an efficient technique for computing a shortest path for a mobile agent moving among polygonal obstacles that will satisfy the specified clearance requirement. The algorithm utilizes the properties of the Voronoi diagram to obtain a path approximation and then proceeds to refine the path



through the corner-cutting technique. Extensive experimentation demonstrates that the method produces high-quality paths that are near optimal with respect to the length and have the required clearance from obstacles. The ease of implementation makes it an attractive technique for a number of applications, including real-time terrain visualization, risk avoidance, environmental modeling, and motion planning.

The article by Subir Kumar Ghosh, Joel Wakeman Burdick, Amitava Bhattacharya, and Sudeep Sarkar integrates some constraints from visual processing and robot navigation into the well-studied computational geometry problem of exploring unknown polygonal environments with obstacles. The authors propose online algorithms that can be used to explore an unknown polygonal environment by a point robot. The algorithms compute visibility polygons from a set of chosen points on the path of a robot and use the reduction of the number of visibility polygons as a criterion for minimizing the cost of robotic exploration, thus improving efficiency. In addition, the article demonstrates that the presented exploration algorithms for a point robot can be also treated as approximation

***This special issue presents  
a collection of articles on  
topology-based approaches  
in robotics.***

algorithms for the art gallery problem with an additional visibility constraint.

We would like to extend our sincere appreciation to all the authors who submitted their articles to the special issue and to all the referees for their meticulous and valuable reviews. It is our hope that this fine collection of articles presented in this issue will be a valuable resource for all the readers of *IEEE Robotics and Automation Magazine* and will stimulate further research into the growing area of the applications of computational geometry methods in robotics and path planning.



**Riverhawk Company**

# BEYOND BEARINGS.

## FLEXURAL PIVOTS®

Where Tradition Meets Tomorrow.

If you can fold it, move it, or flex it, we can design a motion solution. Riverhawk Free-Flex® Pivots are Frictionless Bearings backed by over 50 years of reliable performance in over 4 million uses. In the precision-driven world of robotics and automation, the Riverhawk Free-Flex® Pivot delivers. From scanning mirrors in fighter jets, to antenna arms and solar panels in space, Riverhawk Free-Flex® Pivots are meeting the challenges. When the situation calls for a reliable, unique solution, consider Riverhawk Free-Flex® Pivots...Engineered Solutions.

- FRICTIONLESS
- STICTION-FREE
- NO LUBRICATION REQUIRED
- MAINTENANCE-FREE
- SHOCK RESISTANT
- INFINITE CYCLE LIFE

**WWW.FLEXPIVOTS.COM**

215 Clinton Road • New Hartford, NY 13413  
Voice: 315.768.4855 • Fax: 315.768.4941 • E-mail: info@riverhawk.com

## Probabilistic Approaches to Robot Navigation

By Wolfram Burgard

Perception and state estimation are still some of the key problems in the area of robotics. Robots need to perceive their environment and potentially update a model of the environment or their own state to operate autonomously. In the last decade, it has been proved that probabilistic techniques are the most powerful tools for numerous state estimation and perception problems including mapping, localization, simultaneous localization and mapping (SLAM), tracking, and many others.

The key advantages of probabilistic approaches are their rigorous mathematical framework and the fact that the underlying equations can be derived from first principles. They even form the basis for theoretic approaches to action planning, which allows physical agents such as robots to act optimally even under uncertainty. Although the mathematical foundations of these approaches are well understood and many successful applications have been developed, there still are open issues for research, and I think that we should focus on these aspects to drive the understanding of core robotics problems and their potential solutions further. For example, real-world applications often require the careful design of the underlying models. Additionally, we need to understand what good approximations are and what impact they might have. One of the most important problems to date is the application of probabilistic approaches to systems with high-dimensional state spaces, whose sheer complexity prevents us from applying standard methods. In the remainder of this article, I will point out some of these aspects. I will also describe potential directions for future research.

### Model Design

Probabilistic approaches typically rely on the recursive Bayesian filtering scheme that requires the careful specification of the so-called perception and transition models. In many approaches found in the literature, these models have carefully been hand-crafted. Although the resulting systems often show an impressive robustness, we usually do not know what the underlying assumptions are, why specific choices in the design of these models have been done, and why they work so well. For example, consider the situation in which a robot has to localize itself based on camera images. One question that strikes me, in this context, is why are majority of the approaches based on features rather than on a direct comparison of the individual pixels. At least in my opinion, one of the reasons for this is that pure images are a high-dimensional input for which it is hard to define a proper likelihood function and, additionally, the

dependencies between the individual pixels are unknown. Most of the sensor models found in the literature, in fact, assume that the individual features extracted in images or range scans are independent—a fact that is rarely true especially when the posterior itself can only be approximated.

One of the open problems in the context of model design lies in the development of sound models that appropriately take the different properties of the measurement devices into account. One potential solution lies in the treatment of these problems as optimization problems in which one seeks to learn optimal model parameters based on the sensor data. We need the appropriate techniques, especially for high-dimensional measurement spaces, to learn about such models based on relatively sparse input data.

### High-Dimensional State Spaces

A further important aspect in the context of probabilistic algorithms is the ability to deal with high-dimensional state spaces. For example, problems such as multiobject tracking, SLAM, and mapping involve high-dimensional state spaces. As the size of the state space typically grows exponentially in the number of dimensions, appropriate techniques are needed to efficiently approximate the full posterior and update it accordingly. Again, a typical approach to overcome this curse of dimensionality is to make independent assumptions, e.g., to consider the marginal distributions instead of the full posterior. Questions such as when are these assumptions justified and what are the implications of these assumptions have not been answered sufficiently in the past. These are the real open questions, and answers to them will improve definitely the robustness of probabilistic approaches and robotic systems.

### SLAM

People have repeatedly argued that the problem of SLAM is solved. I believe that in the past years our community has gained a fundamental understanding of the SLAM problem and has developed a series of innovative solutions that can now be applied to reliably build maps of even large-scale environments. However, I do not believe that the SLAM community in the near future will encounter a novel breakthrough in core technology. Rather, there will be advances in certain branches of the SLAM problem, and those advances will be based on probabilistic approaches. Potential fields and more innovative solutions lie in the areas of dynamic environments, deformable worlds, three-dimensional (3-D) representations, object-oriented

- ◆ Organized Session “Safety, Security, and Rescue Robot Systems” at the International Conference on Instrumentation, Control, and Information Technology (SICE2008 in Tokyo) technically cosponsored by IEEE/RAS. Coorganizers were Matsuno Fumitoshi, TC cochair, Daniele Nardi, TC cochair, and Richard Voyles, TC cochair.
- ◆ Special Issue of *Journal of Advanced Robotics* (scheduled for July 2009) on “Disaster Response Robotics.” Guest coeditors are Satoshi Tadokoro, TC emeritus chair; Matsuno Fumitoshi, TC cochair; Daniele Nardi, TC cochair; Adam Jacoff, NIST (USA).

The TC also aggressively updated its membership list and has been alerting members of activities throughout the year. The current membership list is 1,003 strong. A succession list of cochairs is being developed to keep the leadership fresh and involved. An exciting new plan is being developed to migrate the SSRR TC to become one of the first proposed TCs as the next stage in the life of this

vibrant TC. The TC is sustaining an annual international workshop, as well as numerous other activities, so it is well poised to advance to a new level of permanence.

## Six-Axis Force/Torque Sensors

Custom sensors available.



**Standard Features:** Six Axes of Force/Torque Sensing (Fx Fy Fz Tx Ty Tz) • High Overload Protection • Interfaces for PCI, cPCI, PCMCIA (laptop), USB and more • Sizes from 17 mm - 250 mm diameter.

**Applications:** Product Testing • Biomedical Research • Finger Force Research • Rehabilitation Research • Robotics



**ATI INDUSTRIAL AUTOMATION**  
ISO 9001 Registered  
*Engineered Products for Robotic Productivity*

[www.ati-ia.com/ft](http://www.ati-ia.com/ft)

ROBOTIC END EFFECTORS

## POSITION

*(continued from page 8)*

representations, lifelong SLAM, and the integration of planning and SLAM.

My suggestion would be to focus on the aforementioned problems rather than on tiny aspects of existing SLAM algorithms. Instead of studying additional submapping approaches in SLAM or more efficient variants of particle filters, the robotics community requires techniques for real applications such as the lifelong coexistence of robots with humans in

domestic homes, navigation through rain forests, or dealing with objects and complex 3-D structures.

### Summary

At least in my opinion, probabilistic techniques will continue to be the most robust approach to state estimation problems. They will serve as a powerful tool for understanding problems and the approximation of their optimal solutions.

## Summer School on Robot Learning

By Paolo Fiorini and Erwin Prassler

The fourth IEEE Robotics and Automation Society/International Federation of Robotics (RAS/IFR) Summer School of Robotics Science on robot learning was held in Lazise (on the shores of the Lake of Garda), Italy, on 24–28 September 2007. The previous summer schools included topics such as haptic interaction (Paris, France, 2006), robot design (Tokyo, Japan, 2005), and human robot interaction (Volterra, Italy, 2004). This series of events is made possible through the cosponsorship of the RAS and the International Foundation of Robotics Research (IFRR). The aim of the school is to provide high-quality education in a specific topic by inviting distinguished researchers in the field, who provide a mixture of lectures and hands-on experiments.

This year's venue was the medieval customs house (Dogana Vecchia) in Lazise, which provided an excellent setting for the school, with its little harbor, coffee shops, and restaurants, all around the conference hall.

The full-day program started with the concepts on machine learning (ML), presented by Ivan Bratko and Janez Demsar, University of Ljubljana, Slovenia. In the afternoon, students had the opportunity of testing ML algorithms on case studies using the tool *Hyper* developed in Ljubljana.

In the morning of the second day, robotic learning (RL) was introduced by Erwin Prassler of Bonn-Rhein-Sieg University of Applied Sciences and by Ulrich Nehmzow of University of Essex. Unlike previous summer schools, this year, the hands-on experiments were organized around a common platform, the educational mobile robot *Eddy*, developed at the University of Verona. Therefore, in the afternoon of the second day, Monica Reggiani, University of Verona, gave a presentation on the robot and led the students in some preliminary RL exercises.

Three specific RL paradigms were presented during the rest of the week. Jan Peters of the Max-Planck Institute of Biological Cybernetics, Germany, presented a lecture on policy learning, Jun Tani of the RIKEN Brain Science Institute, Japan, discussed supervised learning. The last lecture was given by Ulrich Nehmzow, who spoke on novelty detection.

Thursday afternoon and Friday morning were devoted to the hands-on experiments. A set of specific exercises was developed and tested in advance on the mobile robot *Eddy*, thanks to the lecturers' collaboration. Furthermore, the software necessary to run the experiments was made available on the Web page of the school. The students could prepare the exercises before



Digital Object Identifier 10.1109/MRA.2008.921535

### Call for Nominations: 2008 AdCom Elections

**N**ominations of candidates for the 2008 Administrative Committee (AdCom) elections must be submitted by 1 July 2008. Nominators (or self-nominations) should indicate the position for which the nominee is being considered and the nominee's agreement to serve if elected.

Six of the 18 elected members of the AdCom are elected each year to serve a three-year term. All graduate student members and higher-grade members of the Society are eligible to vote.

Nominees will be considered by the Nominations Committee in consultation with the Advisory Committee. Petition candidates who acquire at least 121 signatures (2% of 31 December 2007 RAS voting members) will automatically be included on the ballot.

A slate of candidates will be announced prior to the elections via the society Web page, e-newsletter, and magazine. Society members may cast their ballots electronically or through post. All members with valid e-mail addresses will be sent electronic ballots, but request for paper ballots may also be made. All others will be sent paper ballots by post. Those who receive paper ballots will receive instructions with their ballots for casting their vote electronically if they choose.

Send in your nominations to the Society administrator at [r.g.snyder@ieee.org](mailto:r.g.snyder@ieee.org).

***The aim of the school is to provide high-quality education in a specific topic by inviting distinguished researchers in the field, who provide a mixture of lectures and hands-on experiments.***

On Wednesday evening, a bus tour to Verona was organized, where the school social dinner took place.

Registration cost was kept at a reasonable level, thanks to the financial sponsorship of the European project XPERO, whose support we gratefully acknowledge. In addition, students and researchers of the ALTAIR laboratory of the University of Verona took care of arrangements, logistics, etc. Without their help, the event would not have been possible. For additional information visit: <http://metropolis.sci.univr.it/summerschool07/>.

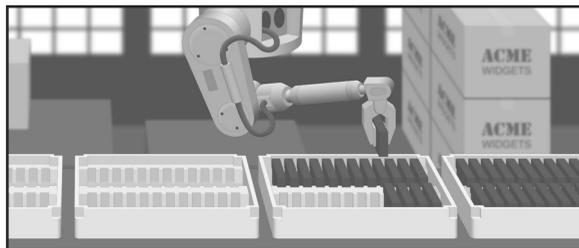
coming to Lazise, and during school, they practiced the lecture material without delay.

An open call for student participation was issued in June 2007. By the deadline, 40 applications were received. On the basis of an analysis of student motivation, research experience, and recommendations from their supervisors, a set of 35 students were admitted to the school. The students were geographically diverse, with three students from Japan, one from Australia, one from Canada, and 29 from Europe (Austria, Belgium, Canada, Denmark, Estonia, France, Germany, Greece, Italy, Netherlands, Spain, Sweden, and the United Kingdom). One admitted student from China could not attend because of visa problems. Of the admitted students, eight were women. The students' backgrounds covered mechanical engineering, computer science, and mechatronics and topics such as psychology and philosophy. Hence, the school provided a good mix of disciplines, nationalities, and gender. The students were extremely active and highly motivated.

On Wednesday afternoon, a public event was held to present robotics learning to the local community and to expose the students to topics that were not presented during the lectures. Thus, Jun Tani and Ulrich Nehmzow presented additional aspects of their research and Maarja Kruusmaa, Tallin University of Technology (Estonia), and Rüdiger Dillmann, Universität Karlsruhe (Germany), discussed safety and learning and imitation learning.

## **MATERIAL HANDLING TRAYS FOR PARTS PROCESSING**

- **Clean room compatible**
- **Static dissipative certification**
- **Smooth surfaces**
- **Low mold costs for custom designs**
- **Rapid turnaround**
- **Low quantities available**



**TEMPO**  
**PRECISION MOLDED FOAM**

[www.temp-foam.com](http://www.temp-foam.com) 559-651-7711

## Safety, Security, and Rescue Robotics

By Cochairs Daniele Nardi ([nardi@dis.uniroma1.it](mailto:nardi@dis.uniroma1.it)), Università di Roma; Matsuno Fumitoshi ([matsuno@hi.mce.uec.ac.jp](mailto:matsuno@hi.mce.uec.ac.jp)), University of Electro-Communications, Japan; and Richard Voyles ([rvoyles@du.edu](mailto:rvoyles@du.edu)), University of Denver

The Technical Committee (TC) on Safety, Security, and Rescue Robotics (SSRR) was organized in 2002 to help stimulate and coordinate research and development of robotics, automation, and intelligent devices and systems for civilian safety and rescue applications in the wake of natural and man-made disasters such as tornadoes, earthquakes, floods, fires, and explosions.

The SSRR TC has been extremely active this past year, as in previous years. Recent organized activities include the following:

- ◆ IEEE International Conference on Intelligent Robots and Systems (IROS) 2007 Workshop on Rescue Robotics, organized by TC Emeritus Chair Satoshi Tadokoro, Matsuno Fumitoshi, TC cochair, Hajime Asama, University of Tokyo (Japan), Koichi Osuka, Kobe University (Japan), and Masahiko Onosato, Hokkaido University (Japan).
- ◆ Special issue on SSRR for *Journal of Field Robotics (JFR)*, organized by Cochairs Richard Voyles and Howie Choset. According to Sanjiv Singh, *JFR* editor, this issue has three of the most downloaded articles in the history of the journal.
- ◆ The Fifth IEEE International Workshop on SSRR in Rome, Italy, 2007. Conference chair was Daniele Nardi, TC cochair. Program cochairs were Richard Voyles, TC cochair, Satoshi Tadokoro, TC emeritus chair, and Anibal Ollero, University of Seville (Spain).
- ◆ Planning for the Sixth IEEE International Workshop on SSRR in Sendai, Japan, 2008 is underway. Conference chair is Satoshi Tadokoro, TC emeritus chair. Discussion of the location for 2009 in the United States (likely Denver, Colorado) has been initiated.
- ◆ Rescue Robotics Exercise 2007 was held in Rome, Italy, to correspond with the International Workshop. This was organized by Daniele Nardi, TC cochair, and was held at the Istituto Superiore Antincendi (Italian National Firefighter Training Center). This exercise brought together robotics researchers, corporate developers, and emergency responders to demonstrate, operate, and critique robotic equipment and novel operator interfaces intended for safety, security, and rescue robotics applications. Approximately 50 researchers and 50 emergency responders attended.
- ◆ Rescue Robotics Camp 2007 was also held and organized by Daniele Nardi, TC cochair. This camp, intended

for robotics students and researchers, schooled participants on the needs and challenges of rescue robots and the current state of the art of rescue technology. Approximately 50 young scientists and engineers attended this important event for keeping the pipeline of researchers full.

- ◆ Special issue of *JFR* on “Quantitative Performance Evaluation of Robotic and Intelligent Systems” was organized by Raj Madhavan, National Institute of Standards and Technology (NIST) (USA).



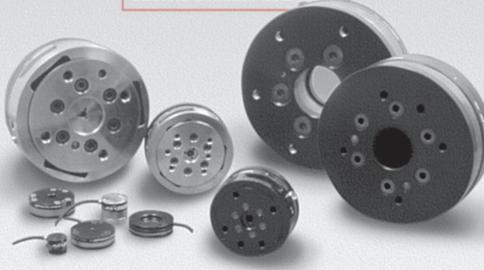
- ◆ Organized Session “Safety, Security, and Rescue Robot Systems” at the International Conference on Instrumentation, Control, and Information Technology (SICE2008 in Tokyo) technically cosponsored by IEEE/RAS. Coorganizers were Matsuno Fumitoshi, TC cochair, Daniele Nardi, TC cochair, and Richard Voyles, TC cochair.
- ◆ Special Issue of *Journal of Advanced Robotics* (scheduled for July 2009) on “Disaster Response Robotics.” Guest coeditors are Satoshi Tadokoro, TC emeritus chair; Matsuno Fumitoshi, TC cochair; Daniele Nardi, TC cochair; Adam Jacoff, NIST (USA).

The TC also aggressively updated its membership list and has been alerting members of activities throughout the year. The current membership list is 1,003 strong. A succession list of cochairs is being developed to keep the leadership fresh and involved. An exciting new plan is being developed to migrate the SSRR TC to become one of the first proposed TCs as the next stage in the life of this

vibrant TC. The TC is sustaining an annual international workshop, as well as numerous other activities, so it is well poised to advance to a new level of permanence.

## Six-Axis Force/Torque Sensors

Custom sensors available.



ROBOTIC END EFFECTORS

**Standard Features:** Six Axes of Force/Torque Sensing (Fx Fy Fz Tx Ty Tz) • High Overload Protection • Interfaces for PCI, cPCI, PCMCIA (laptop), USB and more • Sizes from 17 mm - 250 mm diameter.

**Applications:** Product Testing • Biomedical Research • Finger Force Research • Rehabilitation Research • Robotics



**ATI INDUSTRIAL AUTOMATION**  
ISO 9001 Registered  
Engineered Products for Robotic Productivity

[www.ati-ia.com/ft](http://www.ati-ia.com/ft)

## POSITION

(continued from page 8)

representations, lifelong SLAM, and the integration of planning and SLAM.

My suggestion would be to focus on the aforementioned problems rather than on tiny aspects of existing SLAM algorithms. Instead of studying additional submapping approaches in SLAM or more efficient variants of particle filters, the robotics community requires techniques for real applications such as the lifelong coexistence of robots with humans in

domestic homes, navigation through rain forests, or dealing with objects and complex 3-D structures.

### Summary

At least in my opinion, probabilistic techniques will continue to be the most robust approach to state estimation problems. They will serve as a powerful tool for understanding problems and the approximation of their optimal solutions.

## RoboticsCourseWare.org: An Open Repository for Robotics Pedagogical Materials

By Aaron M. Dollar, Daniela Rus, Paolo Fiorini

During the latter part of 2007, with funding from the IEEE Robotics and Automation Society's "New Initiatives" program, we developed RoboticsCourseWare.org, an open repository for robotics pedagogical materials. We created this site primarily for providing a resource to faculty to facilitate the creation of new robotics courses and the improvement of existing ones. The repository is a free and open educational resource for faculty, students, and hobbyists throughout the world. By providing easy access to teaching materials, we hope to facilitate the introduction of robotics courses everywhere from large institutions with established robotics programs to small colleges, ultimately transforming robotics into a core component of computer science and engineering academic programs.

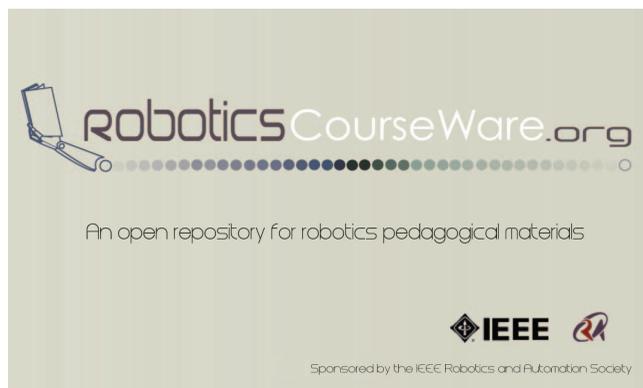
We developed the repository based on feedback from participants at the Robotics Education Workshop at the 2005 Robotics: Science and Systems Conference. More than 30 robotics faculty members from a wide range of institutions and backgrounds met for a day of presentations and discussions focusing on the key issues of integrating robotics in an undergraduate curriculum. A key outcome of the meeting was the general agreement as to the need for an open repository of robotics course materials to enable, support, and coordinate the teaching of robotics across universities. More on the outcome of this workshop can be found in the March 2006 "Education" column of *IEEE Robotics and Automation Magazine*.

### Structure and Contents

RoboticsCourseWare.org is similar to MIT's OpenCourseWare (OCW) initiative, which has since expanded to more than 100 institutions worldwide (<http://www.ocwconsortium.org>). However, unlike the majority of OCW sites, our repository is subject specific and contains materials from many universities.

The repository is searchable, browsable, and open for downloads. No registration or login is required for accessing the posted materials. Materials are typically made available under a Creative Commons License (<http://creativecommons.org>), under which the end users are free to copy and distribute the content, as well as adapt it for their own uses, provided proper attribution is given and the material is not used for commercial purposes.

In developing and populating the site, we have prioritized providing materials in formats that can be easily modified and reused. These materials are intended to cover the range of



primary areas of robotics pedagogy, including robot mechanics, control, motion planning, vision, and localization, with less emphasis on secondary areas and courses in which robotics is used as platform to teach concepts in other academic areas.

As of the writing of this column, we have published materials for four courses: Introduction to Autonomous Mobile Robots (Roland Siegwart, EPFL); Robotics: Science and Systems (Daniela Rus, Nick Roy, and Seth Teller, MIT); Introduction to Robotics (Rob Wood, Harvard); Motion Planning and Applications (David Hsu, National University of Singapore). Materials available for these courses include lecture slides and notes, course exercises, examinations, laboratory projects, code repositories, videos, and other media.

We also created a user forum associated with the site to allow the end users to communicate with each other. Among other things, this capability will serve as a help forum for users to assist each other with questions concerning the posted course materials.

### Contributions

We are actively seeking high-quality course materials for posting on the site. Currently, we are limiting the scope of potential contributions to university-level courses created by established robotics researchers. Final published materials must conform to the terms of an open-access license such as that described earlier. Interested parties should contact one of the authors of this article or send e-mail to [submissions@roboticscourseware.org](mailto:submissions@roboticscourseware.org). The editors will collaborate with potential contributors to ensure that the published materials meet appropriate accessibility and quality standards, such as the removal of copyrighted material.

It is our ultimate hope that RoboticsCourseWare.org will become a hub for robotics education that will expand to incorporate many additional resources. Along these lines, we encourage members of the robotics community who have developed resources related to robotics pedagogy that may be of interest to educators in higher education to contact us about the possibility of cross-linking materials or other similar collaborations.

**Aaron M. Dollar** is with the Harvard/MIT Health Sciences & Technology Biomechatronics Laboratory and the Massachusetts Institute of Technology (MIT) Media Lab.

**Daniela Rus** is with MIT. She is a cochair of the Robotics and Automation Society Education Committee.

**Paolo Fiorini** is with the University of Verona, Italy. He is a cochair of the Robotics and Automation Society Education Committee.

**Address for Correspondence:** Aaron M. Dollar, 20 Ames Street, Room E15-421, Cambridge, MA 02139. E-mail: [adolmar@mit.edu](mailto:adolmar@mit.edu).

# HUMAN-ROBOT INTERACTION 2009

4th ACM/IEEE International Conference, March 11-13, 2009, San Diego, CA, <http://www.hri2009.org>



# 2009



Technical Co-Sponsors

## Interacting Naturally with Robots

The Fourth Annual Conference on Human-Robot Interaction is dedicated to the advancement of natural human-robot interaction, which highlights the importance of the technical and social issues underlying future long-term human-robot interaction, in the context of companion and assistive robots for long-term use in everyday life and work activities.

HRI is a single-track, highly selective annual international conference that seeks to showcase the very best interdisciplinary and multidisciplinary research in human-robot interaction with roots in social psychology, cognitive science, HCI, human factors, artificial intelligence, robotics, organizational behavior, anthropology and many more, and we invite broad participation. The conference offers submission opportunities in the following categories: full and short papers, video submissions, tutorials and workshops, and exhibitions.

## Important Dates

- |                   |  |
|-------------------|--|
| 15 September 2008 | Submission of full papers, and tutorial/workshop proposals |
| 01 December 2008  | Submission of videos                                       |
| 12 December 2008  | Notification of acceptance                                 |
| 15 December 2008  | Submission of late-breaking short papers                   |
| 12 January, 2009  | Final camera-ready papers due                              |



SIGCHI

Sponsors



# Capturing an Evader in a Building

## Randomized and Deterministic Algorithms for Mobile Robots

A three-dimensional (3-D) grid  $G_{n \times n \times n}$ ,  $n \geq 2$ , is the set of points (vertices) with integer coordinates in  $[0, n-1] \times [0, n-1] \times [0, n-1]$  together with their connecting edges, which is viewed as a connected 3-D set [see Figures 1 and 2(a)]. Alternatively,  $G_{n \times n \times n}$  can be viewed as the union of  $2n^2$  horizontal line segments, called *corridors*, and  $n^2$  vertical line segments, called *shafts* (see the “Preliminaries” section for a formal definition).

We view  $G_{n \times n \times n}$  as representing a building and consider a vision-based pursuit-evasion problem in which a group of mobile robots (*pursuers*) are required to search for and capture an *evader* (*intruder*) hiding in it. The robots and the evader—all called *players*—are represented by points that move continuously along the edges of  $G_{n \times n \times n}$ . (Two players can be at the same point at one time.) Any continuous move in  $G_{n \times n \times n}$  is allowed within the speed limit constraint, which is 1 for the evader without loss of generality, and a constant  $s$  for the robots. The evader is considered captured if there exists a time during the pursuit when his position coincides with the position of one of the robots.

The vision of the players is limited to a straight line of sight (i.e., a shaft or a corridor): a player at a vertex can see the corresponding shaft and corridors, whereas the one located in the interior of an edge can see only the shaft or corridor containing that edge. (A player does not block the

view of another.) A player is said to have a *direction-detection capability* if he can see in which direction an opponent moves (left, right, up, or down) when disappearing from the line of sight. A *distance-detection capability* is one that allows a player to know the distance between his current location and that of an opponent in sight. In a realistic scenario, the cost of the used technology must be taken into account, and thus, we assume that mobile robots have no direction-detection capability, and their distance-detection capability is limited: they can only detect whether or not an evader in sight is within distance 1 of their current positions. In contrast, we assume that the evader may have both direction-detection and exact distance-detection capabilities. As for communication, we assume that the robots can communicate with one another in real time (without delay); they can also initiate and start executing any movement without delay. The evader may know the algorithm of the robots and their initial positions, but he does not know the outcomes of their random choices, in case they use a randomized algorithm.

The work presented in this article can be viewed as an extension of a recent work on a similar pursuit-evasion problem in a two-dimensional (2-D) grid  $G_{n \times n}$  [5]. The 2-D case was discussed first by Sugihara and Suzuki [13] as a variant of the well-known *graph search problem* [7], [8], [11], which is essentially the same problem except that it is played in an arbitrary connected graph by “blind” pursuers and an evader having an unbounded speed. One of the results presented in [13] is a deterministic algorithm for capturing an arbitrary evader having a



©DIGITAL VISION

Digital Object Identifier 10.1109/MRA.2008.921544

BY ICHIRO SUZUKI AND PAWEŁ ŻYLIŃSKI

maximum speed of 1, using four pursuers having a maximum speed of  $s = 1$ . The number of deterministic pursuers with  $s = 1$  necessary to capture the evader was later reduced to three [5]. Also presented in [5] are randomized capture algorithms that use one or two pursuers having a maximum speed of 1 or  $1 + \varepsilon$  for small  $\varepsilon > 0$ . Studies have also been conducted on the capabilities of a single pursuer having a much higher speed. Dawes [4] shows that a single pursuer having a speed of  $n$  can detect (i.e., see) an arbitrary evader in  $G_{n \times n}$ , and Neufeld [9] improves the speed bound to  $\lceil 2(n-1)/3 \rceil + 2$ . Dawes also discusses a strategy for a pursuer having a speed of at least  $(n-1)/k$ ,  $k \geq 1$ , capture the evader after locating him, assuming a certain “weak” distance-detection capability for the pursuer. A variation in which all players have “full vision” and thus know the positions of the others at all times in a 2-D grid has been considered in [12].

Various discrete pursuit-evasion games in an arbitrary graph, where the moves are restricted to discrete time steps and to the vertices of the graph, have been considered in the literature under the names of *cops and robbers* and *hunter and rabbit* with either full vision on both parties [2], [3], [10] or limited vision [1], [6]. For instance, it has been shown that under the assumption of no vision, a single pursuer can catch an evader with a nonzero (1 over a polynomial in  $n$ ) probability of success on any connected graph  $G$  having  $n$  vertices in expected  $O(n \log(\text{diam}(G)))$  time units, where  $\text{diam}(G)$  is the diameter of  $G$  [1]. If the players have “one-edge visibility” (i.e., they can see each other and, hence, have complete knowledge about each other’s location, if and only if they occupy adjacent vertices), then two pursuers are always sufficient to catch an evader with a nonzero (1 over a polynomial in  $n$ ) probability of success in any connected graph having  $n$  vertices in expected  $O(n^5 (\log n)^2)$  time [6]. Obviously, with higher-vision capabilities, the pursuers may be able to find the evader more easily, but at the same time, the evader will more easily see the pursuers and avoid a close encounter with them.

## Our Results

We first present a randomized algorithm by which one mobile robot with a maximum speed of  $s \geq 1$  can locate an evader in  $G_{n \times n \times n}$  in expected  $O(n^2)$  time. Next, we focus on randomized capturing strategies. In particular, we present a randomized algorithm by which six robots having a maximum speed of  $s = 1$  can capture an evader in  $O(n^3)$  expected time; if  $s = 1 + \varepsilon$ ,  $\varepsilon > 0$ ,

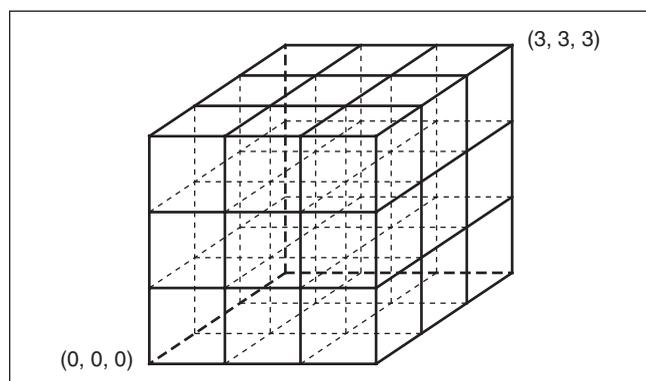


Figure 1. A grid  $G_{4 \times 4 \times 4}$ .

then five robots are enough. We then adopt the approach to use  $f(n) \geq 6$  robots and provide a randomized algorithm with an expected capture time of  $O(n^2 / (f(n)(1 - (1 - \frac{1}{6n})^{\lfloor \frac{f(n)}{4} \rfloor})))$ . All the aforementioned capturing strategies are based on forming a trap and making it smaller while keeping the evader in it, until one of the robots is close enough to the evader to start direct chasing, which ends in a capture. Next, using the idea of hiding from an evader, we provide a randomized algorithm by which two mobile robots having a speed of  $s = 1 + \varepsilon$ ,  $\varepsilon > 0$  can capture an evader in  $O(n^5)$  expected time (i.e., we only require the robots can move slightly faster than the evader). Finally, we present a deterministic  $O(n^3)$ -time strategy that uses  $\lfloor (4n-2)/3 \rfloor + 6$  robots with a maximum speed of  $s = 1$  (see Table 1 for a summary of these results).

To our knowledge, the continuous 3-D pursuit-evasion problem has not been discussed in the literature. Although some of the techniques we use here are adopted from our work on the 2-D case [5], modifications that are necessary to go from 2-D to 3-D are often quite complicated and nontrivial.

The rest of the article is organized as follows. In the next section, we define  $G_{n \times n \times n}$  formally and introduce a useful strategy called *chasing*. A randomized strategy for locating an evader by a single robot is presented in the “Detecting an Intruder” section. In the sections on capturing, we present randomized capturing algorithms, one based on the use of a moving trap and another based on hiding. Finally, a deterministic capturing algorithm that also uses a moving trap is presented.

## Preliminaries

Our pursuit-evasion game is played in a building represented by the grid  $G_{n \times n \times n}$ , which is the union of the following  $3n^2$  line segments [see Figure 2(a)]:

- 1) the line segment between  $(i, j, 0)$  and  $(i, j, n-1)$  called *shaft*  $(i, j)$ ,  $0 \leq i, j \leq n-1$
- 2) the line segment between  $(i, 0, k)$  and  $(i, n-1, k)$ , called *north-south corridor*  $(i, k)$ ,  $0 \leq i, k \leq n-1$ ; (we

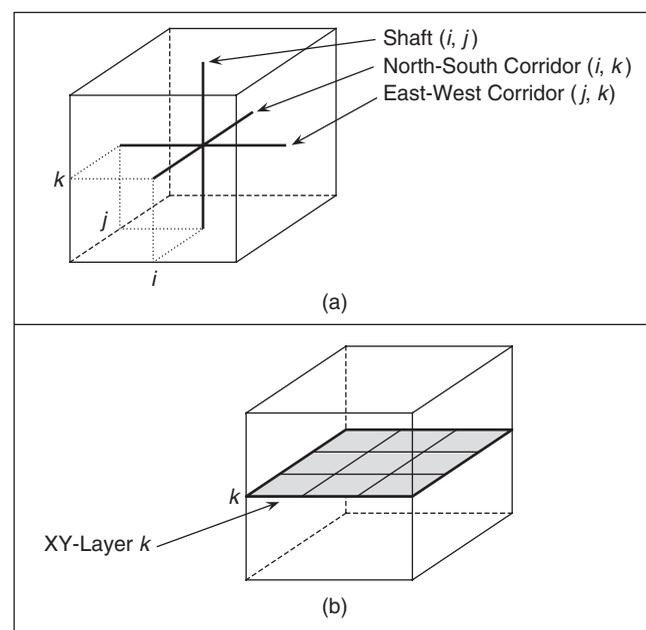


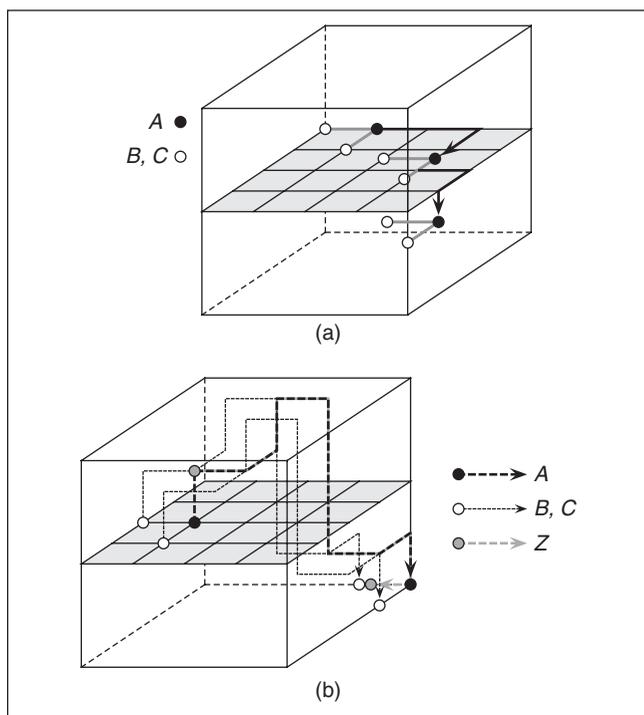
Figure 2. (a) Shafts and corridors. (b) XY-layer  $k$ .

use *east*, *west*, *north*, and *south* to refer to the  $+x$ ,  $-x$ ,  $+y$ , and  $-y$  directions, respectively.)

- 3) the line segment between  $(0, j, k)$  and  $(n-1, j, k)$ , called *east-west corridor*  $(j, k)$ ,  $0 \leq j, k \leq n-1$ .

For a given  $k = 0, \dots, n-1$ , the line segments in 2) and 3) constitute the *XY-layer*  $k$  [see Figure 2(b)]. Analogously, for a given  $j = 0, \dots, n-1$ , the line segments 1) and 3) constitute the *XZ-layer*  $j$ , while the line segments 1) and 2) constitute the *YZ-layer*  $i$  for a given  $i = 0, \dots, n-1$ . According to the definition, we have  $3n$  different layers.

In the rest of the article, we denote the robots by  $A, B, \dots$ , and the evader by  $Z$ . We use shortest path  $L_1$ -distances in  $G_{n \times n \times n}$  for measuring distances between the robot(s) and evader; e.g., the distance between  $(0, 2/3, 0)$  and  $(1, 2/3, 1)$  is  $8/3$ .



**Figure 3.** (a) Robots  $A, B$ , and  $C$  are moving in a triangle formation. (b) Robots  $A, B$ , and  $C$  capture  $Z$  when  $Z$  attempts to cross an edge westward, with  $A$  and  $B$  on the east and west endpoints of the edge, respectively.

## Chasing

The following scheme called *chasing* was found useful for the robots to possibly capture the evader in the 2-D case [5], [13]. We adopt it here for the 3-D case.

We say that robot  $A$  with a maximum speed of  $s \geq 1$  *chases* evader  $Z$  if he continuously moves toward  $Z$  at a speed of  $s$  after seeing  $Z$  within distance  $s$ . Note that chasing by a pursuer forces  $Z$  to continue to move forward to avoid an immediate capture. Although during a chase  $A$  may not know temporarily where  $Z$  is when  $Z$  disappears from the line of sight at a vertex  $v$ ,  $A$  will see  $Z$  again within distance 1 when he reaches  $v$ , because  $A$  is within distance  $s$  of  $Z$  when  $Z$  reaches  $v$ . (Recall that  $A$  has no direction-detection capability.) Thus, if  $s = 1$ , then the distance between the two remains at most 1, and if  $s = 1 + \varepsilon$ ,  $\varepsilon > 0$ , then  $A$  catches up with  $Z$  within  $\frac{s}{\varepsilon} = O(1)$  time.

**Lemma 1:** In  $G_{n \times n \times n}$ :

- 1) If  $s = 1 + \varepsilon$ ,  $\varepsilon > 0$ , then a single robot can capture  $Z$  within  $\frac{s}{\varepsilon} = O(1)$  time after he starts chasing  $Z$ .
- 2) If  $s = 1$ , then three robots can capture  $Z$  within  $O(n^2)$  time after one of them starts chasing  $Z$ .

For convenience, in the following, we shall say “the robots start chasing  $Z$ ” to mean “one of the robots starts chasing  $Z$ .”

**Proof:** We already proved the first claim. For the second claim, while  $A$  chases  $Z$ , we first let  $B$  and  $C$  form a moving triangle formation with  $A$  as shown in Figure 3(a) so that  $B$  is west of  $A$  at distance 1, and  $C$  is south of  $A$  at distance 1. ( $B$  and  $C$  may get closer to  $A$  if  $A$  is near the grid boundary and hence, the formation must be “compressed”.) This can be accomplished in  $O(n)$  time if  $B$  and  $C$  regard their respective target positions (relative to  $A$ ’s position) as virtual evaders and “capture” them using the idea in an algorithm given in [12]. (Recall that the robots can communicate with each other without delay, and thus,  $B$  and  $C$  know where their target positions are at all times.) Once they form the triangle formation, it is easy to see that the robots can stop maintaining the formation and capture  $Z$  when, in  $O(n^2)$  time, one of the following occurs:

- ◆  $Z$  attempts to cross an edge westward (with  $A$  and  $B$  on the east and west endpoints of the edge, respectively), see Figure 3(b) for an illustration, or

Table 1. Summary of the main results.

Number of Pursuers	Robot’s Speed ( $s$ )	Duration	Probability of Locating	Expected Time to Locate	Probability of Capture	Expected Time to Capture
1	1	$O(n)$	$\frac{1}{6n}$	$O(n^2)$	–	–
1	1	$O(n)$	–	–	$\frac{1}{6n}$	$O(n^2)$
2	$1 + \varepsilon$	$O(n^3)$	–	–	$\frac{1}{n^2}$	$O(n^5)$
5	$1 + \varepsilon$	$O(n^2)$	–	–	$\frac{1}{6n}$	$O(n^3)$
6	1	$O(n^2)$	–	–	$\frac{1}{6n}$	$O(n^3)$
$\lfloor \frac{4n-2}{3} \rfloor + 6$	1	$O(n^3)$	–	–	1	$O(n^3)$
$f(n) = O(n)$	1	$O\left(\frac{n^2}{f(n)}\right)$	–	–	$1 - \left(1 - \frac{1}{6n}\right)^{\lfloor \frac{f(n)}{4} \rfloor}$	$O\left(\frac{n^2}{f(n)\left(1 - \left(1 - \frac{1}{6n}\right)^{\lfloor \frac{f(n)}{4} \rfloor}\right)}\right)$

- ◆  $Z$  attempts to cross an edge southward (with  $A$  and  $C$  on the north and south endpoints of the edge, respectively). ■

## Detecting an Intruder

Let us start with the task of detecting (i.e., seeing) an intruder  $Z$ . Consider the following procedure (see Figure 4 for an illustration). A similar procedure was used for the 2-D case in [5].

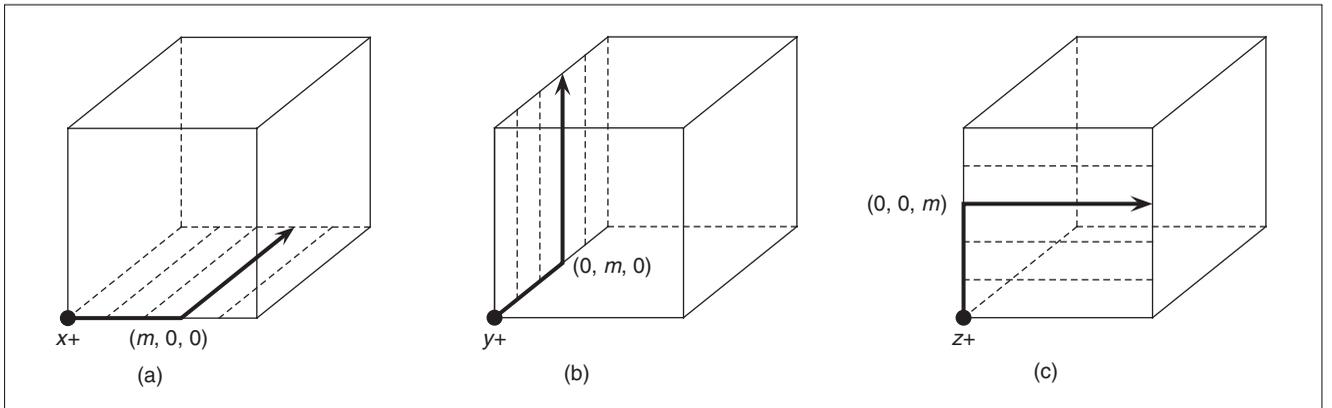
### Procedure Locate

- 1) Robot  $A$  moves to vertex  $(0, 0, 0)$  from its current location at maximum speed  $s$ . Time is reset to 0.
- 2)  $A$  selects a start time  $s \in [0, 4n]$  uniformly at random and stays at  $(0, 0, 0)$  until time  $s$ .
- 3) At time  $s$ , uniformly at random,  $A$  selects one of the three axis directions  $x+$ ,  $y+$ , or  $z+$ , and  $m \in \{0, n-1\}$ . Depending on the axis direction selected:
  - ◆  $x+$ :  $A$  starts moving from  $(0, 0, 0)$  in direction  $x+$  at maximum speed  $s$  and at vertex  $(m, 0, 0)$  turns into north-south corridor  $(m, 0)$  toward  $(m, n-1, 0)$ .
  - ◆  $y+$ :  $A$  starts moving from  $(0, 0, 0)$  in direction  $y+$  at maximum speed  $s$  and at vertex  $(0, m, 0)$  turns into shaft  $(m, 0)$  toward  $(m, 0, n-1)$ .
  - ◆  $z+$ :  $A$  starts moving from  $(0, 0, 0)$  in direction  $z+$  at maximum speed  $s$  and at vertex  $(0, 0, m)$  turns into east-west corridor  $(0, m)$  toward  $(n-1, 0, m)$ .

In the following, we show that the probability that  $A$  sees  $Z$  by executing `Locate` is at least  $\frac{1}{6n}$ . We need the following notation. For a time interval  $T = [t_1, t_2]$ ,  $t_1 \leq t_2$ , we denote by  $|T| = t_2 - t_1$  the length of  $T$ . For any  $t_3 \geq 0$ ,  $T - t_3$  denotes the interval  $[t_1 - t_3, t_2 - t_3]$  obtained by shifting  $T$  early by  $t_3$ .

Consider two time intervals,  $W = [0, 4n]$  and  $I = [2n, 4n] \subset W$ . Fix a move of  $Z$  in  $W$ , and for each pair of  $0 \leq i, j \leq n-1$ , let  $x_{i,j}$  be the total time in  $I$  during which  $Z$  is in shaft  $(i, j)$ . Similarly,  $y_{j,k}$ ,  $0 \leq j, k \leq n-1$ , denote the total time in  $I$  during which  $Z$  is in the east-west corridor  $(j, k)$ , while  $z_{i,k}$ ,  $0 \leq i, k \leq n-1$ , denote the total time in  $I$  during which  $Z$  is in the north-south corridor  $(i, k)$ . Obviously, we have

$$\sum_{\substack{0 \leq i \leq n-1 \\ 0 \leq j \leq n-1}} x_{i,j} + \sum_{\substack{0 \leq j \leq n-1 \\ 0 \leq k \leq n-1}} y_{j,k} + \sum_{\substack{0 \leq i \leq n-1 \\ 0 \leq k \leq n-1}} z_{i,k} \geq |I|. \quad (1)$$



**Figure 4.** An illustration of procedure `Locate`.

(At a vertex,  $Z$  is simultaneously in the shaft and the two corridors passing through it.)

Now, for some fixed  $i$  and  $j$ , let  $J_1, J_2, \dots, J_l$  be the disjoint maximal intervals in  $I$  in which  $Z$  is in shaft  $(i, j)$  where  $|J_1| + |J_2| + \dots + |J_l| = x_{i,j}$ . Suppose in Step 3  $A$  selects direction  $+x$  (out of three possible choices) and  $m = i$  (out of  $n$  possible choices). Since it takes exactly  $\frac{i+j}{s}$  time units for  $A$  to go from  $(0, 0, 0)$  to  $(i, j, 0)$ ,  $A$  will see  $Z$  in shaft  $(i, j)$  at the moment he reaches  $(i, j, 0)$  if  $A$  starts Step 3 at any time in any of  $J_1 - \frac{i+j}{s}, J_2 - \frac{i+j}{s}, \dots, J_l - \frac{i+j}{s}$ . Since these intervals are pairwise disjoint subintervals of  $W$  and  $A$  chooses the start time  $s$  uniformly randomly in  $W$ , the probability of the above event is

$$\frac{\sum_{1 \leq t \leq l} |J_t - \frac{i+j}{s}|}{|W|} = \frac{\sum_{1 \leq t \leq l} |J_t|}{|W|} = \frac{x_{i,j}}{|W|}.$$

Consequently, disregarding for now the possibility that  $A$  may see  $Z$  before reaching the north-south corridor  $(i, 0)$  or after reaching the destination,

$$\text{Prob}(A \text{ sees } Z \text{ in } I \text{ in shaft } (i, j) \text{ from } (i, j, 0)) = \frac{1}{3n} \cdot \frac{x_{ij}}{|W|},$$

and hence, the probability that  $A$  sees  $Z$  in  $I$  in any shaft from the north-south corridor  $(i, 0)$  is

$$\sum_{0 \leq j \leq n-1} \frac{1}{3n} \cdot \frac{x_{ij}}{|W|}.$$

(If  $s = 1$ , then  $A$  may see  $Z$  in multiple shafts while moving along the north-south corridor  $(i, 0)$ . However, for any given move of  $Z$ , the set of starting times for  $A$  that make this possible has measure zero, and hence, the above equation always holds.) Using a similar argument for axis directions  $+y$  and  $+z$  and summing all up, and also taking into account the possibility that  $A$  may see  $Z$  in other ways (before reaching the selected corridor or shaft, after reaching the destination, or outside  $I$ ), we obtain that the probability that  $A$  sees  $Z$  from a vertex by executing `Locate` is at least  $\frac{1}{6n}$  by (1). The next lemma summarizes this result.

**Lemma 2:** In  $G_{n \times n \times n}$ , with probability at least  $\frac{1}{6n}$ , a single robot with a maximum speed of  $s \geq 1$  can detect  $Z$  from a vertex within  $O(n)$  time.

Consequently, by repeating procedure `Locate`, we obtain the following theorem.

**Theorem 3:** Using a randomized algorithm, a single robot  $A$  with a maximum speed of  $s \geq 1$  can detect  $Z$  in  $G_{n \times n \times n}$  in  $O(n^2)$  expected time.

### Capturing by Forming a Trap

The capturing strategies discussed in this section are based on constructing a moving trap for evader  $Z$ , which—taking into account Lemma 1—aims to let one of the robots start chasing  $Z$ . The definition of a trap and its properties are given in the “Trap” section, and the “Forming a Trap and Capturing” section is devoted to a complete description of our capturing strategies using a trap. We also use a trap in a deterministic capturing algorithm.

The use of a trap has been the basis of the deterministic three-pursuer and four-pursuer algorithms for the 2-D case presented in [5] and [13]. However, as we will see below, extending the idea to the 3-D case is not straightforward: in particular, the process of forming a trap can be quite complicated, depending on the number of robots and their speed.

#### Trap

Suppose that four robots  $A, B, C,$  and  $D$  form a “star” formation centered at vertex  $(i, j, 0)$  in the bottom layer  $z = 0$ , by occupying vertices  $(i - 1, j, 0), (i, j - 1, 0), (i + 1, j, 0),$  and  $(i, j + 1, 0)$ , respectively. (Here and in the following, we assume that the star formation will be suitably “collapsed” if  $i$  or  $j$  equals 0 or  $n - 1$ . For instance, if  $(i, j, 0) = (0, 1, 0)$ , then  $A$  is at  $(0, 1, 0)$ , and if  $(i, j, 0) = (0, 0, 0)$ , then both  $A$  and  $B$  are at  $(0, 0, 0)$ .) We let

$$\begin{aligned} \text{TRAP}(i, j, 0) = \\ = \{(x, y, z) \mid i - 1 \leq x \leq i + 1, j - 1 \leq y \leq j + 1, z > 0\} \end{aligned}$$

be the region, called the *trap*, which lies above the star formation [see Figure 5(a)]. Using the operation described below, the robots 1) keep  $Z$  inside the trap by translating the formation in the current layer, and at the same time, 2) make the trap smaller by moving to upper layers with help from another

robot (or two, depending on the speed of the robots). At any time during the operation, if  $Z$  appears within distance 1 of any of the robots, then chasing starts immediately. (For succinctness of presentation, we do not mention all the possible situations in which this close encounter occurs.)

#### Keeping $Z$ in the Trap

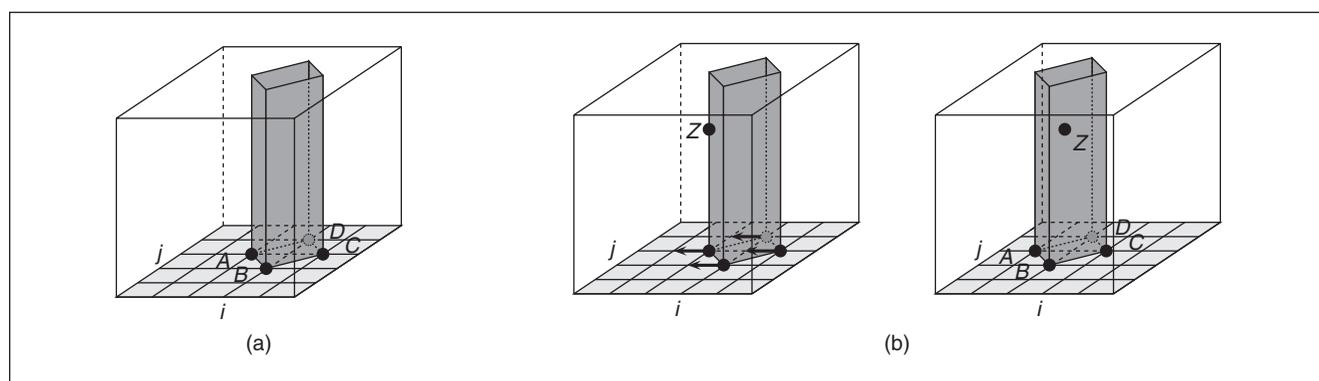
Suppose that  $A$  sees  $Z$  (i.e.,  $Z$  appears in the shaft  $(i - 1, j)$  above  $A$ ) at time  $t$  [see Figure 5(b)]. If  $Z$  is within distance 1 from  $A$ , then  $A$  starts chasing  $Z$ . If  $Z$  is at a distance greater than 1, then the robots simultaneously move west over one edge at speed 1 maintaining the formation, so that  $Z$  (having a maximum speed of 1) will be in  $\text{TRAP}(i - 1, j)$  at time  $t + 1$ . Similarly, if  $B, C,$  or  $D$  sees  $Z$  at time  $t$ , then the robots move the trap south, east, or north, respectively, to keep  $Z$  inside the trap at time  $t + 1$  (or chasing starts at  $t$ ).

#### Moving the Formation to Upper Layers

Let  $p$  be the center of the moving star formation in the bottom layer, and let  $p'$  be the point in the second layer right above  $p$ . (Both  $p$  and  $p'$  are moving points.) While  $A, B, C,$  and  $D$  keep  $Z$  inside the moving trap as explained earlier, robot  $E$  moves to the point in the second layer right above  $A$ . At this moment,  $E$  starts to play the role of  $A$  so that robots  $E, B, C,$  and  $D$  maintain the moving trap and keep  $Z$  inside it. Robot  $A$ , which is now free, moves to the point in the second layer right above  $B$  in a similar manner, so that  $E, A, C,$  and  $D$  maintain the trap. Continuing in a similar manner,  $B$  moves to the second layer right above  $C$ , and  $C$  moves to the second layer right above  $D$ , so that the four robots  $E, A, B,$  and  $C$  form a star formation centered at  $p'$ .  $D$ , which is now free, moves to  $p$  to see whether  $Z$  is hiding in the edge between  $p$  and  $p'$ ; if so, then  $D$  starts chasing  $Z$ . (If  $p$  moves while  $D$  is moving toward it, then  $Z$  must have become visible to one of the searchers in the star formation, and hence,  $Z$  could not have been in the edge between  $p$  and  $p'$  when  $D$  attempted to move toward  $p$ .) Now  $E, A, B,$  and  $C$  form a star formation in the second layer, with  $Z$  in

$$\begin{aligned} \text{TRAP}(i, j, 1) = \\ = \{(x, y, z) \mid i - 1 \leq x \leq i + 1, j - 1 \leq y \leq j + 1, z > 1\}, \end{aligned}$$

for some  $i$  and  $j$ . The robots then repeat the above operation and move the formation to the third layer using robot  $D$ ,



**Figure 5.** (a) A trap  $(i, j, 0)$ . (b) Keeping  $Z$  in the trap.

which is now free. By continuing in the same manner, the robots can make the trap smaller and smaller while keeping  $Z$  inside it. Thus, within  $O(n^2)$  time, one of the robots reaches a point within distance 1 of  $Z$  and starts chasing  $Z$ .

**Lemma 4:** *In the situation described above, let  $A'$  be the moving point in the second layer right above  $A$ . Robot  $E$  with a maximum speed of 1 can move to  $A'$  in  $O(n)$  time, with a help from another robot  $F$ . If  $E$  has a maximum speed of  $1 + \varepsilon$ ,  $\varepsilon > 0$ , then  $E$  can do so alone in  $O(n)$  time without  $F$ .*

**Proof:** Since the robots can communicate with each other in real time,  $E$  knows where  $A'$  is at all times.  $E$  regards  $A'$  as a virtual target in the second layer and captures it using a two-robot capture algorithm for a 2-D grid  $G_{n \times n}$  given in [12], with help from another robot  $F$  (both with a maximum speed of 1). Briefly, the algorithm mentioned above allows  $E$  to reach  $A'$  in  $O(n)$  time while keeping  $A'$  in one of the quadrants given by the two diagonals through  $E$ 's position, if  $F$  chases  $A'$  and forces it to continue to move forward. In the 3-D situation we have,  $F$  cannot force  $A'$  to move forward, and hence, we simply let  $F$  move toward  $A'$ . Then, within  $O(n)$  time, either  $A'$  moves enough and gets caught by  $E$ , or  $F$  reaches  $A'$ . Furthermore, we can show that even if robot  $F$  does not exist,  $E$  alone (even with a maximum speed of 1) can get within distance 2 of  $A'$  using a technique similar to the one outlined above. Once this is done, and if  $E$  has a maximum speed of  $1 + \varepsilon$ ,  $\varepsilon > 0$ , then  $E$  can catch up with  $A'$  within  $O(1)$  time by simply following the trajectory of  $A'$  at its maximum speed. ■

By the argument outlined above, we obtain the following theorem:

**Theorem 5:** *In  $G_{n \times n \times n}$ :*

- 1) *Six robots with a maximum speed of 1 can start chasing  $Z$  within  $O(n^2)$  time, once four of them form a trap with  $Z$  inside it.*
- 2) *Five robots with a maximum speed of  $1 + \varepsilon$ ,  $\varepsilon > 0$ , can start chasing  $Z$  within  $O(n^2)$  time, once four of them form a trap with  $Z$  inside it.*

Suppose that now there are  $f(n)$  robots, all having a maximum speed of 1, for some function  $f(n) = O(n)$  such that  $f(n) \geq 6$ . (As a matter of fact,  $f(n)$  ( $\geq 6$ ) may be an arbitrary positive function. However, for succinctness of presentation, we assume  $f(n) = O(n)$ . Otherwise, the total time to start chasing  $Z$ , after forming a trap with  $Z$  in it, is  $O(n^2/f(n) + n)$ . In particular, it is easy to see that  $n^2$  robots are enough to sweep the whole grid from the bottom to the top and deterministically capture an evader within  $O(n)$  time.) In this case, the total time to start chasing  $Z$ , after forming a trap with  $Z$  in it, can be reduced to  $O(n^2/f(n))$  time. The idea is to form a star formation in  $\Theta(f(n))$  consecutive layers (all vertically aligned) to maintain a moving trap with  $Z$  inside it, and then move all formations (except the top one) concurrently to the next cluster of  $\Theta(f(n))$  consecutive layers when making the trap smaller. We repeat this until the trap is small enough and a chasing starts. To be specific, we first use four robots to form a star formation in the bottom

layer, with  $Z$  in the trap, and then move the remaining  $f(n) - 4$  robots, divided into  $\lfloor (f(n) - 4)/2 \rfloor$  pairs, to suitable positions in upper layers so that we have vertically aligned star formations in consecutive  $\lfloor (f(n) - 4)/8 \rfloor$  layers above the bottom layer. (We use a pair of robots to move one robot to its destination, as discussed in the proof of Lemma 4.) Then, all pairs of robots, except those forming the topmost star formation, move concurrently to suitable positions in the upper layers to again form a stack of star formations right above the current topmost star formation, after making sure that  $Z$  is not hiding in the center shaft (of the star formations) below the topmost formation (which requires only one robot to move to the center of the bottom star formation and move up along the shaft if  $Z$  is seen; if the robot fails to reach the shaft because the star formation moves, then  $Z$  cannot be hiding in the shaft). This move takes a total of  $O(n)$  time. The number of times the stack of star formation must be moved to upper layers before the trap becomes small enough for some robot to start chasing  $Z$  is  $O(n/f(n))$ . Thus, a chasing starts within  $O(n^2/f(n))$  time after the formation of a trap.

**Theorem 6:** *In  $G_{n \times n \times n}$ , for any function  $f(n) = O(n)$  such that  $f(n) \geq 6$  (or  $f(n) \geq 5$ ),  $f(n)$  robots with a maximum speed of 1 (or  $1 + \varepsilon$ ,  $\varepsilon > 0$ , respectively) can start chasing an evader within  $O(n^2/f(n))$  time, once four of them form a trap with the evader inside it.*

### Forming a Trap and Capturing

Taking into account the results in Theorems 5 and 6, all we need is a strategy for the robots to form a trap with  $Z$  in it.

We use four robots having a maximum speed  $s \geq 1$ . By Lemma 2, with probability at least  $\frac{1}{6n}$ , the four robots, moving together as one team, can see  $Z$  from a vertex, say  $v$ , within  $O(n)$  time. The moment this happens, the robots spread into a star formation centered at  $v$  to form a trap with  $Z$  in it. Consequently, by Theorem 5 and Lemma 1, we obtain the following theorem.

**Theorem 7:** *In  $G_{n \times n \times n}$ , using a randomized algorithm, six (five) robots with a maximum speed of  $s \geq 1$  (of  $s > 1$ , respectively) can capture an evader in  $O(n^3)$  expected time.*

Furthermore, if  $f(n) = O(n)$  robots are available, then by dividing the robots into  $\lfloor f(n)/4 \rfloor$  independent teams, with probability at least  $1 - (1 - \frac{1}{6n})^{\lfloor \frac{f(n)}{4} \rfloor}$ , one team of robots locates an evader from a vertex within  $O(n)$  time, and thus, by Theorem 6 and Lemma 1, we obtain:

**Theorem 8:** *In  $G_{n \times n \times n}$ , using a randomized algorithm,  $f(n) \geq 6$  robots with a maximum speed of  $s \geq 1$  can capture an evader in  $O(n^2/(f(n)(1 - (1 - \frac{1}{6n})^{\lfloor \frac{f(n)}{4} \rfloor})))$  expected time.*

### Capturing by Hiding

Bearing in mind the purpose of minimizing the number of mobile robots, we continue our presentation of capturing techniques with a randomized strategy using only two robots having the maximum speed  $s = 1 + \varepsilon$ , for arbitrary  $\varepsilon > 0$ , with an  $O(n^5)$  expected time for a capture. (In the discussion below, we assume  $\varepsilon < 1$ ; however, it is easy to see that our approach can easily be modified for arbitrary  $\varepsilon > 0$ .)

At first glance, to capture the evader, it seems natural for the robots to first locate the evader and, then, attempt to capture it right away. In this section, we present a strategy in which, surprisingly, it is crucial for the robots to hide from the evader, i.e., the overall strategy of the robots is to first “hide” and “guard” one of the layers and then advance it toward  $Z$  until  $Z$  is forced to appear in a guarded layer. At that moment, with a probability of at least  $\frac{1}{n^2}$ , the robot guarding the layer is within distance of  $Z$  small enough to start chasing.

### Hiding

The general idea for a robot, say  $A$ , to hide is to choose one of the “distinguished” edges in a (XY-, XZ-, or YZ-) layer  $L$  uniformly at random, say edge  $e$ , and hide in the interior of  $e$  without letting  $Z$  know which edge he is in. If subsequently  $Z$  reaches layer  $L$  at some vertex  $v$ , then with a probability at least  $\frac{1}{n^2}$ ,  $v$  will be an endpoint of  $e$ , and thus,  $A$  will be within distance 1 of  $Z$ , which will allow  $A$  to start chasing  $Z$  at that moment.

Formally, for  $k = 0, 1, \dots, n - 1$ , let us define the set  $H_{XY}^k$  of hiding points as follows (see Figure 6 for an illustration):

$$H_{XY}^k = \bigcup_{j=0}^{n-1} \left\{ \left( \frac{\epsilon}{2}, j, k \right) \right\} \cup \bigcup_{j=0}^{n-1} \left\{ \left( i - \frac{\epsilon}{2}, j, k \right) : i = 1, \dots, n - 1 \right\}.$$

Observe that by the definition, we have  $|H_{XY}^k| = n^2$ , and for every vertex in XY-layer  $k$ , there exists a unique point in  $H_{XY}^k$  at distance  $\frac{\epsilon}{2} < 1$  on the same east-west corridor. (Later, we use the property that the hiding points cannot be seen along any north-south corridor.) Analogously, we define the sets  $H_{XZ}^j$  and  $H_{YZ}^i$  of hiding points, for  $j, i = 0, \dots, n - 1$ , respectively.

Our hiding procedure requires that the robots first detect  $Z$ . To achieve this, robots  $A$  and  $B$  repeatedly execute procedure Locate by always moving together. By Lemma 2, they can see

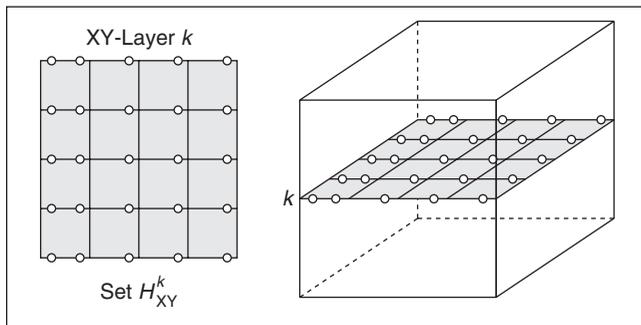


Figure 6. The set  $H_{XY}^k$  of hiding points.

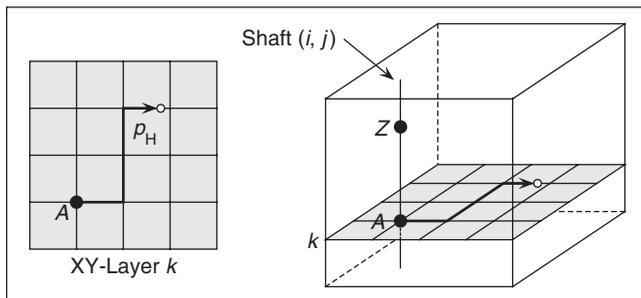


Figure 7. An illustration of procedure Hide.

$Z$  from a vertex in expected  $O(n^2)$  time. With loss of generality, suppose that robots  $A$  and  $B$  are at vertex  $(i, j, k)$  and see  $Z$  in shaft  $(i, j)$  at time  $t$  at distance  $> 1$  (an analogous approach can be applied when they see  $Z$  either in north-south corridor  $(i, k)$  or in east-west corridor  $(j, k)$ ). Then  $B$  moves straight toward  $Z$  to force  $Z$  to leave shaft  $(i, j)$ , while  $A$  waits at  $(i, j, k)$ . At the moment  $Z$  disappears from shaft  $(i, j)$ ,  $A$  uses the following procedure Hide to hide in XY-layer  $k$ ; when it is finished, we say that XY-layer  $k$  is guarded by  $A$  (see Figure 7 for an illustration).

### Procedure Hide

$A$  chooses a point  $p_H \in H_{XY}^k$  in XY-layer  $k$  uniformly at random, and immediately starts to move to  $p_H$  at speed  $s$  along one of the shortest paths ( $A$  can use any shortest path to  $p_H$ ).

**Lemma 9:** Suppose robot  $A$ , whose maximum speed is  $s > 1$ , starts procedure Hide at vertex  $(i, j, k)$  at time  $t$  and hides in XY-layer  $k$ . When  $Z$  reaches XY-layer  $k$  for the first time after  $t$ ,  $A$  is within distance 1 of  $Z$  with a probability of at least  $\frac{1}{n^2}$ .

**Proof:** There are only two possibilities for  $Z$  to gain some knowledge about  $A$ 's hiding position  $p_H$ . The first is to move quickly to some shaft  $(i', j')$ ,  $i' \neq i$  or  $j' \neq j$ , no later than when  $A$  would cross vertex  $(i', j', k)$ , and see if  $A$  indeed appears there. However, it is not possible for  $Z$  to do this because  $A$  moves faster than  $Z$ : for every  $0 \leq i', j' \leq n - 1$ ,  $A$  can reach  $(i', j', k)$  at time  $t + (|i - i'| + |j - j'|)/s < t + |i - i'| + |j - j'|$ , while  $Z$  cannot reach shaft  $(i', j')$  at least until  $t + |i - i'| + |j - j'|$ . The second possibility is to move quickly to XY-layer  $k$  at some vertex  $(i', j', k)$  no later than when  $A$  would cross either the north-south corridor  $(i', k)$  or the east-west corridor  $(j', k)$  and see if  $A$  indeed appears there. However, with a probability of  $\frac{1}{n^2}$ ,  $A$  has chosen the point in  $H_{XY}^k$  at distance  $\frac{\epsilon}{2}$  of  $(i', j', k)$  as  $p_H$  and reaches there before  $Z$  reaches  $(i', j', k)$  as discussed above. Thus, when  $Z$  enters XY-layer  $k$  for the first time, with probability at least  $\frac{1}{n^2}$ , robot  $A$  is within distance 1 of  $Z$  and starts chasing  $Z$ . ■

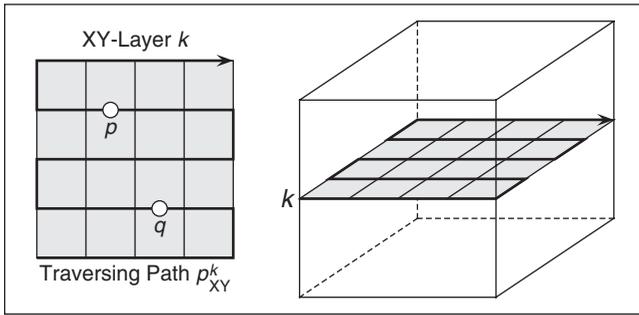
Note that when Hide is completed, XY-layer  $k$  is guarded, and the robots know on which side of the layer  $Z$  lies.

### Capturing

Once a layer is guarded, the robots advance the guarded layer toward  $Z$ , using procedure Traverse given below. First, we need some definitions.

For XY-layer  $k$ ,  $k = 0, \dots, n - 1$ , define the (directed) traversing path  $p_{XY}^k$  as a zigzag path that starts at  $(0, 0, k)$  and visits all vertices in the layer, by first going east to vertex  $(n - 1, 0, k)$ , then north to  $(n - 1, 1, k)$ , then west to  $(0, 1, k)$ , and so on, as shown in Figure 8. Clearly, if  $n$  is even, then  $p_{XY}^k$  finishes at vertex  $(0, n - 1, k)$ ; otherwise, it finishes at vertex  $(n - 1, n - 1, k)$ . For two points  $p, q \in p_{XY}^k$ , we say that  $p$  succeeds  $q$ , and  $q$  precedes  $p$ , if  $q$  appears before  $p$  in  $p_{XY}^k$  (see Figure 8). Finally, for every vertex  $v$  in XY-layer  $k$ , except the last one in  $p_{XY}^k$ , we denote by  $\text{succ}(v)$  the next vertex along  $p_{XY}^k$ .

Analogously, we define traversing paths  $p_{XZ}^j$  and  $p_{YZ}^i$ , for  $j, i = 0, \dots, n - 1$ , and the relevant successor relation.

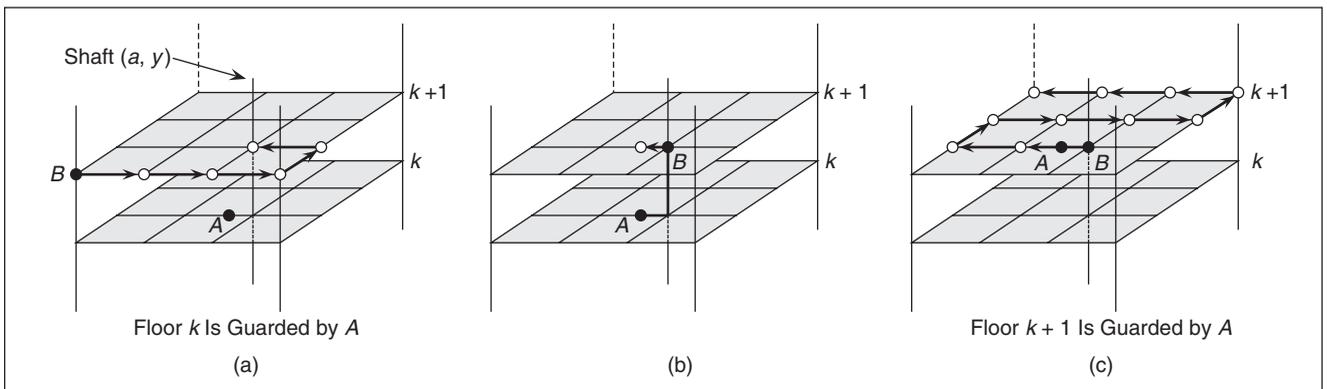


**Figure 8.** Traversing path  $p_{XY}^k$ . Point  $p$  succeeds point  $q$  on  $p_{XY}^k$ .

Without loss of generality, assume that at time  $t$ , robot  $A$  is guarding XY-layer  $k$  at point  $p_H = (x, y, k) \in H_{XY}^k$ , and  $Z$  lies above the layer. Let  $(a, y, k)$  be the vertex at distance  $\frac{\epsilon}{2}$  of  $p_H$ , where  $a$  is either  $x + \frac{\epsilon}{2}$  or  $x - \frac{\epsilon}{2}$ . From now on, for convenience, we shall refer to an XY-layer as a *floor*. Of course,  $Z$  does not know where  $p_H$  is. Our next objective is to advance the guarded floor from  $k$  to  $k + 1$  by either moving  $A$  to floor  $k + 1$  without revealing to  $Z$  where  $A$  is, or hiding  $B$  in floor  $k + 1$ . We use the following procedure to achieve this, where  $p_B$  denotes the current position of  $B$  (Figure 9).

### Procedure Traverse

- 1)  $B$  goes to vertex  $p_B = (0, 0, k + 1)$  and stays there for 1 time unit. If  $p_B = (a, y, k + 1)$ , then  $A$ , currently at  $p_H = (x, y, k)$ , which is at distance  $\frac{\epsilon}{2}$  of  $(a, y, k)$ , moves to  $(x, y, k + 1)$  through shaft  $(a, y)$ . (In this case,  $p_H$  is updated by  $p_H := (x, y, k + 1)$ .) For convenience, at this moment, we reset time to  $t = 0$ .
- 2) For  $l = 0, 1, \dots, n^2 - 1$ :
  - ◆ In the time interval  $[2l, 2l + 1]$ , robot  $B$ , currently at vertex  $p_B$ , moves to vertex  $\text{succ}(p_B)$  along path  $p_{XY}^{k+1}$  at speed 1. (Thus,  $p_B$  is updated by  $p_B := \text{succ}(p_B)$ .)
  - ◆ In the time interval  $[2l + 1, 2l + 2]$ ,  $B$  stays at vertex  $p_B$ . Meanwhile, if  $p_B = (a, y, k + 1)$ , then  $A$ , currently at  $p_H = (x, y, k)$ , which is at distance  $\frac{\epsilon}{2}$  of  $(a, y, k)$ , moves to  $(x, y, k + 1)$  through shaft  $(a, y)$  at speed  $s$  (which takes exactly 1 time unit). (In this case,  $p_H$  is updated by  $p_H = (x, y, k + 1)$ .)



**Figure 9.** Procedure Traverse. (a)  $A$  is hiding at  $p \in H_{XY}^k$  at distance  $\frac{\epsilon}{2}$  of  $(a, y, k)$  and remains stationary until  $B$  reaches  $(a, y, k + 1)$ . (b)  $A$  moves along shaft  $(a, y)$  to floor  $k + 1$  while  $B$  stays at  $(a, y, k + 1)$  for 1 time unit. (c)  $B$  continues his move toward the endpoint of path  $p_{XY}^{k+1}$ .

### Termination Conditions

As soon as one of the following holds, the above iteration is terminated, and the specified action is taken:

- a) If  $A$  sees  $Z$  within distance 1, then  $A$  starts chasing  $Z$ .
- b) Otherwise, if  $B$  sees  $Z$  within distance 1, then  $B$  starts chasing  $Z$ .
- c) Otherwise, if  $B$  sees  $Z$  above in some shaft, then  $B$  hides in floor  $k + 1$  using Hide at some point in  $H_{XY}^{k+1}$ .
- d) Otherwise, if  $B$ , currently at  $p_B$ , sees  $Z$  at some point  $p_Z$  succeeding  $p_B$  on  $p_{XY}^{k+1}$ , then  $B$  starts moving toward  $Z$  and forces  $Z$  to disappear from  $B$ 's line of sight at some vertex  $(i', j', k + 1)$ . At the moment  $Z$  this happens, if  $A$  is still located at point  $p_H$  in floor  $k$ , then  $B$  tells  $A$  to move to floor  $k + 1$  via the closest shaft at speed  $s$ . When  $A$  reaches floor  $k + 1$  within 1 time unit,  $A$  is within distance 1 of  $Z$  with probability  $\frac{1}{n^2}$  and starts chasing  $Z$ . Traverse ends in failure if  $Z$  is not within distance 1 of  $A$  when  $A$  reaches floor  $k + 1$ , and chasing does not start.

In Traverse, we attempt to advance the guarded floor from  $k$  to  $k + 1$  by moving  $A$  from its hiding location  $p_H$  in floor  $k$  to the point in floor  $k + 1$  right above it. Since we do not want  $Z$  to find out where  $A$  is hiding by seeing  $A$  while he is moving to floor  $k + 1$ , we first place  $B$  on the shaft  $A$  uses so that if  $Z$  appears on that shaft, then  $B$  can hide in floor  $k + 1$  using Hide. Since  $Z$  also gains some knowledge about  $A$ 's hiding position by *not* seeing  $A$  in a shaft when  $A$  would be using it if he was hiding near it, we let  $B$  visit every vertex on floor  $k + 1$  along  $p_{XY}^{k+1}$  and stay there for 1 time unit, so that in case  $Z$  appears on the corresponding shaft (to see whether or not  $A$  appears there),  $B$  can immediately hide in floor  $k + 1$ .  $Z$  may also gain some knowledge about  $A$ 's position by seeing a potential hiding location  $p$  for  $A$  on floor  $k + 1$  near a vertex that  $B$  has already visited and finding out that  $A$  is or is not there. Recall that the hiding locations cannot be seen along any north-south corridor. Thus,  $Z$  has to enter an east-west corridor containing  $p$  and risk being seen by  $B$  ( $B$  will then force  $Z$  to leave the corridor at a vertex and ask  $A$  to come up to floor  $k + 1$ , and when  $A$  reaches floor  $k + 1$ , he will be within distance 1 of  $Z$  with a probability of  $\frac{1}{n^2}$ ) or encounter  $A$  within distance 1 with a probability of at least  $\frac{1}{n^2}$ . The only remaining possibility for  $Z$  to gain knowledge about  $A$ 's position

is to see a potential hiding location  $p$  for  $A$  on floor  $k$  near a shaft that  $B$  has not visited (in floor  $k + 1$ ). Again, in this case,  $Z$  encounters  $A$  within distance 1 with a probability of at least  $\frac{1}{n^2}$  when he enters the east-west corridor containing  $p$ .

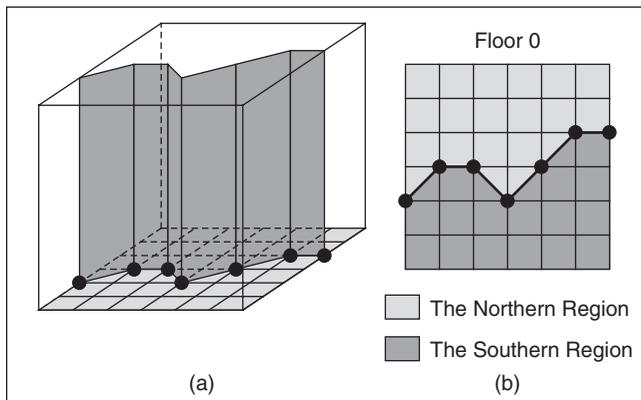
**Lemma 10:** Assume that *Traverse* is executed with  $A$  hiding in floor  $k$  and  $Z$  lying above floor  $k$ . Then, within  $O(n^2)$  time, either chasing starts with probability at least  $\frac{1}{n^2}$ , or one of the robots hides in floor  $k + 1$  with  $Z$  lying above that floor.

**Proof:** By assumption,  $Z$  has no knowledge about  $A$ 's position when *Traverse* is started. As explained above, chasing starts with a probability of  $\frac{1}{n^2}$  if  $Z$  attempts to gain any knowledge about  $A$ 's hiding location ( $p_H \in H_{XY}^k$  or the point in floor  $k + 1$  right above it). Thus, in the following, assume that  $Z$  gains no knowledge about  $A$ 's hiding location during the execution of *Traverse*. Since  $B$  visits every vertex in floor  $k + 1$  (unless chasing starts early), in order for  $Z$  to be on or below floor  $k + 1$  at the end of *Traverse*,  $Z$  must either 1) reach floor  $k$  through edge  $\{(i, j, k), (i, j, k + 1)\}$  before  $B$  reaches vertex  $(i, j, k + 1)$  or 2) enter edge  $\{(i, j, k), (i, j, k + 1)\}$  through vertex  $(i, j, k + 1)$  after  $B$  leaves  $(i, j, k + 1)$ , for some  $0 \leq i, j \leq n - 1$ . Since  $Z$  has no knowledge about  $A$ 's hiding location, at the moment case 1) occurs for the first time,  $A$  is within distance 1 of  $(i, j, k)$  and starts chasing  $Z$  (condition a) with a probability of at least  $\frac{1}{n^2}$ . For the same reason, at the moment case 2) occurs for the first time,  $A$  is within distance 1 of  $(i, j, k + 1)$  and starts chasing  $Z$  (condition a) with a probability of at least  $\frac{1}{n^2}$ . This completes the proof. ■

Clearly, it may happen that  $Z$  successfully escapes to the area below floor  $k$  during the execution of *Traverse* without  $A$  and  $B$  knowing it.  $A$  and  $B$  will not notice this until they reach floor  $n - 1$ . Then, of course, our capturing strategy ends in failure.

**Theorem 11:** In  $G_{n \times n \times n}$ , with probability at least  $\frac{1}{n^2}$ , two robots with a maximum speed of  $s = 1 + \varepsilon$  can start chasing an arbitrary evader  $Z$  within  $O(n^3)$  time. The expected time to capture  $Z$  by repeating this process is  $O(n^5)$ .

**Proof:** Detecting  $Z$  takes  $O(n^2)$  expected time by Theorem 3. Once  $Z$  is found, at most  $n$  executions of procedure



**Figure 10.** A barrier and the relevant southern and northern regions.

*Traverse* in total  $O(n^3)$  time allow the robots to guard floor  $n - 1$  with the area above the guarded floor being empty. Thus, before this happens,  $Z$  must attempt to escape to the area below a guarded floor. When that happens, by Lemma 10 chasing starts with a probability of at least  $\frac{1}{n^2}$ . A capture then follows within  $O(1)$  time by Lemma 1. ■

## Deterministic Strategy Using $\lfloor (4n - 2)/3 \rfloor + 6$ Robots

All the aforementioned strategies use randomized techniques, which result in an expected time for successfully capturing an evader. We conclude our study with a deterministic strategy that uses  $\lfloor (4n - 2)/3 \rfloor + 6$  robots. This seems to be quite a lot, but we conjecture that at least  $n + 1$  robots are required even to deterministically detect the evader in  $G_{n \times n \times n}$ . (In the 2-D case, two robots are sufficient to detect the evader [13].) Similar to the randomized algorithm described in the “Capturing by Forming a Trap” section, our algorithm is based on forming a trap. Thus all we need is to show how to form a trap in a deterministic manner. The idea is to sweep floor 0 with the robots and maintain a “trap barrier” that allows the robots to form a trap as soon as  $Z$  crosses the barrier. We shall discuss this in the following sections.

### A Trap Barrier

We say that a set of robots, located at vertices of floor 0, forms an *east-west barrier* if 1) for every north-south corridor, exactly one of its vertices is occupied by a robot (or a group of them), and, 2) for every  $i = 0, \dots, n - 2$ , the distance between a robot in the north-south corridor  $(i, 0)$  and a robot in the north-south corridor  $(i + 1, 0)$  is at most 2 (see Figure 10).

A barrier divides floor 0 and the grid above it into two (not necessarily connected) regions called the *northern* and the *southern* regions [see Figure 10(b) for an illustration]. We say that the southern (or northern) region is *clean* if the robots know that there is no evader in it; otherwise, it is *contaminated*.

The idea of our strategy is first to build a barrier with the certainty that the southern region is clear, and then to advance it to the north, thus increasing the clean southern region. If at any time moment an evader wants to cross the barrier and enter the clean region, the robots are able to form a trap.

### Skewed 211-Barrier

Define a *skewed 211-barrier* as a barrier that is ordered from west to east such that 1) the numbers of robots in the relevant (unique) vertices, from west to east, are 3, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1,  $\dots$ , 3, and 2) for any single robot placed at vertex  $(i, j, 0)$ , there are at least two robots at one of the following vertices:  $(i - 1, j - 1, 0)$ ,  $(i - 1, j + 1, 0)$ ,  $(i + 1, j - 1, 0)$ ,  $(i + 1, j + 1, 0)$ . The crucial property of a skewed 211-barrier is the following lemma.

**Lemma 12:** Suppose that evader  $Z$  crosses a skewed 211-barrier above vertex  $(i, j, 0)$ . Then, the robot(s) located at  $(i, j, 0)$ , with help from the robots in its neighborhood, is (are) able to form a trap with  $Z$  inside it within one unit time.

**Proof:** The claim follows by a simple enumeration of possible configurations (details are omitted). ■

## Forming and Advancing a Trap Barrier

The first step of our deterministic capturing strategy is to place  $\lfloor (4n-2)/3 \rfloor + 4$  robots in floor 0 in the following manner, which clearly results in a skewed 211-barrier (we use the remaining two robots later):

- ◆ three robots at vertex  $(0, 0, 0)$
- ◆ one robot at vertex  $(1, 1, 0)$  and at vertex  $(2, 0, 0)$
- ◆ two robots at vertex  $(3, 1, 0)$
- ◆ one robot at vertex  $(4, 0, 0)$  and at vertex  $(5, 1, 0)$
- ⋮
- ◆ three robots at vertex  $(n-1, 0, 0)$  if  $n$  is even, or at vertex  $(n-1, 1, 0)$  otherwise.

Of course, at this moment, both the southern and northern regions are contaminated. However, by repeating the following procedure  $\text{Clear}(i) \lfloor \frac{n}{2} \rfloor$  times for  $i = 1, 2, \dots, \lfloor \frac{n}{2} \rfloor$ , within  $O(n^2)$  time, either the southern region becomes clear or a trap is formed with  $Z$  in it.

### Procedure $\text{Clear}(i)$

Another robot  $R$  goes to vertex  $(2i-1, 0, 0)$  and then moves up along shaft  $(2i-1, 0)$  at a speed of 1 to vertex  $(2i-1, 0, n-1)$  (Figure 11).

### Termination Conditions

As soon as one of the following holds, the above procedure is terminated, and the specified action is taken:

- a) If one of the robots sees  $Z$  within distance 1, then he starts chasing  $Z$ .
- b) If  $Z$  crosses the barrier, a trap is formed.

It is easy to see that when procedure  $\text{Clear}(i)$  is executed for  $i = 1, 2, \dots, \lfloor \frac{n}{2} \rfloor$ , in order to avoid being chased by  $R$  (condition a),  $Z$  (if it is in the southern region) is forced to cross the barrier above one of the vertices in the barrier, which immediately results in the formation of a trap by Lemma 12 (condition b). As  $\text{Clear}(i)$  takes  $O(n)$  time, the overall time to confirm the clearance of the southern region, to form a trap, or to start chasing otherwise, is  $O(n^2)$ .

So suppose that up to now neither a trap has been formed nor chasing has started. All we need is to show how to advance our skewed 211-barrier to the north and increase the size of the clear southern region.

Our advancing strategy is based on a local flipping of the barrier. Consider the initial skewed 211-barrier. Let us use two robots  $R_1$  and  $R_2$  (currently not forming the barrier) and, for every  $i = 0, 1, 2, \dots, \lfloor (n+1)/2 \rfloor$ , execute the following procedure  $\text{Flip}(i)$  (Figure 12).

### Procedure $\text{Flip}(i)$

- 1)  $R_1$  goes to vertex  $(2i, 1, 0)$ , while  $R_2$  goes to  $(2i, 2, 0)$ .
- 2)  $R_2$  remains at  $(2i, 1, 0)$ , and  $R_1$  moves up along shaft  $(2i, 1)$  at a speed of 1 to vertex  $(2i, 1, n-1)$ .
- 3) One of the robots occupying  $(2i, 0, 0)$ , say  $R$ , remains stationary, and all other robots (if any) occupying  $(2i, 0, 0)$  go straight to  $(2i, 2, 0)$  in 2 time units. Then, we exchange the roles or names of robots  $R$  and  $R_2$ .

### Termination Conditions

As soon as one of the following holds, the above procedure is terminated and the specified action is taken:

- a) If one of the robots sees  $Z$  within distance 1, then he starts chasing  $Z$ .
- b) If  $Z$  crosses the barrier, a trap is formed.
- c) If  $Z$  appears above  $R_2$ , a trap is formed.

**Lemma 13:** Assume that  $\text{Flip}(i)$  is executed with  $Z$  hiding in the northern region, for  $i = 0, 1, \dots, \lfloor (n+1)/2 \rfloor$ . Then, within  $O(n^2)$  time, either chasing starts, a trap is formed with  $Z$  in it, or a new skewed 211-barrier is maintained with a larger clear southern region.

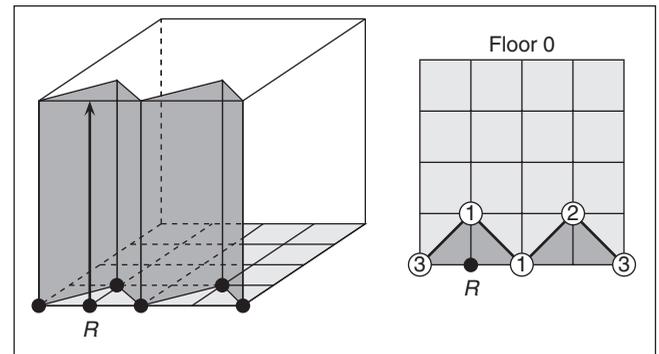


Figure 11. An illustration of procedure  $\text{Clear}$  for  $i = 1$ .

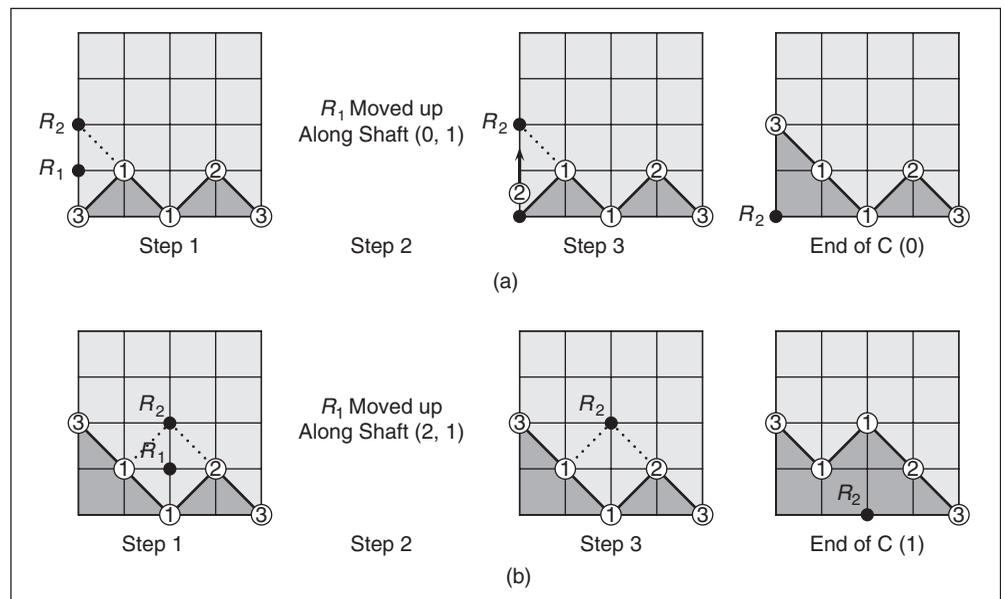


Figure 12. An illustration of procedure  $\text{Flip}$ : (a)  $i = 0$ , (b)  $i = 1$ .

**Proof:** Consider the execution of `Flip` for  $i = 0$  [see Figure 12(a)]. Clearly, if during Step 1 or 2 evader  $Z$  tries to cross the current barrier, a trap is formed (condition b). Next, observe that robots at vertices  $(0, 0, 0)$ ,  $(0, 2, 0)$ , and  $(1, 1, 0)$  form a local trap (possibly empty), and thus, by moving robot  $R_1$  along shaft  $(0, 1)$  during Step 2, evader  $Z$  is either trapped or chasing starts (conditions a–c). Consequently, if none of the above happens, at the end of Step 2,  $Z$  must lie in the northern region, but not in the local trap. Finally, notice that during Step 3, the local trap is maintained, and any attempt of  $Z$  to cross the barrier to the east to the local trap results in forming a new trap, now with  $Z$  in it (it follows because this section of the 211-barrier has not changed). It is also easy to see that the new barrier is a skewed 211-barrier. Consequently, the assertion of the lemma follows for  $i = 0$ .

Consider now the case  $i = 1$  [see Figure 12(b)]. Again, if during Step 1 or 2, evader  $Z$  tries to cross the current barrier, a trap is formed (condition b). Next, observe that robots at vertices  $(1, 1, 0)$ ,  $(2, 2, 0)$ ,  $(3, 1, 0)$ , and  $(2, 0, 0)$  form a local trap (possibly empty), and thus, by moving robot  $R_1$  along shaft  $(2, 1)$  during Step 2, evader  $Z$  is either trapped or chasing starts (conditions a–c). Therefore again, if none of the above happens, at the end of Step 2,  $Z$  must lie in the northern region, but not in the local trap. Step 3 is not executed, as there is only one robot at  $(2, 0, 0)$ , i.e., we only rename robots. Finally, it is easy to see that the new barrier is the skewed 211-barrier as well, and, thus, the assertion of the lemma follows for  $i = 1$ .

Clearly, by similar arguments and a simple induction argument, one can prove the assertion of the lemma for every  $i = 0, 1, \dots, \lfloor (n+1)/2 \rfloor$ .

Observe that when procedure `Flip`( $i$ ) is finished for all  $i$ , the new barrier is a “flipped” copy of the previous barrier along the east-west corridor  $(1, 0)$ . It is easy to see that the flipping operation can be applied successively along the east-west corridors  $(2, 0)$ ,  $(3, 0)$ ,  $\dots$ . Consequently, the clean southern region becomes larger and larger, and hence, eventually either chasing starts or evader  $Z$  is trapped. Clearly, this must happen within  $O(n^3)$  time, as this is the time for completing the entire flipping process. And, if chasing has started, then by Lemma 1, a capture occurs within  $O(n^3)$  time. Otherwise, by an argument similar to those in the “Capturing by Forming a Trap” section, once  $Z$  is trapped, the linear number of robots allows us to start chasing within  $O(n)$  time, which then again results in a capture within  $O(n^3)$  time. Consequently, we obtain the following theorem. ■

**Theorem 14:** *In  $G_{n \times n \times n}$ , using a deterministic algorithm,  $\lfloor (4n - 2)/3 \rfloor + 6$  robots with a maximum speed of  $s = 1$  can capture in a building an arbitrary evader within  $O(n^3)$  time.*

## Conclusions

We have presented a number of randomized and deterministic algorithms for capturing an evader in a 3-D grid  $G_{n \times n \times n}$ . As noted earlier, the algorithms for the 3-D case turned out to be more complicated than those for the 2-D case.

For future research, we need to establish a conjectured lower bound of  $n + 1$  on the number of robots having a maximum speed of 1 necessary to detect an evader. Another interesting problem is to detect an evader deterministically using  $O(1)$

robots having a much larger speed  $s$ , such as  $s = \Theta(n)$  for the case of a single robot in a 2-D grid  $G_{n \times n}$  [4], [9].

## Acknowledgments

Paweł Żyliński was supported by the Visby Programme Scholarship 01224/2007 (the Swedish Institute).

## Keywords

Pursuit game, pursuit evasion, randomized algorithms.

## References

- [1] M. Adler, H. Räcke, N. Sivasadan, C. Sohler, and B. Vöcking, “Randomized pursuit-evasion in graphs,” *Combinat. Prob., Comput.*, vol. 12, no. 3, pp. 225–244, 2003.
- [2] M. Aigner and M. Fromme, “A game of cops and robbers,” *Discrete Appl. Math.*, vol. 8, no. 1, pp. 1–12, 1984.
- [3] T. Andreae, “Note on a pursuit game played on graphs,” *Discrete Appl. Math.*, vol. 9, no. 2, pp. 111–115, 1984.
- [4] R. W. Dawes, “Some pursuit-evasion problems on grids,” *Inf. Process. Lett.*, vol. 43, no. 5, pp. 241–247, 1992.
- [5] A. Dumitrescu, H. Kok, I. Suzuki, and P. Żyliński, “Vision-based pursuit-evasion in a grid,” to appear in *Lecture Notes in Computer Science*. Berlin: Springer-Verlag, to be published.
- [6] V. Isler, S. Kannan, and S. Khanna, “Randomized pursuit-evasion with local visibility,” *SIAM J. Discrete Math.*, vol. 20, no. 1, pp. 26–41, 2006.
- [7] L. M. Kirousis and C. H. Papadimitriou, “Searching and pebbling,” *Theoret. Comp. Sci.*, vol. 47, no. 2, pp. 205–218, 1986.
- [8] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C.H. Papadimitriou, “The complexity of searching a graph,” *J. ACM*, vol. 35, no. 1, pp. 18–44, 1996.
- [9] S. W. Neufeld, “A pursuit-evasion problem on a grid,” *Inf. Process. Lett.*, vol. 58, no. 1, pp. 5–9, 1996.
- [10] S. W. Neufeld and R. Nowakowski, “A game of cops and robbers played on product of graphs,” *Discrete Math.*, vol. 186, no. 1–3, pp. 253–268, 1998.
- [11] T. D. Parsons, “Pursuit-evasion in a graph,” in *Lecture Notes in Mathematics*, vol. 642, Berlin: Springer-Verlag, 1976, pp. 426–441.
- [12] K. Sugihara and I. Suzuki, “On a pursuit-evasion problem related to motion coordination of mobile robots,” in *Proc. 21st Hawaii Int. Conf. System Sciences*, Kailua-Kona, Hawaii, 1988, pp. 218–226.
- [13] K. Sugihara and I. Suzuki, “Optimal algorithms for a pursuit-evasion problem in grids,” *SIAM J. Discrete Math.*, vol. 2, no. 1, pp. 126–143, 1989.

**Ichiro Suzuki** received his D.E. degree in information and computer sciences from Osaka University, Japan, in 1983. He is currently a professor of computer science at the University of Wisconsin, Milwaukee. He has held visiting positions at Osaka University and Kyushu University. His research interests include distributed/concurrent systems, computational geometry, and computational robotics. He is a member of the Association for Computing Machinery.

**Paweł Żyliński** received his M.Sc. and Ph.D. degrees in mathematics from the University of Gdańsk in 2000 and 2004, respectively. He is currently on sabbatical leave from the Institute of Computer Science, University of Gdańsk, and is visiting the Lund University under the postdoctoral Visby program scholarship.

**Address for Correspondence:** Paweł Żyliński, Department of Computer Science, University of Gdańsk, Gdańsk, 80-952, Poland. E-mail: pz@inf.univ.gda.pl.

# Ways to Tell Robots Where to Go

## Directing Autonomous Robots Using Topological Instructions

**W**e often need to navigate directly to goals in unfamiliar or unknown places. Language allows us to exploit the knowledge of others by transferring all the necessary information from one person to another. What if robots could do the same? Most modern robots must explore and map before goal-directed navigation can occur, because they cannot navigate using the maps in peoples' heads. This article presents our attempts to direct an autonomous robot using efficient and universal topological instructions, which can be incrementally interpreted by a moving robot that does not have its own map initially. Many real-world experiments are included, featuring autonomous exploration and mapping. Surprisingly, we conclude and show that for this type of navigation, abilities in object recognition are more important than better mapping.

### Human-Robot Collaboration on Navigation

#### Map First, Achieve Goals Later

A robot's ability to sense and map its environment has always been the most limiting factor in autonomous navigation. Research has rightly focused on improved numerical methods for the integration of data from many noisy sensors [1]. The output has been increasingly accurate [2], detailed [3]–[5], and large-scale [6] geometric models of the environment. In other words, better maps have been generated.

Digital Object Identifier 10.1109/MRA.2008.921538

Nevertheless, we usually have a very good idea of where we want our robots to go and how to get there. If only this could be explained to robots, the mapping stage could be bypassed. We could use the maps in our heads to plan a route for the robot to execute. Previously, human-robot collaboration on navigation had been limited to three scenarios.

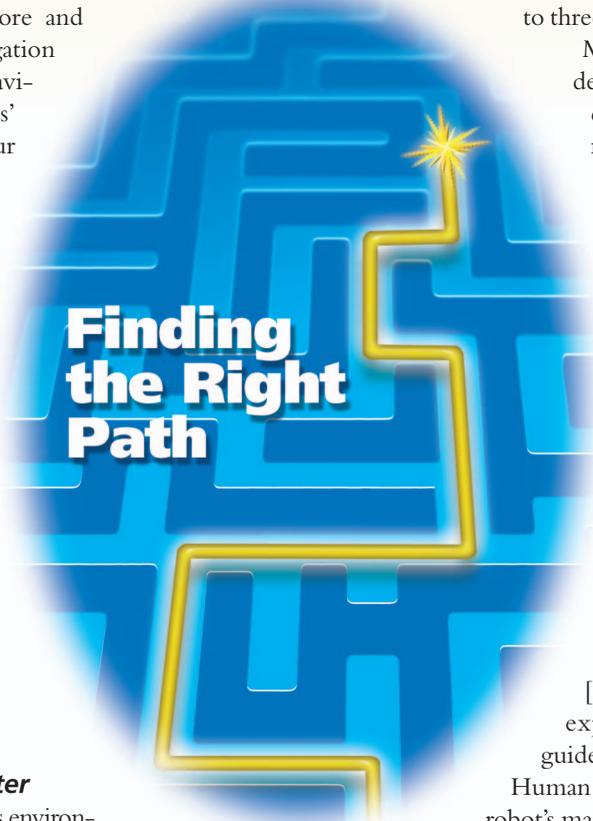
Most often, the robot's programmer defines an exploration policy that determines how the robot should make decisions during exploration. For example, "follow the left wall" is a simple exploration policy. Policies often optimize the accuracy [7], utility [8], or coverage [9] of the robot's map. This is great if the objective is mapping, but if the objective is navigation to a goal, comprehensive exploration and mapping is wasteful.

Other human-robot cooperative scenarios require humans to interpret the robot's internal map, usually to identify interesting places and order the robot to navigate to these goals. Annotation can occur offline post-mapping [10] or online during handheld exploration, where the operator guides the robot as it explores [11].

Human annotation cannot occur before the robot's map exists, because the human is doing all the interpretation work. If the robot could interpret humans' internal maps, it could be instructed on how to reach a goal before it does any mapping.

#### Directed Navigation in Unknown Environments

We envisage a fourth scenario in which humans plan a route using the information already available in their memory by imagining or looking at the environment and deciding which route the



©DIGITAL VISION

BY DAVID RAWLINSON AND RAY JARVIS

robot should follow. The robot begins the navigation task near the human director. The director gives the robot a series of instructions before it begins to navigate. Once the robot has begun its mission, it is fully autonomous. No further intervention is allowed. Except for the instructions, the robot is given no information about its environment. We call this directed navigation.

This scenario is analogous to many real-world situations. For example, being lost in a strange city you could systematically explore until you find your hotel or you could ask someone for directions. Although some helpful people will show you the way, usually you have to follow such directions independently.

### Navigational Directions in Natural Language

The problem of transforming human knowledge into useful navigational information (and vice versa) spans several active research fields beyond robotics, including natural language processing and human-computer interaction. Levit and Roy [12] have tried to develop an autonomous interpretation of human navigational language, ignoring many robotic problems such as symbol grounding and obstacle avoidance. Park and Kender [13] did consider some of these aspects, in an attempt to autonomously construct and interpret topological directions, but they limited their research to artificial environments with only sparse monolithic features.

An interesting article by Setalaphruk et al. [14] begins with the observation that although humans' topological knowledge is good, their distance estimation is poor. They developed a simulated robot, which uses a geometrically inaccurate but topologically correct sketch floor map to navigate. The map is provided in advance by a human.

The interpretation of human topological directions has also impacted the Humanoid Research Project at the National Institute of Advanced Industrial Science and Technology (AIST). Their robot, HRP-2, is intended to function as a domestic assistant [15]. To fulfill this role, HRP-2 must interpret human instructions, not necessarily navigational but of a topological type similar to those used in navigation; i.e., relationships between objects rather than geometric descriptions. The researchers at AIST call this type of instruction high-level teleoperation.

Finally, Wang et al. [11] created a variably autonomous teleoperated robot that explores under human guidance. Human input can be high level (topological landmark goals in a complete map given to the robot) or low level (motion commands). Their robot can choose to ignore low-level commands that it perceives to be dangerous.

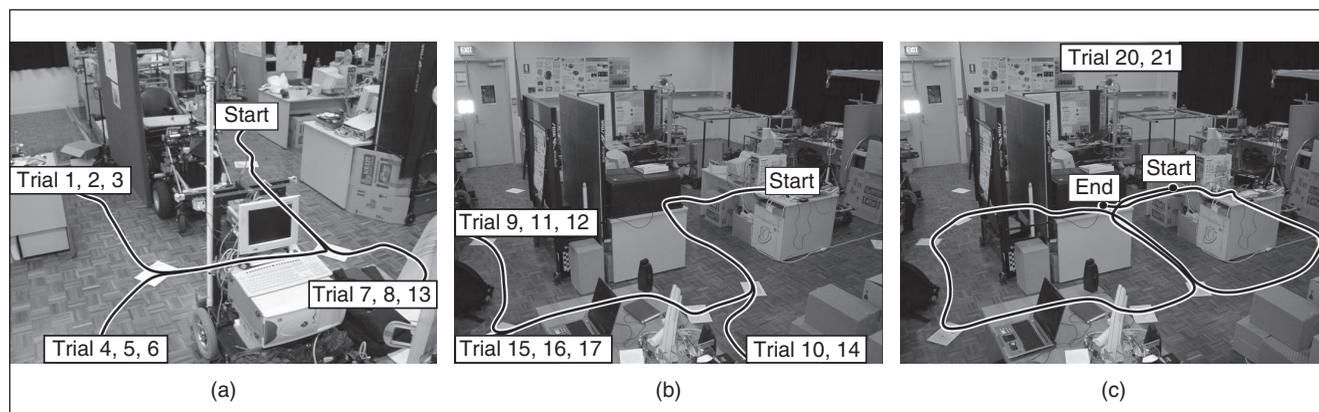
These works illustrate the various ways people think with regard to navigational problems and the sort of information a person could easily provide to a robot navigator. We tend not to describe a journey in terms of vectors and do not have the information available to do so accurately. Instead, we tend to describe a journey in terms of landmarks and the actions triggered by those landmarks, such as "turn left at the end of the road. . . ."

### Challenges in This Scenario

So, the robot must accept topological directions and not geometric ones for the convenience of the human director. Instead of "move  $N$  meters at bearing  $\theta$  radians," the robot's instructions will be of the form "do <action> [at or until] <landmark>" (see Figure 1). The robot will not be provided with a conventional map in advance. It must construct a map while executing the instructions. This section describes the problems that must be overcome for this type of navigation to succeed. Many of the traditional challenges to autonomous navigation are exaggerated by these conditions.

To date, the most closely related works are the robot delivery tasks of the 1994–1995 International Joint Conferences on Artificial Intelligence (IJCAI) robot competitions [16], where robots navigated in an artificial office environment using topological directions such as "exit room and turn first right." The most successful team in 1995 was CAIR-2 [17], which relied on distinctive visual markers over doorways to associate topological concepts with their real-world equivalents. Although the environment was simplified, a number of challenges were tackled including human-robot interaction and robot detection of instruction error or impossibility.

Surprisingly, since 1995 similar work on topological direction has not been published. This may reflect a change in emphases in robotics, which are toward unpredictable, implicit



**Figure 1.** Routes taken by the robot in the laboratory as directed by the author. Starting from the author's desk, the robot was directed to all other students' desks by a variety of routes. The robot was given topological instructions but no map. In two trials, (c) the robot performed a figure-8 route back to the author's desk. The following is a typical instruction sequence: "go <this way> until junction. At next junction, turn left. Then, take second exit clockwise at next junction. . . ."

topological partitioning [16], [18], [19], improved geometric mapping via simultaneous localization and mapping (SLAM) [2], and new problems, which are, consequently, solvable. The IJCAI task changed in 1996, omitting autonomous topologically directed navigation before robot exploration.

### Analogy to Topological Maps

A topological map is usually represented as a graph. Vertices represent landmarks or landmark locations. Typically, an edge between two vertices indicates that it is possible to travel directly between them. So, a series of topological navigational instructions are similar to an incomplete topological map. The big difference is that while most real-world topological mapping systems are geometric or topological hybrids, in this scenario, topological instructions are provided without the underlying geometric data.

### Incremental Topological Interpretation

To execute a series of topological instructions, a robot must simultaneously avoid obstacles, plan motion toward the next landmark (in accordance with the current instruction), map its surroundings, detect topological landmarks, and localize itself topologically within the instructions and internal mappings. It must produce both geometric and topological models of its environment, just in time, by detecting and avoiding obstacles before it hits them and modeling topological landmarks accurately before they are reached, using only the information gathered by sensors during the approach. Landmarks relevant to instructions must be identified and others ignored (Figure 2).

While following the instructions, landmarks must not be missed. Movement between any pair of landmarks is conducted without any foreknowledge of the environment between those landmarks, and, in our experiments, often without knowing in what direction or how far apart the landmarks are. Most existing autonomous mapping systems generate topology offline, postexploration, or are inaccurate until mapping is nearly complete [20]–[22] because of a lack of geometric data.

Finally, in the proposed scenario, instructions will be in a form that humans could verbally transmit to a robot, which limits the amount of information that can be conveyed about landmarks. Only abstract landmarks can be described. The robot must translate these abstract concepts into concrete entities in the world, which is a classic example of the symbol-grounding problem.

### Choosing a Suitable Topology

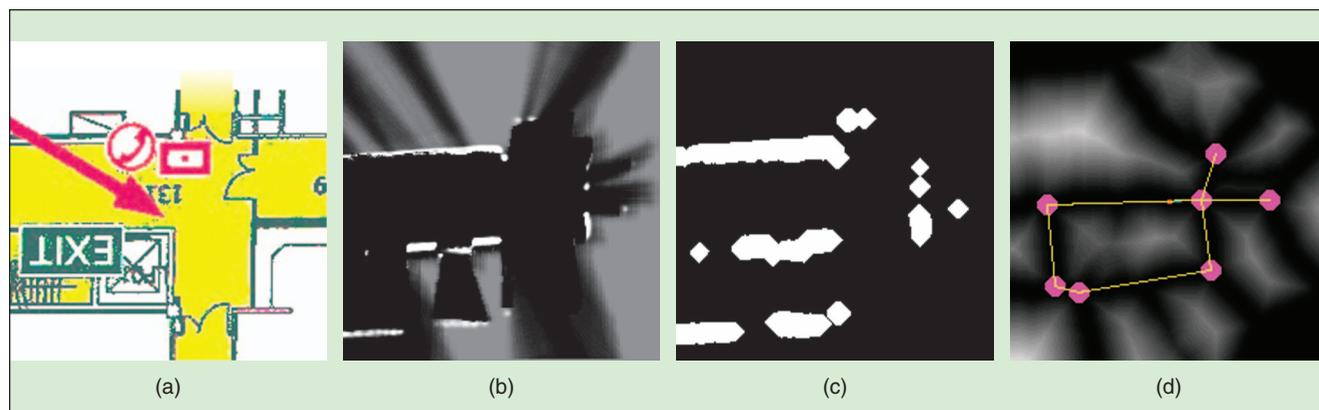
The topology in which the instructions are based is crucial to both the viability of autonomous, incremental interpretation and the generation of correct instructions by a person. The topology is defined by the choice and range of landmarks that the robot will recognize (e.g., doors and junctions) and the relationships that will be permitted between them.

Minimizing the number of different landmark types reduces the burden on the robot, as it must reliably perceive them all. However, the set of landmarks chosen must occur throughout the environment; e.g., instructions based on traffic lights would be of no use in the jungle.

There are many tradeoffs to be made in choosing landmarks. They must be frequent enough so that whenever a behavior change is required, there is a landmark present to trigger it. Landmarks should not be more frequent than necessary, as this would complicate the directions and increase the chance of human or robot error. Therefore, the simplest structural features, such as corners and edges often used in hybrid mappings [6], [10], are unsuitable as topological landmarks. We would struggle to correctly enumerate the huge number of such features in any typical scene. Certainly, we would not be able to describe these geometric relationships with useful accuracy.

### Navigating the Generalized Voronoi Diagram

Mathematically speaking, the generalized Voronoi diagram (GVD) is the locus of points in a plane that are equidistant to two or more feature points. More intuitively, the GVD can be imagined as a particular skeleton of the background space that



**Figure 2.** An example of an egocentric modeling of obstacles and topology from our experiments. The robot is always at the center of these rolling windows. Here, it approaches a four-way junction. Despite having only very limited obstacle information, the robot's estimate of local topology is very accurate. In fact, the topology only needs to be accurate at the center of the window, where the robot makes decisions. (a) Floor plan of the area (not given to robot). (b) Occupancy probability; white pixels are most likely to be occupied. (c) Thresholded, dilated obstacles (white) and free space (black). (d) Topological graph overlaid on distance from edges (paths) in a gray scale.

is defined by these features. When applied to navigation, GVD features often represent obstacles, making the GVD a skeleton of the free space between obstacles. The skeleton may be more simply represented as a generalized Voronoi graph (GVG) in which every endpoint and meet-point of the GVD's skeletal lines becomes a vertex. Choset and Nagatani published a series of detailed works describing the application of the GVD and GVG to autonomous navigation [21], [23].

### Properties of the GVD

The GVD has many useful properties, but further processing is required to extract useful topological features and avoid some problems, as illustrated in Figure 3(a) and (b). In these graphics, black dots are obstacles, and a pair of concentric blue circles denote the robot and the limits of its sensor range. The true GVD is very complex, including both black and red lines.

The robot must model the GVD using only the obstacles it can detect with its sensors. This is particularly difficult when the robot moves from a corridor environment [Figure 3(a)] to an open space [Figure 3(b)]. The robot cannot detect obstacles on both sides of this place due to its width. Instead, it chooses to follow the left wall. In Figure 3(b), a single new obstacle has radically altered the graph and created a new junction. To avoid erratic behaviour in chambers like this that are slightly too wide for centreline-following, the wall-following distance can be set to overlay the centreline graph when obstacles appear at the limit of sensor range. This makes the behavior constant despite changes in the perceived topology.

The navigationally relevant meet points of the GVD skeleton are called junctions and are used as landmarks for topological instructions. Few meet points are useful. Many represent narrow gaps between obstacles (often merely sensing artefacts) that the robot cannot pass through. The red lines in Figure 3 are irrelevant parts of the GVD. Junctions, however, conveniently occur only when navigational decisions need to be made (i.e., at ideal frequency) and are universal (i.e., junctions occur in any configuration of obstacles). This is a neat inversion of the problem of choosing landmarks, mapping the impact of obstacles on the robot's choices, rather than the myriad shapes of the obstacles themselves. The few black lines in Figure 3(a) and (b) represent this pruned graph of junctions.

Being gross features defined by many data points, junctions are also reliably detected by the robot's sensors. Most junctions are also local features that could be modeled on approach, making it possible to follow a topological map of junctions without completing geometric mapping first. The GVD is also an optimally safe path for the robot to follow, maximizing distances from obstacles.

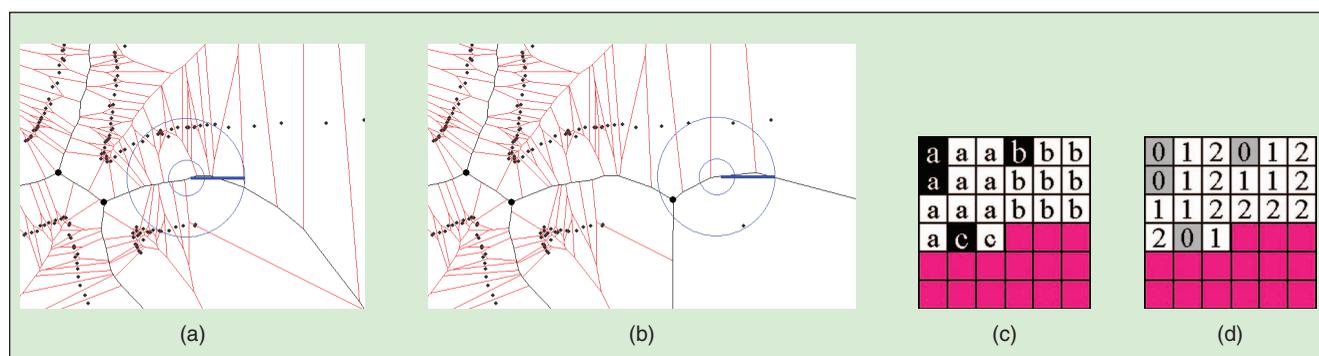
### Related Work

In navigation studies, the GVD is most commonly used in hybrid maps that contain topological and geometric data in parallel models. Typically, the GVD is computed from global occupancy data [23]–[25], and the key contribution of the GVD is efficient topological planning (the detailed geometric model provides robust localization). Choset and Nagatani were influential pioneers of this approach [21], [23].

The GVD may be computed using geometric methods [26], but such accuracy and cost is usually excessive. The alternatives are discretizing the obstacles' perimeters and discretizing space (i.e., a raster, image, or grid system). Discretizing space makes sense when constructing the GVD from popular grid-based occupancy maps [4], [5]. Hybrid maps have been effectively demonstrated in some very difficult conditions such as mine exploration [26].

### Pruning the GVD

When used for navigation, the GVD must be heavily pruned to remove irrelevant edges [red lines in Figure 3(a)]. This includes spurs formed by perturbations in obstacle outlines and edges between obstacles that are too close for the robot to pass between. Edges that should be pruned cannot be identified by neighborhood functions in grid-based GVD models. Most commonly, analysis of edge endpoints and cycles (finding weak meet-points) is used to identify redundant edges [24], [25]. This can be complex and computationally expensive. Instead, two methods exist that directly generate an ideal graph of useful junctions. The first, recently published by Ko et al., [22] uses morphological thinning on free space. In the second, the authors have reported an alternative method that also produces a perfectly pruned GVD directly [27]. This linear-complexity method dilates obstacles by the radius of the robot (to eliminate spurious edges) before assigning a different label to the pixels of each obstacle. A modified



**Figure 3.** The problems encountered when trying to model a GVD-based topology using typical range sensors (sonar or laser) and our proposed solutions. (a) The true GVD is very complex, including both black and red lines. However, the red lines are irrelevant and should be pruned. (b) The benefit of coastal navigation. (c) and (d) GVD approximation via a custom DT. The labels of obstacles a, b, and c are propagated along with distance from these obstacles and stored as two separate images.

distance transform (DT) propagates each obstacle's unique label with increasing distance (see Figure 3). A simple  $3 \times 3$  neighborhood operation then gives the GVD. If a pixel has two different labels in its neighborhood, it lies on an edge. Neighborhoods with three different labels occur only at GVD junctions. The example in Figure 3(c) and (d) shows the images part-way through the first of two passes needed to compute the DT.

### Local or Global Hybrid Maps

Recently, a few researchers have moved toward global topological maps annotated with only local geometric maps of important places. Probably, these were inspired by Kuipers' spatial-semantic hierarchy, which describes how robust topological mapping and navigation can result from a set of complementary behaviors including local homing and edge traversal [28]. Other local or global hybrids arise from the simple expediency that detailed structural data is needed for localization, and localization is most critical where decisions are needed [6], [10], [25], [29].

### Local Topological Perception

Many authors have demonstrated the computation of the GVD after constructing a global map. A few have shown the construction of Voronoi-based topological models on the fly [20]–[22]. These tend to depict inaccurate topology, which in existing applications is not a significant problem. However, in our scenario, the robot must determine an accurate topology using only local geometric data that it accumulates while approaching a junction or following an edge. A few authors [notably Kuipers' [30] and Beeson's [31] local perceptual mapping (LPM)] have tried to dispense with global geometric data and, instead, have maintained a scrolling window of occupancy data in which the topology is modeled. However, since these authors use geometric data to distinguish places from each other, as well as for topological modeling, a much larger window is required. The limits of the local context are defined by the structure of local free space. In contrast, in this work, the local map is a fixed-size window that is always centered on the robot (we call this an egocentric map). The dimensions of the window are defined with respect to the robot's sensing capabilities, physical size, and modeled topology. In this way, we can ensure that the robot will always be able to correctly model local topology in the window (for a more detailed discussion, see [27]).

Modeling the topology using only local geometric data is more difficult when compared with that using global geometric data. Modeling topology on an approach is harder still, because there is no opportunity to correct initial mistakes. To generate consistent and smooth robot motion while modeling the topology and the obstacles in a fixed-size window, the topology must be designed carefully to be stable as new obstacles roll into the window. Critically, the GVD-derived topology must be modified to handle transitions to and from the open spaces larger than the range of the robot's sensors.

### The Extended GVD

Although the GVD is a useful structure for navigation in enclosed spaces such as corridors and rooms, it is less useful in large open

spaces where shortcuts are safe. However, an incrementally perceived GVD has another more severe problem in open spaces due to the limited range of contemporary laser and sonar sensors. In most mobile robots, sensor data is used to correct odometric error and keep the robot on course via fixed reference points in the world. When moving through a large space, with no obstacles in view, odometry must instead be used without corrective feedback. Worse still, when transitioning from open to obstructed space, the GVD can be radically altered by the detection of a solitary point obstacle. A pruned GVD topology is robust to changes in obstacle shape but is drastically altered by the appearance of a solitary obstacle in an open space. When using an egocentric map (which is a must for topologically directed navigation), this problem is exacerbated because distant obstacles that might stabilize the GVD are quickly forgotten.

The most common solution to this problem, which is also adopted in our work [27], is coastal navigation; i.e., following one wall around the boundary of large open spaces (Figure 3). Constant observation of obstacles makes navigation more replicable and geometric estimation more accurate. The technique is named after the practice of early seafarers, who hugged the coasts for fear of being lost at sea. This approach was also used with local (egocentric) maps as in [31]. It is worth mentioning that refusing to cross large open spaces can leave a robot stranded on an island of obstacles.

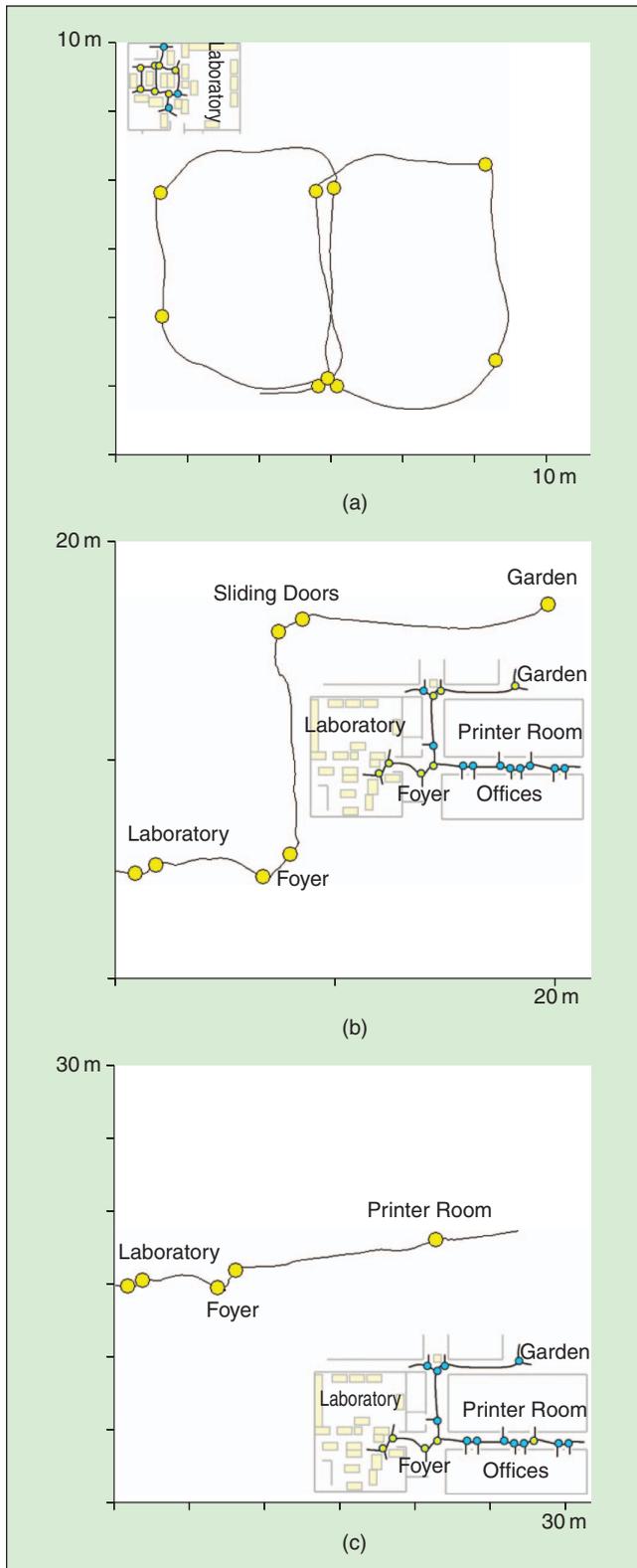
### Explicit Topologically Directed Navigation

In an earlier article [27], we detailed a method of directing our robot around the laboratory via a series of topological instructions that defined the robot's intended action at each junction. The article describes a GVD-derived topology of spatial affordances, in which junctions are defined by the physical capabilities of the navigating robot. Similar to the extended GVD, our topology follows walls in open spaces to ensure robust edge transition so that all features can be modeled egocentrically. The specified wall-following distance is calculated to maximize the stability of the egocentrically modeled topology even when obstacle detection is intermittent (see Figure 3).

In [27], the system was tested thoroughly in our laboratory, a realistic, cluttered indoor location (see Figure 1). Here, we present additional results in a mix of environments. In over 25 trials, the robot was instructed to navigate from the author's desk to all accessible nearby locations: a garden outside the building, to the other end of the building, to a road bordering the building, and the length of the upper floor of the building. The robot was also ordered to return to the author's desk from these locations. These environments were not modified to suit the robot. Figure 4 shows the paths of the robot in some of these trials, and the positions where junctions were detected.

### Topological Instructions

For these trials, the instructions provided were of two types. The first specified the angle at which the robot should try to depart a junction relative to the direction in which it approached. Precision was limited to  $90^\circ$  intervals; i.e., left, forward, or right. The robot selected and followed the edge that most closely matched the specified angle in its egocentric representation.



**Figure 4.** Three examples of explicit topologically directed navigation in which the robot was given a series of instructions dictating how it should behave at each junction. (a) A figure-8 route around the laboratory. (b) The robot was given the instruction “F,F,F,F,L,R,F” to go from the author’s desk to the garden. The same results were achieved with “F,F,F,1CW,1CW,1ACW,1ACW.” (c) From the author’s desk to the printer room.

The second type of instruction allowed more accurate determination of the edge that the robot should follow by avoiding angle estimation. Instead, edges were counted clockwise or anti-clockwise from the edge on which the robot approached the junction. For example, 1ACW means “take the first edge anti-clockwise from the approach edge.” Examples of the navigation achieved using both types of instruction can be seen in Figure 4. Both methods were successful in the test environments, but enumeration of edges is preferable for accurate navigation, especially where two edges have similar bearing on a junction.

## Metatopology of Behavior Change

Although our robot was able to accurately and incrementally perceive junctions during navigation, longer journeys are problematic using explicit topological directions because the topology of the environment is hard to remember accurately and may change during the robot’s movement. For example, doors may open and close.

We can imagine a more efficient and robust method of instructing the robot by relying on the fact that, generally, continuing in the same direction is most often the correct choice. Ideally, the human instructor should not have to specify the robot’s behavior at all junctions and, instead, describe only junctions where the robot’s behavior will be unusual. The robot should by default move straight over all junctions, unless some other action is specified. If the robot is commanded to turn right or left, it will afterward revert to traveling straight over all junctions.

Note that straight over is only a local definition: the direction in which the robot was traveling within its egocentric window. We assume that neither the robot nor the human director can accurately measure the direction of travel over long distances.

Note that the topology modeled by the robot is unchanged. It uses the GVD-derived topology of spatial affordances to describe the places at which navigational decisions must be made (junctions). However, the robot is now instructed on a metatopology, which is a graph of only the junctions in which behavior differs from default during a particular navigation task. This is similar to the concept of a layered hierarchical GVD [23].

This policy is convenient for the director, because it simplifies the instructions. It is more likely to be accurate because changes in topology between the few significant junctions do not affect navigation. Similarly, errors in the robot’s ability to detect junctions are less likely to be important. (However, errors at significant junctions are irrecoverable.)

It is interesting that when giving directions, we habitually omit possible turnings and tend to specify only changes in behavior, such as “follow the road until the bus stop, then go left.” This compresses the necessary instructions like a run-length-encoding: “do something many times, until landmark.” For most routes, this policy is very efficient; theoretically, the shortest path is always a straight line. Generally, extra turnings will only lengthen a route.

Since our robot already has the ability to select and traverse specific edges at a junction, it remains to define how the robot should interpret the default instruction to travel straight over a junction. This is easily achieved by ordering the robot to traverse the edge closest to 180° away from the edge along which it arrived at a junction. Since the robot continues to traverse only

the edges of its egocentrically modeled topology, it does not need accurate defaults nor will it crash trying to follow a path that does not exist.

### Topologically Teleoperated Demonstration

To show how efficient this policy could be, our robot was teleoperated across the School of Engineering at Monash University. The longest possible route was selected, with a small loop at the end. In the trials described previously and later, the robot was given no assistance beyond the initial set of topological instructions. Here, instead, the robot was accompanied by an operator who gave one of two instructions at any time: “turn left at the next junction” or “turn right at the next junction.” Figure 5 shows the robot’s route and where the robot was when the instructions were issued. Despite crossing 32 junctions, only five instructions were needed (excluding “start”). Similar efficiency could be expected in many environments. In the map shown, any room could be reached from any other using less than five instructions.

The authors suggest the term *topologically teleoperated* to describe this method of controlling the robot. It places very low demands on the operator and could have applications in scenarios where full teleoperation is difficult because of environmental conditions or the remoteness of the operator. For example, in a smoky building, the robot’s interpretation of sensor data might be better than that of its operator, yet the operator would still be necessary for gross guidance. This system allows both parties to cooperate effectively.

### Recognizing Junctions by Appearance

In later experiments, a panoramic camera was added to the robot, enabling it to recognize junctions previously visited. This allowed the robot to construct a topological map during exploration and to recognize places where specific navigational actions were necessary. As discussed previously, the convenience of directing a robot is greatly increased by needing intervention only when exceptional behavior is required. However, convenience is reduced if the operator has to accompany or remotely supervise the robot. If there was a way to identify, in advance, the junctions where turns should be made, the robot could again operate in a fully autonomous manner.

### Characteristics of Panoramic Images at GVD Junctions

We chose to use a panoramic camera to identify important junctions by their appearance. Vertices of a GVD-derived topology are finely localized in the ground plane. This means that, barring rotation, we can expect images captured on two

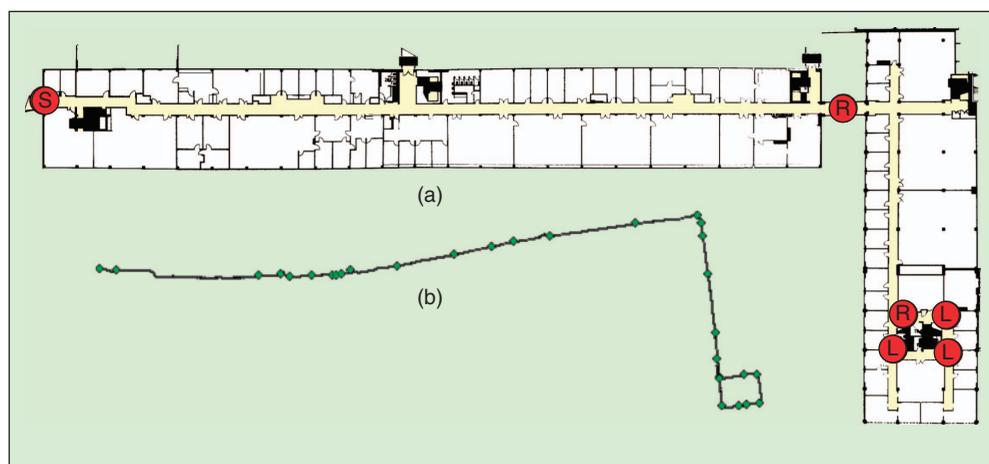
successive visits to a single junction to be very similar and easily matched. The greater the precision with which the robot positions itself for image capture, the easier the image-matching process will be.

Recognizing junctions by comparing panoramic images not only allows the construction of a topological map and the subsequent fully autonomous navigation within that map, as a conventional robot can do, it also allows operators to preemptively capture images at strategic locations to later direct an exploring robot. For example, capturing a few images with the camera on a tripod prior to comprehensive policy-based exploration could then mark the limits of the area to be explored. Dangerous areas could be excised from exploration, which is not possible using geometric constraints unless the environment is measured to start with, in which case the robot might as well be given a map.

### Matching Unwarped Panoramic Images of GVD Junctions

Since our experiments include places with variable, outdoor lighting conditions, the accuracy of GVD junction localization contributes significantly to the feasibility of the visual matching problem. GVD-derived topologies have few vertices and little aliasing. Many authors have exploited these properties, using panoramic images of GVD junctions to produce topological maps [32]. However, no one has examined the feasibility of topological mapping with panoramic images using only local geometric models to detect and localize junctions.

The literature provides many techniques for matching panoramic images. The solution used in our work was chosen under the assumption that accurate localization of junctions eliminates all geometric distortions between images except the rotation of the robot (i.e., the camera is identically placed for capturing all the images). For a high level of image discrimination, a two-stage approach was adopted. First, the Euclidean distance between pairs of thumbnail-sized, unwarped, normalized panoramic images was used to filter



**Figure 5.** Topologically directed teleoperation. The robot was instructed to travel straight over any junction unless given special instructions. On a handful of occasions, the robot was instructed to turn (left or right) at the next junction. Red circles denote these instructions. The position of the circle indicates the position of the robot when each instruction was issued. (b) The topological model produced by the robot during navigation, which shows all detected junctions.

the majority of false matches and recover a rough estimate of the rotation  $x$  between the pairs.  $x$  was then used in a second filter stage to accelerate the matching of scale-invariant feature transform (SIFT) key points [33] detected in full-size unwarped images. Key points were compared only with the possible corresponding key points within a small window predicted by  $x$ , and Figure 6 shows some examples.

**Algorithm 1** Preprocess red, green, blue (RGB) Image /

- 1  $N = \text{Normalize}(I)$
- 2  $U = \text{Unwarp}(N)$
- 3  $T = \text{Shrink}(N)$
- 4  $G = \text{Grayscale}(U)$
- 5  $K = \text{SIFT Key points}(G)$

**Algorithm 2** Similar (RGB Images  $I_1, I_2$ )

- $d$  = Euclidean distance between RGB thumbnails  
 $x$  = Estimated horizontal offset between unwarped images  
 $n$  = Number of SIFT feature correspondences
- 1  $d, x = \text{CompareThumbnails}(T_1, T_2)$ ;
  - 2 **if**  $d \geq t_1$  **then**
  - 3     **return false**
  - 4 **end if**
  - 5  $n = \text{MatchFeatures}(K_1, K_2, x)$
  - 6 **if**  $n \geq t_2$  **then**
  - 7     **return true**
  - 8 **end if**
  - 9 **return false**

**Results with Image Recognition**

The robot was required to complete two types of fully autonomous navigation tasks to test the robot’s ability to recognize, firstly, that it had arrived at a junction, and secondly, which junction it had reached. First, the robot completed an exhaustive exploration of the lab. Second, a few images of critical junctions (visual landmarks marked V in Figure 7) were manually captured and provided to the robot along with metatopological directions (see Figure 7). Using these, the robot navigated from the author’s desk to the garden and back four times using the same two visual landmarks. It navigated to the printer room and back using one visual landmark. It also explored the upper floor of the building twice, requiring only one landmark to turn at a critical point (see Figure 7). The robot was able to build coherent and increasingly comprehensive maps over a number of journeys by visually recognizing the junctions that it had previously encountered.

The robot was able to construct topological maps of these journeys. More interestingly, because it captured images at all junctions, it was able to combine maps of earlier and current explorations at the first common junction. This means that as soon as the robot intersects an earlier path it is able to exploit its existing knowledge. Figure 7(c) shows a topological map of the upper floor constructed from four passes through the building.

Figures 4, 5, and 7 use both the robot’s onboard odometry and loop closure via visual junction recognition to augment the presented topological graphs. The robot added this information during navigation. Although in these trials the robot navigated without advance geometric knowledge of its environment, the augmented graphs are useful in more conventional modes of navigation, including unguided exploration and autonomous generation of topological instructions based on the geometric positioning of vertices.



**Figure 6.** Matching images from different visits to locations in and around our laboratory. Dots with blue tails indicate correspondences between the top and bottom unwarped panoramic images. The robot autonomously chose where to capture each image. The robot’s camera was also taken to places it had not explored and was used to capture images with the given instructions. When the robot reached places matching a supplied image, it knew that it had to perform a specific behavior such as turning left or right. The precision with which junctions are defined greatly assists the image-matching problem.

## Conclusions: Toward Navigation by Visual Object Recognition

We defined a GVD-derived topology that can be perceived easily and accurately by robots and humans. It functions as a language by which navigational concepts can be transferred from humans' internal representations to a robot. The topology is ideally suited to navigation, as it is relevant to all environments and is minimally complex. The topology can be modeled correctly using only egocentric data essential for incrementally interpreting topological directions and crucial for directed navigation without prior digital mapping. It does not require people to estimate distances or angles accurately. Instead, instructions can be generated simply by remembering the environment. This form of navigation has not previously been demonstrated in realistic environments and might allow robots to function usefully in unfamiliar environments without first mapping exhaustively.

Subsequently, we tried to address problems in larger environments where the topology may change during navigation, making directions inaccurate, or the directions may be wrong simply due to peoples' inability to remember places correctly. It would be preferable to build a language for directing robots around constructs such as "after you have passed <object>" and "at the junction where there is a <object>." Instructions would still be based in the underlying topology of junctions but omitting junctions where no special behavior is required. A metatopology such as this maximizes convenience and reliability, but there is no easy way to identify critical junctions in isolation. For many everyday navigation tasks in realistic environments, changes in overall direction are rarely needed.

By equipping the robot with a panoramic camera, it was possible not only to produce topological maps conventionally (by visually recognizing junctions) but also to make the robot follow a specific route prior to mapping, using manually captured images as landmarks. In another experiment, the robot was accompanied and topologically teleoperated. Both these modes could have immediate practical uses, which are mentioned already.

In the introductory paragraph, we suggested that better object recognition skills may have more to offer navigating robots than do more accurate methods of geometric mapping. We support this claim by showing that our robot can navigate

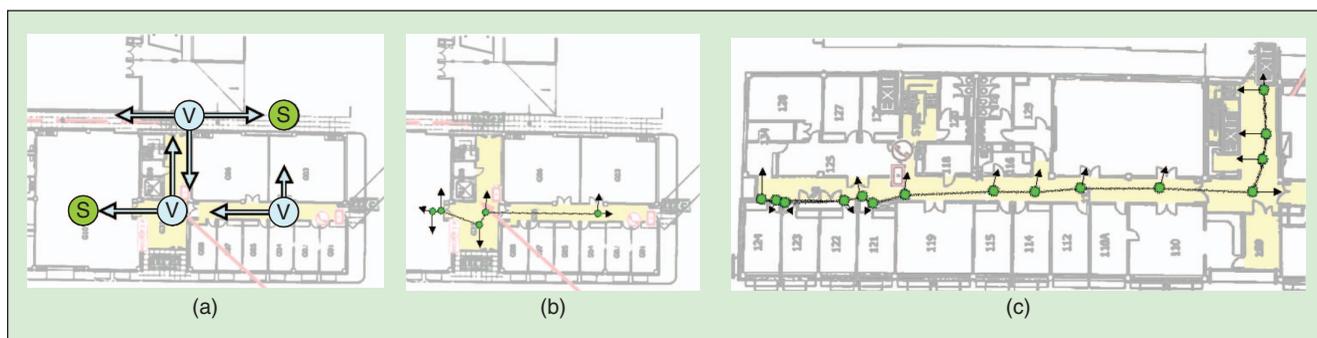
directly to goals with only high-level topological instructions and collecting any necessary geometric data along the way. The hard task is recognizing when to change behavior, which is unsurprising. Local navigation problems such as obstacle avoidance are well addressed in the literature, and, for much of the journey, little intelligence is needed. Only occasionally, at junctions, do errors become important and are quickly compounded if the wrong path is followed.

Better geometric mapping postpones errors at junctions by minimizing cumulative error, but it does not eliminate them. A key deficiency is the paucity of the structural (obstacle) data used, necessitating continuously accurate localization. In contrast, visual recognition of the appearance of scenes or objects at junctions instantly provides high confidence, simply because more data is available, without the need for accuracy between landmarks. This approach is well suited to existing vision sensors and techniques and was used successfully in our work.

The system presented here requires the robot to perceive a small vocabulary of high-level concepts, such as junctions and edges, of the topology. If the robot had a larger vocabulary of abstract objects (landmark objects such as chairs and trees), it would be possible to direct it using more efficient and intuitive metatopological instructions, such as describing places using language, without having to supply photographs in advance. In effect, simply adding object recognition to the existing system would offer major improvements in utility and convenience.

In some ways, following topological instructions is similar to following a topological plan in a conventional hybrid map. However, having only instructions makes the problem much harder because so much geometric and topological information is missing. Much of this data is not available to people or cannot be communicated verbally. Without it, the robot must interpret and respond to new topology as it is encountered. Compared with traditional explore-map-plan-navigate methods, in which the human must interpret the robot's map to specify goals, topological direction transfers work from humans to robots, requiring mutual perception of the same topology.

In future work, we would like to explicitly model uncertainty in the robot's progression through its instructions and consider whether it is possible to bring the concept of a



**Figure 7.** Topological mapping and topologically directed exploration. (a) Visual landmarks needed to replicate earlier experiments that directed the robot to and from the garden outside and down the corridor. (b) Topological map generated by one trip to the printer room (and back). (c) Topological map of a building (actually the result of four journeys back and forth). A visual landmark was needed at the corner of this L-shaped route and at both ends to stop each run.

junction closer to human norms. We would also like to integrate object recognition abilities and use these to direct the robot.

## Keywords

Topological, goal-directed navigation, Voronoi, panoramic vision.

## References

- [1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.
- [2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*, 2003, pp. 1151–1156.
- [3] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robot.*, vol. 4, no. 4, pp. 333–349, 1997.
- [4] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Comput.*, vol. 22, no. 6, pp. 46–57, 1989.
- [5] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. Ser. B*, vol. 39, no. 1, p. 138, 1977.
- [6] J. Folkesson and H. I. Christensen, "Robust SLAM," in *Proc. IFAC Symp. Intelligent Autonomous Vehicles (IAV)*, 2004.
- [7] C. Stachniss, D. Hahnel, and W. Burgard, "Exploration with active loop-closing for FastSlam," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2004.
- [8] R. Grabowski, P. Khosla, and H. Choset, "Autonomous exploration via regions of interest," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2003.
- [9] C. Stachniss, O. M. Mozos, and W. Burgard, "Speeding up multi-robot exploration by considering semantic place information," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2006.
- [10] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Hybrid simultaneous localization and map building: A natural integration of topological and metric," *Robot. Auton. Syst.*, vol. 44, no. 1, pp. 3–14, 2003.
- [11] Y. Wang, M. Huber, V. N. Papudesi, and D. J. Cook, "User-guided reinforcement learning of robot assistive tasks for an intelligent environment," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2003.
- [12] M. Levit and D. Roy, "Interpretation of spatial language in a map navigation task," *IEEE Trans. Syst. Man, Cybern.*, vol. 37, no. 3, pp. 667–679, 2007.
- [13] I.-P. Park and J. R. Kender, "Topological direction-giving and visual navigation in large environments," *Artif. Intell.*, vol. 78, nos. 1–2, pp. 355–395, 1995.
- [14] V. Setalaphruk, A. Ueno, I. Kume, Y. Kono, and M. Kidode, "Robot navigation in corridor environments using a sketch floor map," in *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation (CIRA)*, 2003, pp. 544–551.
- [15] K. Okada, T. Ogura, A. Haneda, J. Fujimoto, F. Gravot, and M. Inaba, "Humanoid motion generation system on hrp2-jsk for daily life environment," in *Proc. IEEE Int. Conf. Mechatronics and Automation*, 2005, vol. 4, pp. 1772–1777.
- [16] I. Nourbakhsh, R. Powers, and S. Birchfield, "Dervish: An office-navigating robot," *Artif. Intell. Mag.*, vol. 16, no. 2, pp. 53–60, 1995.
- [17] D. Hinkle, D. Kortenkamp, and D. Miller, "The 1995 robot competition and exhibition," *Artif. Intell. Mag.*, vol. 17, no. 2, pp. 31–45, 1996.
- [18] A. Ranganathan and F. Dellaert, "Data driven MCMC for appearance-based topological mapping," *Robot. Sci. Syst.*, vol. 41, pp. 209–216, June 2005.
- [19] A. Tapus and R. Siegwart, "Incremental robot mapping with fingerprints of places," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2005.
- [20] D. Silver, D. Ferguson, A. Morris, and S. Thayer, "Feature extraction for topological mine maps," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2004.
- [21] K. Nagatani, H. Choset, and S. Thrun, "Towards exact localization without explicit localization with the generalized Voronoi graph," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 1998.
- [22] B.-Y. Ko and J.-B. Song, "Real-time building of a thinning-based topological map with metric features," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2004.
- [23] H. Choset and J. Burdick, "Sensor-based exploration: The hierarchical generalized Voronoi graph," *Int. J. Robot. Res.*, vol. 19, no. 2, pp. 96–125, 2000.
- [24] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE Trans. Robot. Automat.*, vol. 17, no. 2, pp. 125–137, 2001.
- [25] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset, "The hierarchical atlas," *IEEE Trans. Robot. Automat.*, vol. 21, no. 3, pp. 473–481, 2005.
- [26] A. Morris, D. Silver, D. Ferguson, and S. Thayer, "Towards topological exploration of abandoned mines," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2005, pp. 2117–2123.
- [27] D. Rawlinson and R. Jarvis, "Topologically-directed navigation," *Robotica*, vol. 26, no. 2, pp. 189–203, 2008.
- [28] B. Kuipers, "The spatial semantic hierarchy," *Artif. Intell.*, vol. 119, pp. 191–233, 2000.
- [29] R. S. N. Tomatis and I. Nourbakhsh, "Simultaneous localization and map building: A global topological model with local metric maps," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2001, pp. 421–426.
- [30] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2004.
- [31] P. Beeson, N. K. Jong, and B. Kuipers, "Towards autonomous topological place detection using the extended Voronoi graph," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2005.
- [32] D. C. K. Yuen and B. A. McDonald, "Natural landmark based localization system using panoramic images," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2002.
- [33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

**David Rawlinson** received his B.Sc. degree in computer science and artificial intelligence from the Centre for Research in Cognitive Science (Sussex University) in 2000. He received his Ph.D. degree in autonomous visual navigation from Monash University in 2008. He has been working on the use of artificial intelligence for antimoney laundering, credit/debit card fraud prevention, logistics, timetabling and scheduling problems. He is currently working in video analytics for surveillance.

**Ray Jarvis** completed his B.E. and Ph.D. degrees in electrical engineering at the University of Western Australia in 1962 and 1968, respectively. After two years at Purdue University, he took up a senior lectureship at the Australian National University, where he was instrumental in establishing the Department of Computer Science. In 1985, he became the chair of the Department of Electrical and Computer Systems Engineering at Monash University, where he established the Intelligent Robotics Research Centre in 1987. He is a Fellow of the IEEE. His research interests include artificial intelligence, computer vision, pattern recognition, and intelligent robotics.

**Address for Correspondence:** Ray Jarvis, IRRC, Department of Electrical and Computer Systems Engineering, School of Engineering, Monash University, Melbourne, Australia. E-mail: ray.jarvis@eng.monash.edu.au.

# Hybrid Control for Robot Navigation

## A Hierarchical Q-Learning Algorithm

Autonomous mobile robots have been widely studied and applied not only as a test bed to academically demonstrate the achievement of artificial intelligence but also as an essential component of industrial and home automation. Mobile robots have many potential applications in routine or dangerous tasks such as delivery of supplies in hospitals, cleaning of offices, and operations in a nuclear plant. One of the fundamental and critical research areas in mobile robotics is navigation, which generally includes local navigation and global navigation [1], [2]. Local navigation, often called reactive control, learns or plans the local paths using the current sensory inputs without prior complete knowledge of the environment. Global navigation, often called deliberate control, learns or plans the global paths based on a relatively abstract and complete knowledge about the environment.

Local navigation is the fundamental ability for mobile robots, and several reactive control approaches [3]–[5] have been proposed and implemented for such local navigation tasks as obstacle avoidance, wall following, point-to-point moving, etc. For example, both the potential field method and the virtual force field method [3] are early reactive control methods to navigate in a local environment. Occupancy maps have been built using sonar sensors to model the environment, which has led to a series of localization and path-planning methods [4]. In the mid 1990s, behavior-

based control architecture was proposed and gained a lot of attention. A typical example is subsumption architecture [5], where the stimulus-action pairs are defined for the decision and control rules of a robot after a proper coordination. Various soft-computing and machine-learning methods also have been applied to reactive control, such as fuzzy logic, neural network, evolutionary computation, reinforcement learning (RL), etc. [6]–[9].

Although reactive control for local navigation can adapt to dynamic environments and be easily implemented, it is vulnerable to local minimum traps because of no prior information on the environment. Hence, many researchers aim at solving this problem. Deliberate control is usually used to carry out high-level planning using model-based methods. Hybrid control architecture, which combines reactive and deliberate control, gains the advantage from both worlds, i.e., the local and global optimization, and it has received considerable attention in the research area of mobile robotics [1], [2]. When we turn our attention to the issue of hybrid control for robot navigation, three problems should be regarded as critical and be carefully studied to guarantee a good performance.

- 1) *Control architecture*: Although hybrid control is mostly adopted for robot navigation, the practical design and its performance is determined by the realization of behaviors and models of the sensors and environments.
- 2) *Learning and control algorithm*: After the control architecture and detailed structure are designed, the second



Finding  
the Right  
Path

©DIGITAL VISION

Digital Object Identifier 10.1109/MRA.2008.921541

BY CHUNLIN CHEN, HAN-XIONG LI, AND DAOYI DONG

important problem is what kind of learning and control algorithm should be proposed and applied to the robot navigation system to drive the robot work effectively and efficiently.

- 3) *Behaviors design*: As the model of the environment has been chosen and constructed, robot behaviors have to be designed according to the sensor information and control architecture.

In this article, hybrid control architecture is conceived via combining reactive and deliberate control using a hierarchical Q-learning (HQL) algorithm. This control approach can be specified as follows. First, the grid-topological maps are constructed and maintained online to provide a model of the environment, and then, hybrid control architecture is proposed based on the grid-topological maps. Second, a novel HQL algorithm is presented based on the hybrid Markov decision process (MDP), which works as an integrated learning and control algorithm for the proposed hybrid control architecture. Third, a mobile robot ATU is introduced and a target identification method using a multi-ultrasonic sensor system is presented for the identification of landmarks or features of the environment. Finally, detailed behaviors for reactive control and deliberate control are designed based on the presented hybrid control architecture and the configurations of ATU. To verify the performance of the presented methods, an example of indoor navigation is demonstrated via both simulated and real experiments.

## Hybrid Control Using HQL Method

In this section, hybrid control architecture is introduced based on the grid-topological representation. Then, a novel HQL algorithm is presented and applied to hybrid control architecture.

### Grid-Based Map and Topological Map

The grid-based map and the topological map are two fundamental paradigms for modeling indoor robot environments. The grid-based approaches [4] represent environments with evenly spaced grids, and each grid cell may indicate whether the corresponding region of the environment is free or occupied space. The topological approaches [10] represent the

environments with graphs, where nodes correspond to distinct situations, places, or landmarks. The nodes are connected by arcs if there is a direct path between them. Both these approaches for robot mapping hold orthogonal strengths and weaknesses [11]. For example, the occupancy grids are easy to construct and to maintain in large-scale environments but suffer from enormous space and time complexity. The topological approaches permit fast planning and provide more natural interfaces for human instructions (such as “go through the corridor to place A”), but they have difficulties in recognizing different places geometrically even in static environments, which makes it unfeasible to construct large-scale maps because of the uncertainty of sensor information and qualitative reasoning. Hence, it is natural to integrate the two paradigms, i.e., the grid-topological approach, to achieve a good performance of robot navigation. The approach adopted in this article is based on the grid-topological representation.

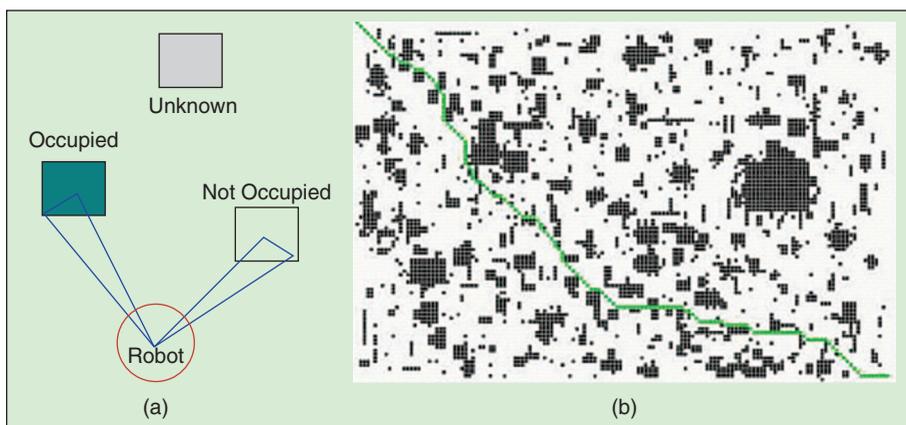
The grid-based maps considered here are discrete, two-dimensional occupancy grids. Each grid cell  $(x, y)$  in the map has related value  $P_{Occ}(x, y) \in [0, 1]$  that measures the subjective belief that the cell is occupied. The values  $P_{Occ}(x, y)$  are updated based on the sensor information such as the data from a multi-ultrasonic sensor system. Suppose the threshold value is  $\delta \in (0, 1)$ , then there is the criterion of  $P_{Occ}(x, y)$  for the decision whether cell  $(x, y)$  is occupied

$$\begin{cases} P_{Occ}(x, y) < \delta, & \text{cell}(x, y) \text{ is Empty} \\ P_{Occ}(x, y) > \delta, & \text{cell}(x, y) \text{ is Occupied} \\ P_{Occ}(x, y) = \delta, & \text{cell}(x, y) \text{ is Unknown.} \end{cases} \quad (1)$$

Figure 1(a) shows how the probability of grid-cell occupancy is determined through perception. A mobile robot with a range of sensors such as sonar sensors, infrared light sensors, or cameras can perceive the circumstance around it while moving in an unknown environment and memorize by updating the grid-based maps. A typical navigation task in a grid-based world is described in Figure 1(b), which shows how the mobile robots work with a well-built grid-based map.

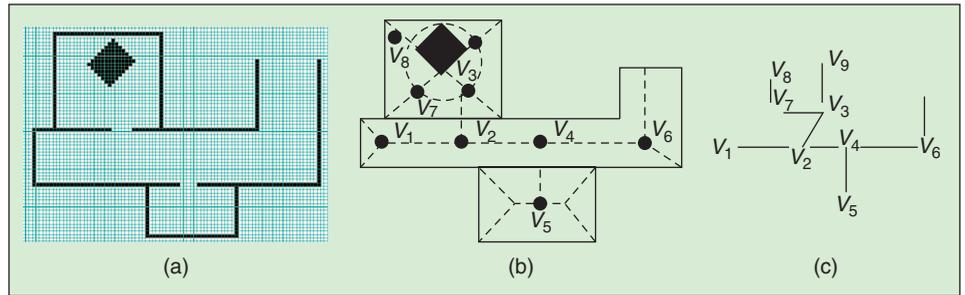
The topological maps are abstract maps and can be built on top of the grid-based maps. Unlike the approaches of Thrun [11], which extract the topological map from the grid-based map using region decomposition, we directly use Voronoi diagram [11] to extract the topological structures from the grid or metric map, which maintains the qualitative space information of the states and the relations between these states. As shown in Figure 2, the procedure for topological map building is as follows:

- 1) *Show free space*: Set a proper threshold value  $\delta \in (0, 1)$  and show free space  $C$  and unreachable space  $\bar{C}$  [black cells in Figure 2(a)].
- 2) *Voronoi graph*: For any cell in free space  $p = (x, y) \in C$ ,



**Figure 1.** Grid-based approach: (a) grid-cell occupancy detection using range sensors, and (b) a typical navigation task in a grid-based world.

there are points in  $\bar{C}$ , called basis points, that are nearest to  $p$ . If there are more than one basis point for  $p$ ,  $p$  is defined as a Voronoi point and all of these Voronoi points constitute a Voronoi graph, which is shown as the dashed line in Figure 2(b).



**Figure 2.** Construction of topological map: (a) grid-based map, (b) mask distinctive region states in the Voronoi diagram, and (c) topological map.

3) *Extract topological nodes:* Extract the Voronoi points

that have more than two adjacent Voronoi points and mark them as nodes, such as  $V_1, V_2, \dots, V_9$  in Figure 2(b); those nodes in the same free region are regarded as one, such as node  $V_5$  in Figure 2(b).

4) *Connect the nodes with arcs:* Connect all the nodes that are reachable with corresponding arcs, and then the topological map [Figure 2(c)] is constructed and may be written as  $M: \{\text{Nodes, Arcs}\}$ , where

$$\text{Nodes} : \{V_i, i = 1, 2, \dots, n\}$$

$$\text{Arcs} : \{(V_i, \text{Action}_i, V_j), i, j = 1, 2, \dots, n\}, \text{Action}_i \text{ is the qualitative action that moves from node } V_i \text{ to } V_j.$$

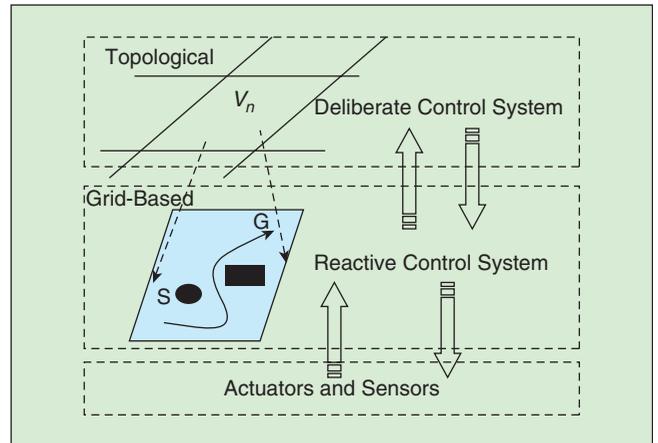
The topological maps can be updated and expanded by the information about the environment that the robot gains while navigating. The map updating procedure works as follows:

- 1) check for the state with approximate equidistant obstacles in more than two directions
- 2) choose adjacent states for further affirmation, then mark it as a landmark state, and match it with all the nodes  $V_i$  in the topological map  $M$ ; 1) if it matches, denote it by  $l_i$ ,  $l_i \in V$ ; 2) if it does not match, denote it by  $l_{n+1}$ , and put it into map  $M$  as a new node  $V_{n+1}$  with a new arc  $(V_{i-1}, \text{Action}_{i-1}, V_{n+1})$ .

In the process of navigation, besides the methods based on the grid model, the topological maps may also be updated and expanded by the information of the typical environment features that the robot identifies. Various approaches [2], [12] may be applied to recognize different regions using different sensors. In our research, the target identification methods based on a multi-ultrasonic sensor system are adopted to help for further affirmation, where typical environments (e.g., door, corner, and wall) are recognized by sensor fusion. This kind of information can help the robot to build the environment map and make localization. The next section will introduce the details of the target identification methods.

### Hierarchical Control Architecture

On the basis of the grid-topological model of the environment, hybrid control architecture is proposed that mainly consists of two control levels: reactive control on the grid-based map and deliberate control on the topological map. The coordination between the two levels is guaranteed by the reasoning and information transmission between the grid and topological maps, and the implementation of this hybrid control structure is based on the HQL control algorithms, which will be introduced in the next section. Supposing that the qualitative action  $\text{Action}_i$  in the topological map corresponds to an action sequence



**Figure 3.** Hybrid navigation control architecture.

$\{a_{i1}, a_{i2} \dots a_{im}\}$  in the grid-based map, we have the following two lemmas [11], which are used to guarantee the existence of a deterministic path between the two connected nodes in a topological map and a counterpart path in the grid-based map.

**Lemma 1:** *In the topological map, if two nodes  $V_i$  and  $V_j$  are connected, there exist deterministic qualitative actions  $\text{Action}_i$  to move robot from  $V_i$  to  $V_j$ .*

**Lemma 2:** *Each path in the topological map has a counterpart in the grid-based map.*

Figure 3 shows the overall navigation architecture that consists of the reactive control system, deliberate control system, and the actuator-sensor system. The reactive control system, which is accomplished based on the grid map [Figure 2(a)], supports the functions of obstacle avoidance, subgoal implementation, and emergency handling. The deliberate control system, which is based on the topological map [Figure 2(c)], plans to achieve the global goal and to coordinate from the upper control level. The whole learning and control architecture is implemented using the HQL algorithm that integrates the multiple control levels and holds a good performance.

### HQL Method

Proper learning and control algorithm is the key to guarantee the performance of robot navigation. In this section, the main idea of HQL is presented based on the model of hybrid MDP. For a more detailed analysis about HQL, see [13].

## Hybrid control architecture, which combines reactive and deliberate control, gains advantage from both worlds, i.e., local and global optimization.

Q-learning is one of the most widely used RL algorithms in many areas [14], [15], especially in robot motion control and navigation. One-step Q-learning is the basic Q-learning algorithm, in which the key formula to modify the Q values is as follows:

$$Q(s_t, a_t) \leftarrow (1 - \eta)Q(s_t, a_t) + \eta(r_{t+1} + \gamma \max_{d'} Q(s_{t+1}, d')), \quad (2)$$

where  $Q(s_t, a_t)$  is the value function of state-action pair at the moment  $t$ ,  $\eta$  is the learning rate,  $\gamma$  is a discount factor, and  $r_{t+1}$  is the reward received when taking action  $a_t$  at state  $s_t$ .

The standard Q-learning learns on a very large flat state-action space [14]. When the Q-learning algorithms are applied to autonomous mobile robots, there are too many limitations. For example, the curse of dimensionality occurs because of the expansion of state-action space, the environments have to be modeled as MDPs and the tasks are mainly confined to the reactive ones. Obviously, it is impractical and insufficient for real mobile robots because the environments are too complex to be observed exactly. Many tasks are not simple, and deliberate control is also necessary. To solve these problems, some researchers in the area of machine learning and robotics have tried different approaches, which can be classified into two categories: modular methods [16], [17] and hierarchical RL methods [18].

The modular methods [16], [17] are the most straightforward approach for the complex tasks that divide the problem into modules according to different functions. That is, divide-and-conquer policy is applied with subagents, and the subagents solve the conflicting desires through negotiation. In hierarchical RL, *hierarchical* mainly emphasizes the temporal abstraction. Barto and Mahadevan [18] reviewed several related approaches to

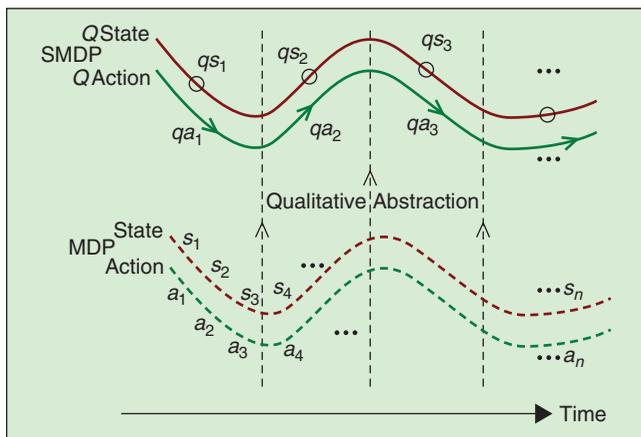


Figure 4. Sketch of qualitative abstraction from MDP to SMDP.

temporal abstraction and hierarchical control that had been developed in machine learning. According to their research, there are mainly three approaches to the hierarchical RL: 1) the options formalism [19], 2) the hierarchies of abstract machines approach [20], and 3) MAXQ value function decomposition framework [21]. All these approaches rely on the theory of semi-Markov decision processes (SMDP) to provide a formal basis and use a simple type of abstraction called macro. In fact, the hierarchical approaches to RL generalize the macro idea to closed-loop policies, which are sometimes called options, skills, behaviors, or temporally extended actions. The hierarchical RL methods will be good candidates to the proposed hybrid navigation architecture. But, most of the existing methods emphasize the decomposition of tasks and lack a mechanism to guarantee the coordination of different parts and learning levels. Hence, in this article, we propose to formulate the abstraction in a hierarchical RL framework from the point of view of quantitative-qualitative operation and present an integrated learning algorithm for hybrid control to coordinate different control levels and speed up the learning process.

The standard framework of RL is based on discrete-time, finite MDP [15]. But, when it is expanded to a hierarchical one, the upper-level learning should be modeled as SMDP. The HQL is based on a hybrid MDP that combines the MDP and SMDP for lower level and upper level, respectively. There are some definitions and propositions for the framework of HQL.

**Definition 1 (MDP):** MDP consists of five factors  $\{S, A_{(i)}, p_{ij(a)}, r_{(i,a)}, V, i, j \in S, a \in A_{(i)}\}$ , where  $S$  is state space;  $A_{(i)}$  is action space for state  $i$ ;  $p_{ij(a)}$  is probability for state transition from state  $i$  to state  $j$  with the action  $a$ ;  $r$  is reward function,  $r: \Gamma \rightarrow [-\infty, +\infty]$ , where  $\Gamma = \{(i, a) | i \in S, a \in A_{(i)}\}$ ;  $V$  is criterion function or objective function.

In qualitative algebra, the quantitative-qualitative issues are regarded as different angles of view on respective levels [22], [23]. Hence, we can look upon the grid-based map and topological map as quantitative and qualitative representations, respectively. The high-level (qualitative) representation may be defined on the abstracting of the low-level (quantitative) representation. As shown in Figure 4, SMDP may be constructed by the qualitative abstraction from MDP, where  $qs_i$  is defined as the qualitative abstraction of a group of local states in a certain region and  $qa_i$  is the counterpart of qualitative action corresponding to the sequential actions in this region.

**Definition 2 (SMDP):** SMDP is composed of six factors  $\{S, A_{(i)}, p_{ij(a)}, T_{(\bullet|i,a,j)}, r_{(u,i,a,j,t)}, V, i, j \in S, a \in A_{(i)}\}$ , where  $S$  is state space;  $A_{(i)}$  is action space for state  $i$ ;  $p_{ij(a)}$  is probability for state transition from state  $i$  to state  $j$  with the action  $a$ ; the time of transition from state  $i$  to  $j$  is a nonnegative stochastic variable  $T_{(\bullet|i,a,j)}$ ; if the transition time is  $t$ , the reward is  $r_{(u,i,a,j,t)}$  in the time of  $[0, u] (u \leq t)$ .

The policy in an SMDP is defined as  $Q\pi = \{q\pi_i\}: QS \times \cup_{i \in S} QA_{(i)} \rightarrow [0, 1]$ . Under the policy  $Q\pi$ , the qualitative state value function (with the state transition time  $\tau$ ) is  $V_{(qs)}^{q\pi} = E\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots + \gamma^{\tau-1} r_{t+\tau} + \dots\}$ .

**Proposition 1:** Under the policy  $Q\pi$ , if  $u = t$ , let

$$\begin{aligned} r_{(u,i,qa,j,t)} &= r_{(i,qa)} = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots + \gamma^{\tau-1} r_{t+\tau} \\ P(qs' | qs, qa) &= \gamma^\tau P(qs', \tau | qs, qa) \end{aligned}$$

the SMDP:

$$\{QS, QA_{(i)}, P_{ij(qa)}, T_{(\bullet|i,a,j)}, r_{(u,i,qa,j,t)}, V, i, j \in QS, qa \in QA_{(i)}\}$$

is equivalent to the MDP:

$$\{QS, QA_{(i)}, P_{ij(qa)}, r_{(i,qa)}, V, i, j \in QS, qa \in QA_{(i)}\}.$$

**Proof (Sketch):** Both the MDP and SMDP consist of 1) a set of states, 2) a set of actions, 3) probabilities for state transitions, and 4) a well-defined reward function for the state transition. The difference between them is that an SMDP uses a time variable  $u$  to explore the state transition embedded within an MDP. If the cumulated rewards of state transition are defined as

$$r_{(u,i,qa,j,t)} = r_{(i,qa)} = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots + \gamma^{\tau-1} r_{t+\tau},$$

the probability is  $P(qs' | qs, qa) = \gamma^\tau P(qs', \tau | qs, qa)$ , and the five-factors  $\{QS, QA_{(i)}, P_{ij(qa)}, r_{(i,qa)}, V\}$  is an MDP.

From Proposition 1, we know that the SMDP of the qualitative environment model remains the characteristics of MDPs, and under certain conditions, the techniques such as dynamic programming, and RL can also be adopted to solve the related problems.

The qualitative state-action value function is

$$Q_{(qs,qa)}^{q\pi} = E\{r_{(qs,qa)} + \gamma^\tau r_{(qs',qa')} \dots\} \quad (3)$$

$$Q_{(qs,qa)}^* = [r_{(qs,qa)} + \sum_{qs'} P(qs' | qs, qa) \max_{qa \in QA_s} Q_{(qs',qa')}^*]. \quad (4)$$

Let  $\eta$  be the learning rate; accordingly, the one-step updating rule for the qualitative state-action value function is

$$Q(qs, qa) \leftarrow (1 - \eta)Q(qs, qa) + \eta(r_{(qs,qa)} + \gamma^\tau \max_{qa'} Q(qs', qa')). \quad (5)$$

Learning in a high-level abstract space can speed up computing remarkably because of the dramatic reduction of state-action space. But, the qualitative abstracting of the origin problem always loses some information, and we can get only a suboptimal policy and imprecise control sequences by the learning and planning in a qualitative space. Hence, in the hybrid navigation control, it is necessary to optimize the learning process by taking utmost advantage

of local precise information. The combination of the qualitative reasoning and quantitative computing [23] is a good candidate, which is also the motivation of the HQL based on hybrid MDP. ■

**Definition 3 (Hybrid MDP) SMDP:**

$\{QS, QA_{(i')}, P_{i'j'(qa)}, T_{(\bullet|i',a,j')}, r_{(u,i',qa,j',t)}, V, i', j' \in QS, qa \in QA_{(i')}\}$  is the qualitative abstraction of MDP:

$\{S, A_{(i')}, P_{i'a}, r_{(i',a)}, V, i', j' \in S, a \in A_{(i')}\}$ , let

1)  $HS = QS \cup S$ , landmark state  $l \in HS, QS, S$

2)  $HA = QA \cup A$

3)  $P = \{p_{ij}\}$ , where  $p_{ij} = \begin{cases} p_{i'j'(a)} & \text{if } i = i', j = j' \\ p_{i'j'(qa)} & \text{if } i = i', j = j'' \\ 0 & \text{otherwise} \end{cases}$

4)  $R = \begin{cases} r_{(u,i',qa,j',t)} & \text{if } i'' \in QS \\ r_{(i',a)} & \text{if } i' \in S \end{cases}$

5)  $H\pi = \pi \cup Q\pi$ ,  $\pi = \{\pi_i\}$ ,  $Q\pi = \{q\pi_j\}$ , where  $\pi_i$  is for quantitative state  $s_i \in S$ , and  $q\pi_j$  is for qualitative state  $qs_j \in QS$ .

The five-factors  $\{HS, HA, P, R, V\}$  is defined as a hybrid MDP on the state universe of HS.

Definition 3 shows a hybrid formulation of the complex problems with abstraction, where a landmark state is the key to connect the different representation levels. In the HQL algorithm, the Q values of landmarks  $Q(l, a)$  are updated using the qualitative Q values with a decreasing learning step  $\alpha^m$  for the  $n$ th updating,  $\alpha \in [0, 1)$  and  $m > 1$ . According to the earlier discussion, the HQL algorithm based on the hybrid MDP  $\{HS, HA, P, R, V\}$  is described as Algorithm 1, and the diagram of this algorithm is shown in Figure 5.

Bertsekas and Tsitsiklis [24] have proved the convergence of Q-learning. As for HQL algorithm, it is based on hybrid MDP  $\{HS, HA, P, R, V\}$  and has no essential difference from the view of convergence according to Proposition 1. If the agent uses a discount factor  $\gamma$ , such that  $0 \leq \gamma < 1$ , and

$$\lim_{T \rightarrow \infty} \sum_{k=1}^T \eta_k = \infty, \quad \lim_{T \rightarrow \infty} \sum_{k=1}^T \eta_k^2 < \infty \quad (6)$$

#### Algorithm 1 HQL Algorithm

- 1) Initialize  $Q(hs, ha)$  arbitrarily
  - 2) Repeat (for each episode):
    - a) Initialize  $hs$
    - b) Repeat (for each step of episode):
      - I) Choose  $ha$  from  $hs$  using policy derived from  $Q$  (e.g.,  $\epsilon$ -greedy)
      - II) Take action  $ha$ , observe  $R, hs'$ 
        - i)  $Q_k(hs, ha) \leftarrow Q_{k-1}(hs, ha) + \eta_k [R + \gamma \max_{a'} Q_{k-1}(hs', ha') - Q_{k-1}(hs, ha)]$
        - ii) If  $hs \in QS$ , update the Q value for corresponding landmark state  $l$  and action  $a$   $Q(l, a) = (1 - \alpha)Q(l, a) + \alpha Q(hs, ha)$ ,  $a \in A$ ,  $\alpha \in [0, 1)$   $\alpha = \alpha^m$ ,  $m > 1$  is a scalar constant
        - iii)  $hs \leftarrow hs'$
- Until  $S$  is terminal  
Until the learning process ends

for all  $hs$  and  $ha$ ,  $Q_k(hs, ha)$  will converge to the optimal state-action value function  $Q^*$  as  $k \rightarrow \infty$ , with probability 1. For a detailed proof, refer to [13].

## Target Identification and Behavior Design

The sketch of the overall hybrid controller using HQL is shown in Figure 6. Besides the integral hybrid control architecture and algorithms, the method for environment recognition and behaviors for multiple control levels have to be designed to accomplish the navigation task. In this section, first, the robot model and its sensor system are described. Second, a target identification method is proposed for the environment recognition based on a multi-ultrasonic sensor system, which is the key technique for the upper-level planning and reasoning. Finally, the qualitative and quantitative behaviors are realized for the deliberate and reactive controls, respectively, based on the grid-topological map.

### Robot Model and Sensor Arrangements

The robot model is based on a mobile platform called ATU [Figure 7(a)], which we have made for the experimental research of navigation using multi-ultrasonic sensors. It is a two-wheel driven robot with 18 sonar sensors, and some of its specifications are as follows:

- ◆ Dimensions: Base  $d = 50$  cm, height = 1 m, weight = 15 kg
- ◆ Can-bus modules: SJA1000 at PeliCAN mode, PCA82C200
- ◆ Motor control modules: two dc motors (maxon RE36) with two shaft encoders
- ◆ Multisonar sensor modules: 18 sonar sensors (EFR-40RSC)
  - ◆ Resonance frequency: 40 kHz
  - ◆ Sensitivity range: 0.2–5 m
  - ◆ Diversity angle:  $-15$ – $15^\circ$

- ◆ Microcontrollers: 80c196kc accompanied with ROMs and RAMs.

There are also some other sensors such as the charge coupled device camera, infrared light sensors and photosensitive sensors, which could be equipped to the robot conveniently when necessary because of the open architecture of ATU based on CAN-bus.

The robot model in a world reference frame is shown in Figure 7(b).  $d_i$  is the distance between the robot and obstacles, which can be obtained from the sonar data.  $d_g$  is the distance to the goal, and  $\phi_g$  is the angle between the direction of the goal and the orientation of the robot.

The arrangement of the multi-ultrasonic sensor system is two layered [Figure 8(a)], and its planform is shown in Figure 8(b), where the 18 sonar sensors can be grouped into six, which can provide two kinds of distance information for the environment perception and behavior control, i.e., the distance information of 18 directions  $\{d_1, d_2, \dots, d_{18}\}$  and the compact distance information of six directions  $\{D_1, D_2, \dots, D_6\}$ .

### Target Identification Based on Multiultrasonic Sensor System

Besides the environment perception, the multi-ultrasonic sensor system can be used for the target identification in an indoor environment. In this article, different environment features are classified, and a multi-ultrasonic sensor system is used to provide relative time-of-flight (TOF) information [4]. According to the TOF information, different decisions are made through Dempster-Shafer [25] evidential reasoning and further active detection. After preprocessing of the sonar data, we have the TOF information about the obstacles around the robot:  $\{d_1, d_2, \dots, d_{18}\}$ . According to the arrangement as shown in Figure 8(a), three adjacent sonar sensors on the same ring (the upper ring or the lower ring) can be selected dynamically, i.e.,  $S_l$ ,  $S_c$ ,  $S_r$  (left, center, right), whose TOF information is  $(d_l, d_r, d_c)$  as shown in Figure 8(c). According to the features of the indoor environment, we define four simple prototype environments:

- ◆ plane: walls and large planes
- ◆ corner: corners in a room
- ◆ door: doors or narrow corridors
- ◆ cylinder: cylinders or small obstacles.

Therefore, we can get the characteristic curves of the TOF information  $(d_l, d_r, d_c)$  for the selected sonar sensors (Figure 9). There are a cluster of characteristic curves for any of the four defined

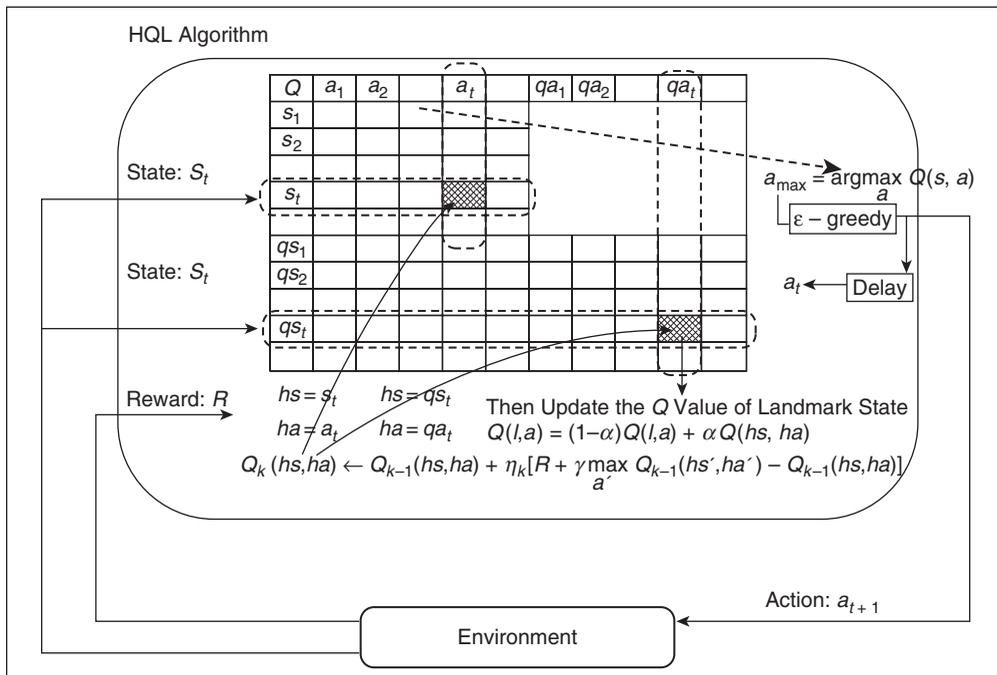


Figure 5. Diagram of HQL algorithm.

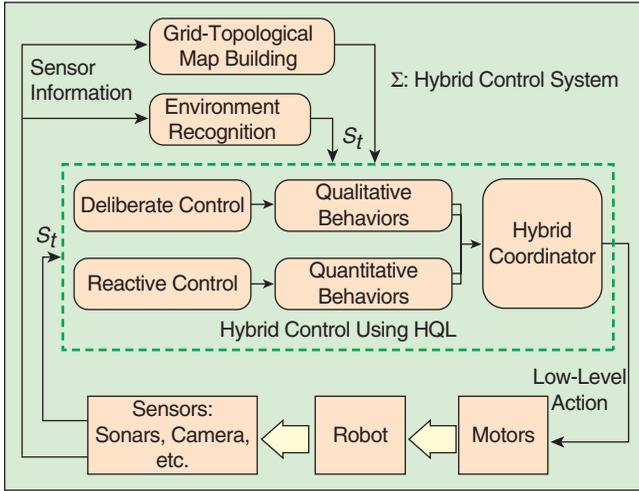


Figure 6. Hybrid controller based on HQL.

prototype environments because of the different view points of the robot. Hence, the TOF characteristic curves are determined by the relations of the distance values ( $d_l, d_r, d_c$ ) instead of the absolute values of them. The values of ( $d_l, d_r, d_c$ ) correspond to the points that are at the same curve in Figure 9 for each measurement. In the proposed approach, the parameters of the TOF characteristic curves are set according to the configuration of the mobile robot platform ATU, and related target identification method is designed and implemented. The main idea of this method is that the probability of the target type is given by the characteristic curves using fuzz logic, and then, sensor fusion is accomplished using Dempster-Shafer theory.

For all the prototypes (plane, corner, door, and cylinder), the basic probabilities are  $m(p)$ ,  $m(cn)$ ,  $m(d)$ , and  $m(cl)$ . These probabilities are defined using the membership function of Bell fuzzy sets [26] shown in Figure 10:

$$m(p) = \exp[-((\Delta l - 0.1)/0.1)^2] \quad (7)$$

$$m(cn) = \exp[-((\Delta l + 0.1)/0.1)^2] \quad (8)$$

$$m(d) = \begin{cases} \exp[-((\Delta l + 1)^2)0.1] & \text{when } \Delta l > -1 \\ \exp[-((\Delta l + 1)^2)5] & \text{when } \Delta l \leq -1 \end{cases} \quad (9)$$

$$m(cl) = \begin{cases} \exp[-((\Delta l - 1)^2)0.1] & \text{when } \Delta l < 1 \\ \exp[-((\Delta l - 1)^2)5] & \text{when } \Delta l \geq 1, \end{cases} \quad (10)$$

where  $\Delta l = (d_l + d_r)/2 - d_c$ ,  $d_l$ ,  $d_c$ , and  $d_r$  are the distance data of the dynamically selected sonar sensors  $S_l$ ,  $S_c$ , and  $S_r$ , respectively. The parameters for these Bell fuzzy sets are defined empirically based on the performance of sonar sensor EFR-40RSC and the TOF curves (Figure 9). According to equations (7)–(10), let  $\delta > 0$  be a pointed distance value and Max be the maximum range of the sonar sensors, we know that

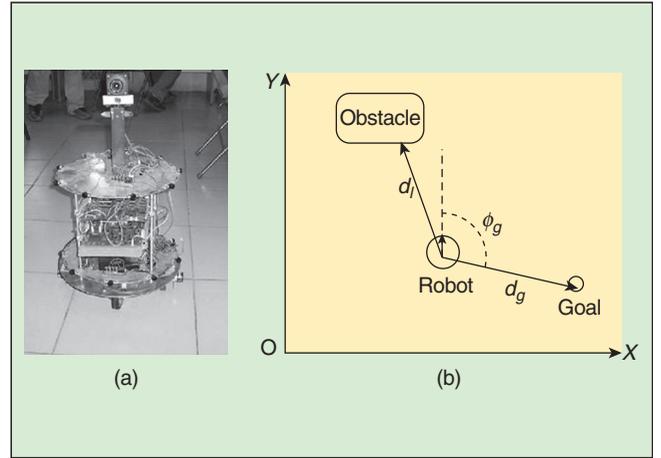


Figure 7. (a) Mobile robot ATU. (b) Robot model in world reference frame.

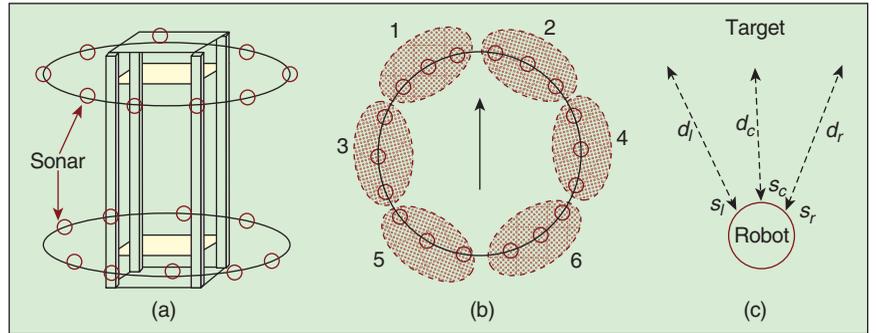


Figure 8. The multi-ultrasonic sensor system: (a) arrangement, (b) planform, and (c) the TOF information of the three neighbor sonar sensors selected for target identification.

$$\text{type} = \begin{cases} \text{Plane} & \Delta l \in (0, \delta) \\ \text{Corner} & \Delta l \in (-\delta, 0) \\ \text{Door} & \Delta l \in (-\text{Max}, -\delta) \\ \text{Cylinder} & \Delta l \in (\delta, \text{Max}). \end{cases}$$

The probability of unknown objects is defined as  $m(u)$ :

$$m(u) = \max(0, 1 - m(p) - m(cn) - m(d) - m(cl)). \quad (11)$$

The Dempster-Shafer theory is applied to the fusion of the TOF information for the same target. Then, we have

$$\begin{aligned} m(p) &= k[m_i(p)m_j(p) + m_i(p)m_j(u) + m_i(u)m_j(p)] \\ m(cn) &= k[m_i(cn)m_j(cn) + m_i(cn)m_j(u) + m_i(u)m_j(cn)] \\ m(d) &= k[m_i(d)m_j(d) + m_i(d)m_j(u) + m_i(u)m_j(d)] \\ m(cl) &= k[m_i(cl)m_j(cl) + m_i(cl)m_j(u) + m_i(u)m_j(cl)], \end{aligned} \quad (12)$$

where

$$\begin{aligned} k^{-1} &= m_i(p)m_j(p) + m_i(cn)m_j(cn) + m_i(d)m_j(d) + m_i(cl)m_j(cl) \\ &+ m_i(u)m_j(p) + m_i(u)m_j(cn) + m_i(u)m_j(d) + m_i(u)m_j(cl) \\ &+ m_i(p)m_j(u) + m_i(cn)m_j(u) + m_i(d)m_j(u) + m_i(cl)m_j(u) \\ &+ m_i(u)m_j(u). \end{aligned}$$

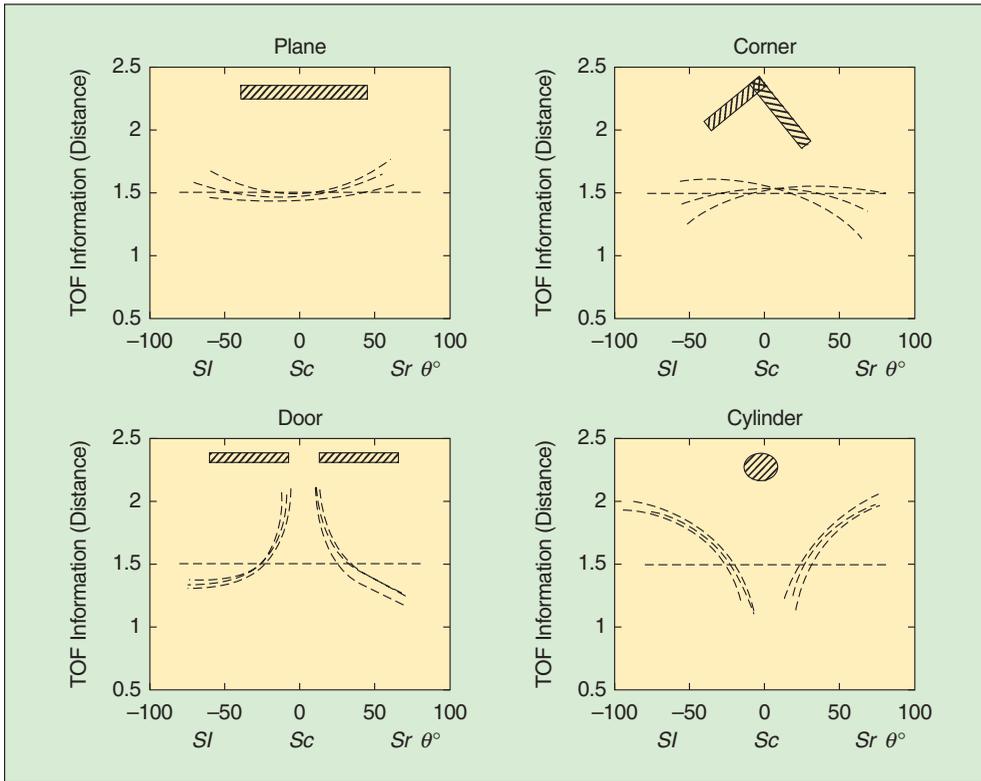


Figure 9. Sketch of target primitives and TOF characteristics.

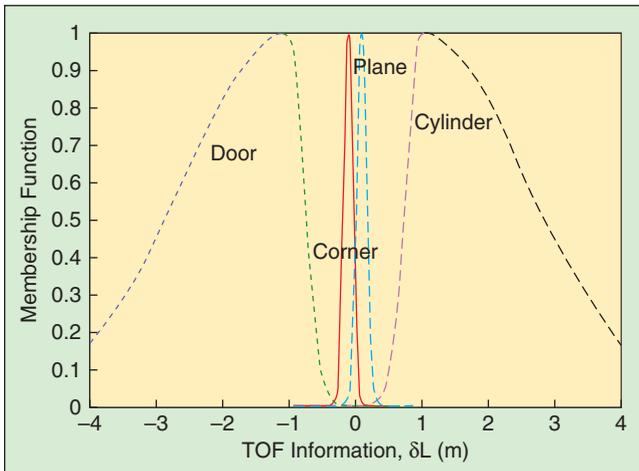


Figure 10. Fuzzy sets for target identification.

After the sensor fusion on all the data collected, the decision of the target is given as

$$\text{type} = \arg \max (m(\text{type})), \quad (13)$$

where  $\text{type} \in \{p, cn, d, c\}$  is the prototype of the environment. Equation (13) means to select the type whose probability  $m(\text{type})$  is maximum as the output. This target identification method takes the most advantage of the acquired limited sensor data and provides an effective approach of environment recognition to help for the searching of landmarks or subgoals.

### Qualitative Behaviors over Topological Map

According to Lemma 1, for the connected nodes  $V_i$  and  $V_j$  in the topological map, there exists deterministic qualitative action

Action <sub>$i$</sub>  or action sequence  $\{\text{Action}_i\}$  to move the robot from  $V_i$  to  $V_j$ . So, we define qualitative state set  $QS$  as the set of topological nodes  $\{V_i, i = 1, 2, \dots, n\}$ , qualitative action set  $QA$  as the set of arcs in the topological map, and the elements of  $QA$  as the abstract actions such as “go-along-the-corridor,” “cross-the-door,” etc.

### Quantitative Behaviors over Grid-Based Map

According to Lemma 2, the subdivisions of global navigation in a local environment are mainly implemented by the reactive behaviors. In a grid world, the state set  $S$  is the set of all the grid cells and the action set  $A$  is  $\{\text{forward one step, backward one step, turn left } 45^\circ, \text{ and turn right } 45^\circ\}$ , which indicates that the movement of the robot is composed

of translating and turning. To make it easier for engineering implementation, we make a transition from grid cell state to range information state according to the distance data and design three kinds of behaviors: obstacle avoidance dynamically, turning to subgoal, and action coordination.

### Obstacle Avoidance

The distance information for obstacle avoidance is gained by detecting the environment with sonar sensors  $d_1, d_2, \dots, d_{18}$ , which is usually represented as  $D_1, D_2, \dots, D_6$  when grouped into six. A threshold value  $\delta = 0.15 \text{ m}$  is set as the safe distance. The maximum sensing range of the sonar sensor is  $L = 5 \text{ m}$ . Then the obstacle-avoidance behavior is trained using Q-learning within the range  $\delta \sim L$ , and the continuous states are partitioned by fuzzification. When the distance between the robot and the obstacle is less than  $\delta$ , this behavior is regarded as a failure and receives a penalty.

### Turning to Subgoal

As for the behavior of turning to subgoal, the state is transformed to  $(d_g, \theta_g)$ , where  $d_g$  is the distance between the robot and the subgoal and  $\theta_g \in (-\pi, \pi)$  is the angle between the orientation of the robot and the direction of the subgoal. The fuzzy logic is adopted to discretize the state  $(d_g, \theta_g)$ . The action set is the same as that of obstacle-avoidance behavior, and the immediate reward is defined as

$$\text{reward} = -\alpha \times d_g - \beta \times \theta_g, \quad (14)$$

where  $\alpha$  and  $\beta$  are parameters to be set. Equation (14) indicates that the nearer the robot is going toward the subgoal, the bigger is the reward.

## Action Coordination

The behavior of action coordination makes decisions between the outputs of the behaviors of obstacle avoidance and turning to subgoal, which works with the aim of getting to the subgoal safely and quickly. Because the subgoals are the source of attracting the robot and the obstacles are the source of repulsing the robot, we have the following definitions to measure the repulsing effect and the attracting effect.

The repulsing effect of the obstacles is

$$E_{\text{rep}} = k_1^* \max_i (v/d_i); \quad (15)$$

the attracting effect of the subgoal is

$$E_{\text{att}} = k_2^* v^* (\cos \phi_g + 1)/d_g; \quad (16)$$

the composition effect of the repulsing and attracting effects is

$$E = E_{\text{att}} - E_{\text{rep}}, \quad (17)$$

where  $k_1, k_2$  are coefficients,  $v$  is the velocity of the robot, and  $d_i$  is the sonar data from every directions. Suppose we execute the output of obstacle avoidance and turning to subgoal, and the variety of composition effects are  $\delta E_a$  and  $\delta E_g$ , respectively. We will make the decision

$$\pi(t) = w_a^* f(\Delta E_a) - w_g^* f(\Delta E_g) \begin{cases} > 0 & \text{obstacle avoidance} \\ < 0 & \text{turning to subgoal,} \end{cases} \quad (18)$$

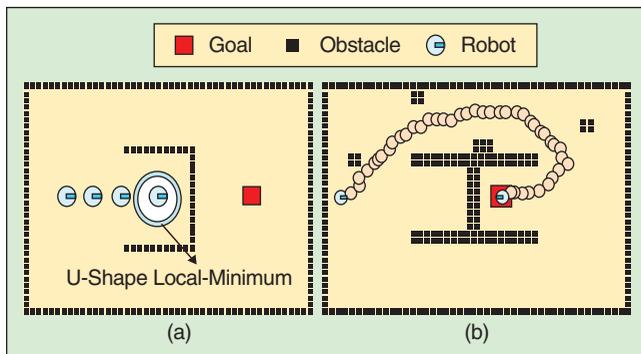
where  $w_g, w_g > 0$  are weights, and  $f(x)$  can be defined as a positive monotone function (e.g.,  $f(x) = e^x$ ).

## Experiments and Results

To demonstrate the feasibility and effectiveness of the proposed approach, a variety of experiments are carried out by computer simulation and real mobile robot ATU, respectively.

### Computer Simulation

A simulation environment is built up with the setting of  $600 \times 400$  (grid representation). The topological information can be acquired using the algorithm introduced earlier in this article. The parameter setting for the learning algorithms is as

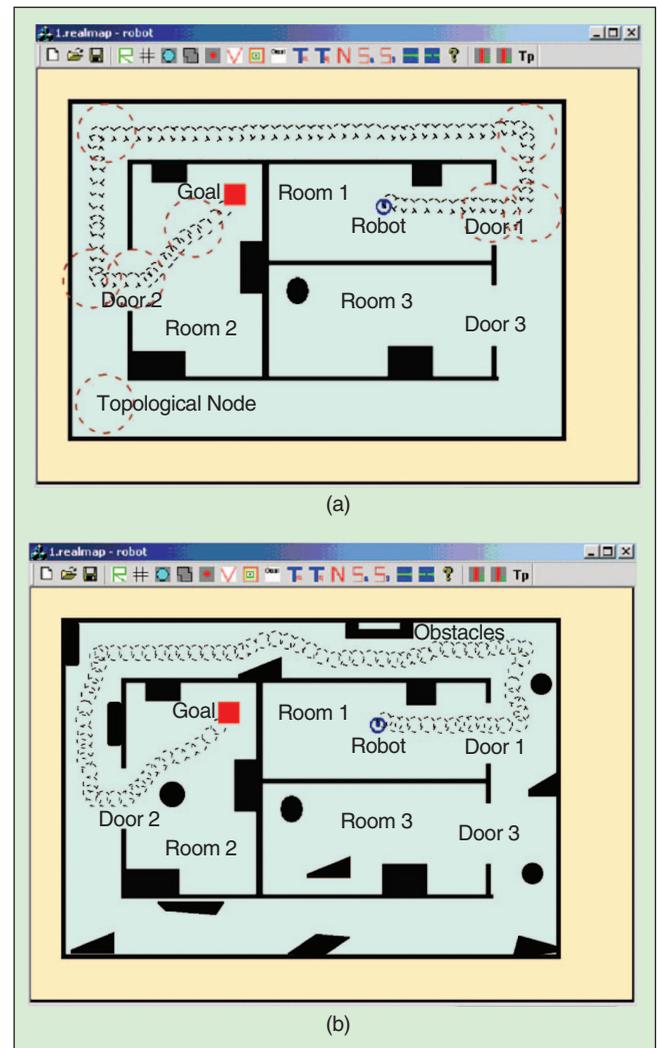


**Figure 11.** Navigating in a local environment: (a) local environment with U-shape and (b) robot performance.

follows: exploration policy  $\varepsilon$  – greedy,  $\varepsilon = 0.1$ , discounted factor  $\gamma = 0.9$ , learning rate  $\eta = 0.01$ , all the Q values are initialized as 0. Other configurations such as the membership function for environment recognition, and action coordination method, are carried out as discussed earlier.

Figure 11 shows the experimental results of navigation in a local environment with a U-shaped local minimum. Figure 11(b) presents the actual path to the goal after learning, which shows good performance to avoid being trapped in the U-shaped local minimum.

Figure 12 shows the experimental results of navigation in a large area that is a typical indoor environment. Figure 12(a) is the navigating result in a neat indoor environment, and the goal is set in a different room that is far from the initial state of the robot. Figure 12(b) is the same environment but with many obstacles. The results indicate that the robot navigates to the global goal without being trapped in a local minimum through learning. Hybrid control using HQL makes the mobile robot competent to navigate in a large unknown dynamic environment. The HQL algorithm also speeds up the



**Figure 12.** Navigating in a large-scale environment: (a) neat indoor environment and (b) clutter indoor environment with obstacles.

## Topological maps are abstract maps and can be built on top of the grid-based maps.

learning process because of its intrinsic structure that takes the most advantage of different levels.

Figure 12 shows that the planning in the high level leads to the nonoptimal local control, and all the results are suboptimal instead of optimal paths. However, the proper high-level planning is necessary to fulfill complex tasks especially in large-scale environments. Our aim is to design good learning algorithms and controllers to approximate the global optimal result.

### Real Experiments

The proposed methods are further tested on the real mobile robot ATU, whose hardware configuration has been presented

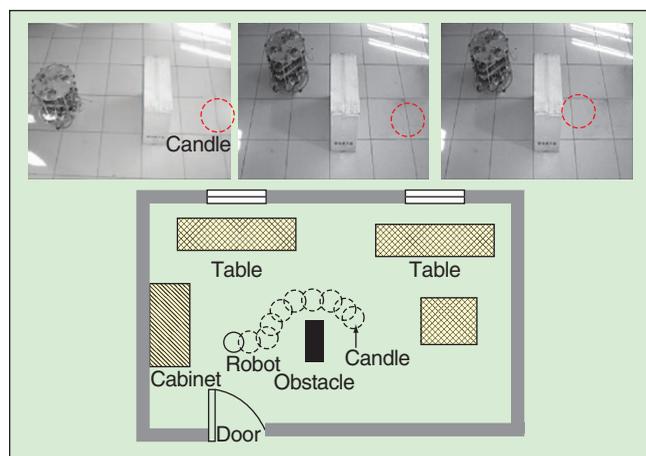


Figure 13. Navigating with obstacle avoidance in a room.

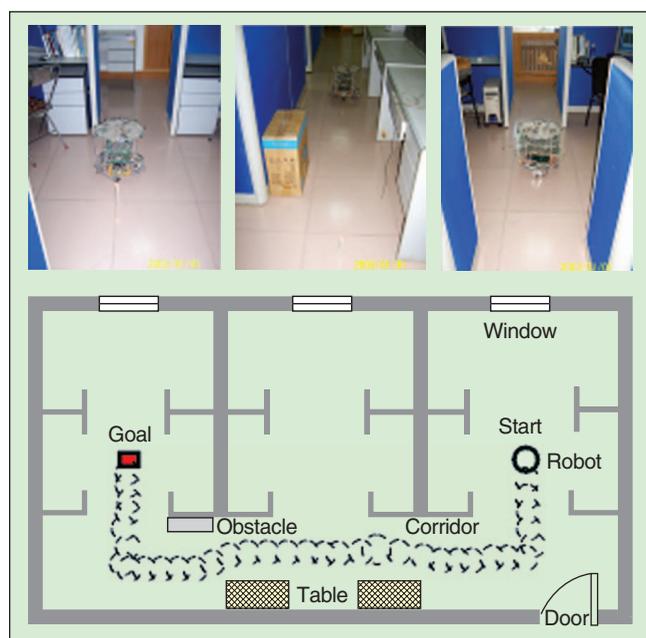


Figure 14. Navigating from room to room through a corridor.

earlier. The navigation environment is our laboratory, which consists of an experimental room and an office as shown in Figures 13 and 14, respectively.

Figure 13 shows the navigating results to find a candle in a local environment with obstacles. The sensors used to detect the candle are temperature-sensitive sensors whose detecting range is rather small (within 0.5 m). So, the robot has been set the rough direction of the candle. From the results, we can see that, after learning, the robot moves round the obstacle (a big paper box) and finds the candle successfully.

Figure 14 shows a more complex experiment, in which the mobile robot learns to reach the goal through a corridor with some obstacles and the goal is set in a different room far from the initial state of the robot. After a period of time of exploration and learning, the robot acquires the ability to navigate in this environment and moves to the goal through the corridor without bouncing into the obstacles.

### Conclusions

This article proposes a hybrid control approach using HQL. Hybrid control has shown to be a more robust and flexible solution to provide a suitable degree of reactivity and deliberation in a mobile robot. The HQL algorithm presents an integrated control scheme to combine the reactive and deliberate navigation, where different control levels are coordinated to gain effectiveness and efficiency. In this article, a grid-topological map-building method is introduced with online updating techniques, which provides a proper representation of the environment. The geometrical (quantitative) and topological (qualitative) navigations have been proved to be two of the effective approaches for robot navigation [12], [27]–[30], although both of them have advantages and disadvantages. So, the more practical solution is an integration of geometrical and topological navigation, which is the strategy that we take in this article. Finally, the simulation and real experiments based on an autonomous mobile robot ATU demonstrate the effectiveness of the proposed hybrid control architecture using HQL.

### Acknowledgments

The work is partially supported by a SRG project from City University of Hong Kong (7002114) and the Natural Science Foundation of China (60703083). The authors wish to thank Prof. Zonghai Chen, Dr. Guoqiang Hu, Dr. Guangming Zhou and Dr. Haibo Wang for helpful discussions. The authors also wish to thank the anonymous reviewers for helpful comments.

### Keywords

Mobile robot navigation, hybrid control, hierarchical Q-learning, grid-topological map.

### References

- [1] M. A. Salichs and L. Moreno, "Navigation of mobile robot: Open questions," *Robotica*, vol. 18, no. 3, pp. 227–234, 2000.
- [2] Y. K. Na and S. Y. Oh, "Hybrid control for autonomous mobile robot navigation using neural network based behavior modules and environment classification," *Autonom. Robots*, vol. 15, no. 2, pp. 193–206, 2003.

- [3] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robot," *IEEE Trans. Syst., Man, Cybern. B*, vol. 19, no. 5, pp. 1179–1187, 1989.
- [4] A. Elfes, "Sonar based real world mapping and navigation," *IEEE J. Robot. Automat.*, vol. RA-3, no. 3, pp. 249–265, 1987.
- [5] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Automat.*, vol. 2, no. 1, pp. 14–23, 1986.
- [6] A. M. Zhu, S. X. Yang, F. J. Wang, and G. S. Mittal, "A neuro-fuzzy controller for reactive navigation of a behavior-based mobile robot," *Lect. Notes Comput. Sci.*, vol. 3498, no. 5, pp. 259–264, 2005.
- [7] S. Thrun, "Probabilistic algorithms in robotics," *AI Mag.*, vol. 21, no. 4, pp. 93–109, 2000.
- [8] T. Kondo and K. Ito, "A reinforcement learning with evolutionary state recruitment strategy for autonomous mobile robots control," *Robot. Autom. Syst.*, vol. 46, no. 2, pp. 111–124, 2004.
- [9] C. L. Chen and Z. H. Chen, "Reinforcement learning for mobile robot: From reaction to deliberation," *J. Syst. Eng. Electron.*, vol. 16, no. 3, pp. 611–617, 2005.
- [10] E. Remolina and B. Kuipers, "Towards a general theory of topological maps," *Artif. Intell.*, vol. 152, no. 2, pp. 47–104, 2004.
- [11] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artif. Intell.*, vol. 99, no. 1, pp. 21–71, 1998.
- [12] P. Ranganathan, J. B. Hayet, M. Devy, S. Hutchinson, and F. Lerasle, "Topological navigation and qualitative localization for indoor environment using multi-sensory perception," *Robot. Autom. Syst.*, vol. 41, nos. 2–3, pp. 137–144, 2002.
- [13] C. L. Chen, D. Y. Dong, H. X. Li, and T. J. Tarn, "An Integrated Hierarchical Q-learning Method Based on Hybrid MDP," *IEEE Transactions on Systems, Man, and Cybernetics B*, under review, 2008.
- [14] J. C. H. Watkins and P. Dayan, "Q-Learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
- [15] R. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
- [16] M. Hallerdal and J. Hallamy, "Behavior selection on a mobile robot using W-learning," in *Proc. 7th Int. Conf. Simulation of Adaptive Behavior on From Animals to Animates*, Edinburgh, 2002, pp. 93–102.
- [17] M. Wiering and J. Schmidhuber, "HQ-Learning," *Adapt. Behav.*, vol. 6, no. 2, pp. 219–246, 1997.
- [18] A. G. Barto and S. Mahadevan, "Recent advances in hierarchical reinforcement learning," *Discrete Event Dyn. Syst.*, vol. 13, nos. 1–2, pp. 41–77, 2003.
- [19] R. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artif. Intell.*, vol. 112, no. 1, pp. 181–211, 1999.
- [20] R. Parr and S. Russell, "Reinforcement learning with hierarchies of machines," *Adv. Neural Inform. Process. Syst.*, vol. 10, no. 10, pp. 1043–1049, 1998.
- [21] T. G. Dietterich, "Hierarchical reinforcement learning with the Maxq value function decomposition," *J. Artif. Intell. Res.*, vol. 13, no. 1, pp. 227–303, 2000.
- [22] C. L. Chen, D. Y. Dong, and Z. H. Chen, "Grey reinforcement learning for incomplete information processing," *Lect. Notes Comput. Sci.*, vol. 3959, no. 1, pp. 399–407, 2006.
- [23] D. Berleant and B. Kuipers, "Qualitative and quantitative simulation: Bridging the gap," *Artif. Intell.*, vol. 95, no. 2, pp. 215–255, 1997.
- [24] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.
- [25] D. Pagac, E. M. Nebot, and H. Durrant-Whyte, "An evidential approach to map-building for autonomous vehicles," *IEEE Trans. Robot. Automat.*, vol. 14, no. 4, pp. 623–629, 1998.
- [26] C. C. Kung and J. Y. Su, "Affine Takagi-Sugeno fuzzy modelling algorithm by fuzzy c-regression models clustering with a novel cluster validity criterion," *IET Contr. Theor. Appl.*, vol. 1, no. 5, pp. 1255–1265, 2007.
- [27] M. Mata, J. M. Armingol, and A. Escalera, "Learning visual landmarks for mobile robot topological navigation," *Stud. Comput. Intell.*, vol. 7, no. 1, pp. 1–55, 2005.
- [28] B. Kuipers, "Cognitive maps for planetary rovers," *Autonom. Robots*, vol. 11, no. 3, pp. 325–331, 2001.
- [29] N. M. Sgouros, "Qualitative navigation for autonomous wheelchair robots in indoor environments," *Autonom. Robots*, vol. 12, no. 3, pp. 257–266, 2002.
- [30] D. Busquets, C. Sierra, and R. L. D. Mantaras, "A multi-agent approach to qualitative landmark-based navigation," *Autonom. Robots*, vol. 15, no. 2, pp. 129–154, 2003.

**Chunlin Chen** is a lecturer in the Department of Control and System Engineering, Nanjing University. He received the B.E. degree in automatic control and Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China in 2001 and 2006, respectively. His research interests include mobile robotics, machine learning, intelligent control, and quantum-inspired algorithm. He was listed in "Who Is Who in the World 2008," and now he serves as a committee member of the Committee of System Simulation Specialty, Chinese Automation Association. He is a Member of the IEEE.

**Han-Xiong Li** is an associate professor in the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong. He received the Ph.D. degree in electrical engineering from the University of Auckland, New Zealand, in 1997, M.E. degree in electrical engineering from Delft University of Technology in The Netherlands in 1991, and B.E. degree from the National University of Defence Technology in China in 1982. He is a "Chang Jiang Scholar," an honorary professor awarded by the Ministry of Education, China. He was awarded the distinguished young scholar fund by the China National Science Foundation in 2004. He has gained industrial experience as a senior process engineer from ASM, a leading supplier for semiconductor process equipment. His research interests include intelligent control and learning, process modeling and control, complex distributed parameter system, and electronics packaging process. He serves as an associate editor for *IEEE Transactions on Systems, Man, and Cybernetics—Part B*. He is a Senior Member of the IEEE.

**Daoyi Dong** is a postdoctoral fellow at the Key Laboratory of Systems and Control, AMSS, Chinese Academy of Sciences. He received the B.E. degree in automatic control and Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China in 2001 and 2006, respectively. His research interests include quantum control and quantum intelligent information processing. He has received the Chinese Academy of Sciences K. C. Wong Postdoctoral Fellowship in 2006 and the President Scholarship of the Chinese Academy of Sciences in 2004. He is a Member of the IEEE.

**Address for Correspondence:** Chunlin Chen, Department of Control and Systems Engineering, Nanjing University, Nanjing 210093, China; and the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong, China. E-mail: clchen@nju.edu.cn or chunlin.chen@ieee.org.

# Sensor-Based Robot Motion Planning

## A Tabu Search Approach

Planning robot motions in unknown environments has been an attractive research theme for many roboticists during the past two decades. The class of motion planners dealing with this kind of problem is known as online, sensor-based, local, a posteriori, real-time, or reactive motion planners.

Among the first works on online motion planning is Lumelsky's Bug algorithm presented for a point robot to move from a source point to a destination point, using touch sensing in a planar terrain populated with arbitrarily shaped obstacles [1]. Cox and Yap developed algorithms to navigate a rod to a destination position in planar polygonal terrains [2]. A survey on early online path planning works is provided in [3]. Another noteworthy approach for real-time planning is Khatib's potential fields (PFs) method, in which a point robot is directed by the forces in a field of potentials exerted by repulsive obstacles and the attractive goal [4].

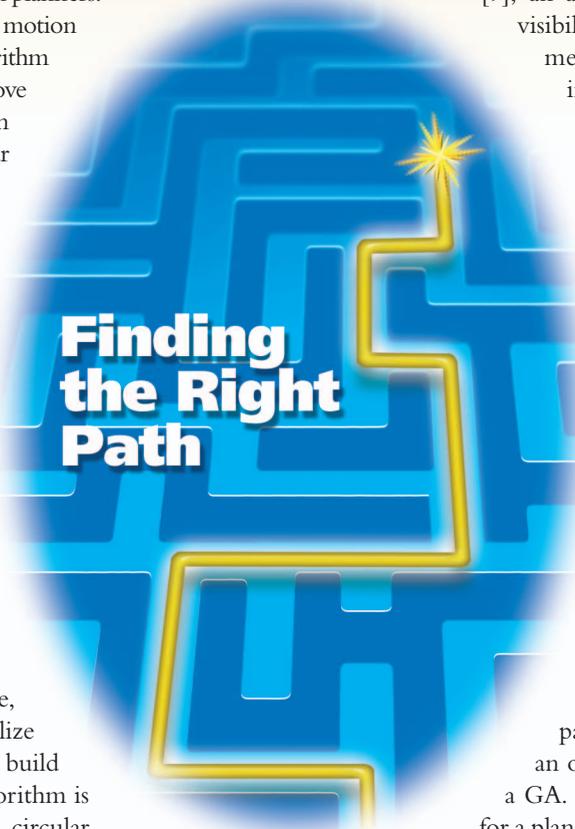
Aiming to take advantage of the properties of roadmaps constructed usually in the offline mode, some researchers have tried to utilize the distance transform approach to build them incrementally. In [5], an algorithm is proposed for the navigation of a circular robot in unknown terrains by iteratively visiting the vertices of the Voronoi diagram. Choset developed an incremental method to construct the hierarchical generalized Voronoi graph (HGVG) [6], which exploits some bridge edges (called GVG<sup>2</sup>) to maintain the connectivity of the GVG in high dimensions. Also, another method is proposed in

[7] for online motion planning through incremental construction of medial axis.

Other works such as [8] have tried to guide the motions of the robot along the edges of the visibility graph of a workspace of convex polygons in online mode. In [9], an algorithm is presented in which the visibility graph of the workspace is incrementally constructed by integrating the information of the paths traversed so far, and then, a globally optimal path is planned after the graph completion, as in offline mode.

In addition to the classic motion-planning approaches, other optimization methods generally known as heuristics have been increasingly employed for planning and optimizing robot motions. Heuristic algorithms do not guarantee to find a solution, but if they do, they are likely to do so much faster than the competing complete methods.

Some well-known metaheuristics such as genetic algorithms (GAs) and simulated annealing (SA) have found applications in robot motion planning. In [10], the path planning problem is expressed as an optimization problem and solved with a GA. It is done by building a path planner for a planar arm with 2 degrees of freedom, and then for a holonomic mobile robot. In [11], the path planning for vehicles is formulated as an optimization problem; the goal is to choose a path connecting initial and final points that crosses the least number of obstacles (with the eventual goal of zero crossings) in configuration space. A GA is devised in which the population is a set of paths. The SA approach is widely used in combination with the artificial potential field approach to escape from local minima, as in [12].



Finding  
the Right  
Path

©DIGITAL VISION

Digital Object Identifier 10.1109/MRA.2008.921543

BY ELLIPS MASEHIAN AND MOHAMMAD REZA AMIN-NASERI

## The Tabu Search

The tabu search (TS) method is another well-known metaheuristic technique first introduced by Fred Glover in 1989 [13]. TS is a powerful algorithmic approach that has been applied with great success to a large variety of difficult combinatorial optimization problem areas, such as assignment, scheduling, routing, TSP, etc.

TS has three phases: preliminary search, intensification, and diversification. During the first of these three steps, TS is similar to some other optimization methods in that whatever point  $x$  in the input space the robot is currently at, it evaluates the criterion function  $f(x)$  at all the neighbors  $N$  of  $x$  and finds the new point  $x'$  that is best in  $N$ . Repeating this idea creates the possibility of endlessly cycling back and forth between  $x$  and  $x'$  (a local minimum). To avoid this, TS differs from many other methods in that the robot moves to  $x'$  even if it is worse than  $x$ .

TS keeps track of performed moves and labels the recent moves as tabu moves, meaning that the search shall not revisit these points. The set of tabu moves is called tabu list (which is actually a push-down stack of  $s$  elements managed in a first-in/first-out manner), and its size is called tabu size. So that, for instance, once the move  $x \rightarrow x'$  has been made, the reverse move  $x' \rightarrow x$  is forbidden for at least the next  $s$  moves. Of course, when a tabu move has a cost lower than an aspiration level, it can be selected regardless of its tabuness. In addition to this tabu list, which is a recency-based short-term memory, we might introduce a frequency-based memory that operates on a much longer horizon (e.g., the last 50 iterations) and penalize the most frequently visited moves.

In the second (intensification) part of the search, it 1) starts with the best solution found so far (which is always stored throughout the entire algorithm), 2) clears the tabu list, and 3) proceeds as in the preliminary search for a specified number of moves. Finally, in the diversification phase, the tabu list is cleared again, and the  $s$  most frequent moves of the run so far are set to be tabu. Then, it chooses a random  $x$  to move to and proceeds as in the preliminary search phase for a specified number of iterations. The intensification phase focuses on the promising regions discovered during preliminary search, whereas the diversification phase forces exploration of completely new regions [14].

In this article, we introduce a TS-based robot motion-planning algorithm, which is in fact the first of its kind, as we did not find any prior instance in the literature. One reason might be that a straightforward way for defining tabu and non-tabu moves is via discretizing the C-space, for which some approaches like PFs have been introduced long ago. Nevertheless, because of our different approach in defining neighborhoods, employing TS has been possible and effective.

## The Motion Planner's Components

The new online motion planner presented in this article incorporates the robot's sensory data into the intelligence induced from the TS technique.

Before dealing with the major components of the motion planner, we define a move: a move is a motion from the current point  $x$  to another point inside the free C-space ( $C_{\text{free}}$ ) with a step size equal to the radius of the current point's locally

maximal disc (LMD)—which is the largest disc centered around  $x$  and completely contained in  $C_{\text{free}}$ —and a direction along one of its radial sensors (Figure 1).

The outline of the algorithm is as follows. Beginning from the start point, the robot performs a visibility scan to find visible obstacle vertices and decides to move toward an obstacle vertex it finds most promising according to a cost criterion. Upon making the move, backward directions are labeled as tabu and excluded from the set of next promising directions. The visibility scan and its ensuing operations are repeated for the new location, and the robot continues to navigate the environment until it sees the goal point. If at any stage the robot is trapped in a local minimum, it takes a random and relatively large step toward unexplored areas of the search space and continues its search in that area. The algorithm's main components are described later.

### Perception Component

The perception component is responsible for acquiring information from the environment and processing those data to determine the appropriate moves for the robot at each iteration. This is done by 1) performing a visibility scan and 2) detecting the visible obstacle vertices.

Upon arriving at a new point in the workspace, the robot first determines its distance to the surrounding obstacles by means of its radial range-finder sensor readings, which yields a list of candidate moves. Suppose that a circular mobile robot with radius  $R_{\text{rob}}$  and  $S$  range-sensors situated equidistantly on its perimeter is centered at point  $c$ . Each sensor projects a ray  $r_i$  ( $i = 1, \dots, S$ , counterclockwise) to find out its distance  $\rho_i$  from the nearest visible obstacle point  $\mathbf{x}_i$  along the  $i$ -th direction (Figure 1).

Taking the metric  $D(\mathbf{x}_i, \mathbf{x}_c)$  for the Euclidean distance of points  $\mathbf{x}_i$  and  $\mathbf{x}_c$ , we have  $\rho_i = D(\mathbf{x}_c, \mathbf{x}_i) - R_{\text{rob}}$ , where  $\mathbf{x}_c$  is the coordinate of the robot center's current position in the workspace. A representation of  $\rho_i$ 's versus ray angles is depicted in Figure 2(a).

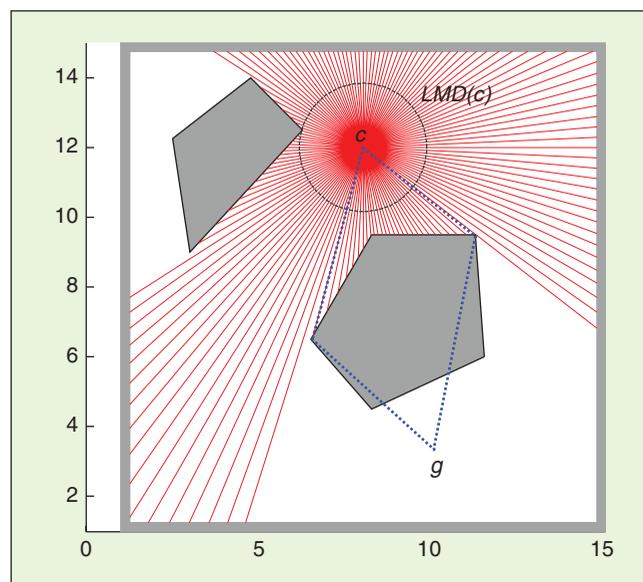


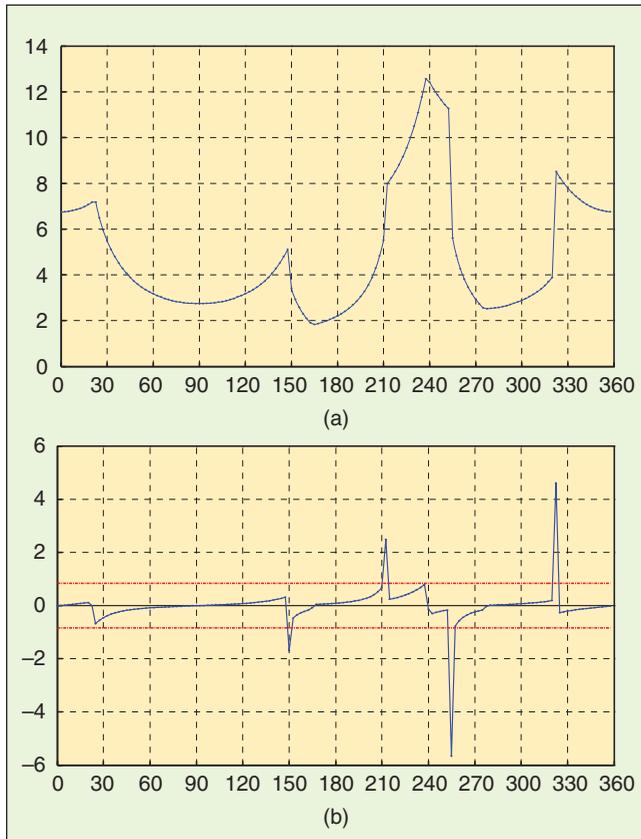
Figure 1. The visibility scan of the environment from the robot's location at point  $c$ .

In order for the robot to avoid getting trapped in obstacles' concave regions and bypass any blocking obstacle, it should move toward the tangent rays of the obstacle's boundaries. A ray  $r_i$  is tangent to an obstacle if in a neighborhood  $U$  of  $\mathbf{x}_i$  the interior of the obstacle lies entirely on a single side of the line  $r_i$ . Otherwise, the robot's motion toward the middle of the obstacle will lead to collision. This strategy stipulates the robot to distinguish the obstacle's outermost vertices, or in a broader sense (if the obstacles are not polygons), the regions adjacent to tangent rays, as viewed from the robot's vantage point.

For determining the tangent rays, a difference function is applied for successive adjacent rays to calculate the ray difference variables, as

$$\hat{\rho}_i = \rho_{i+1} - \rho_i. \quad (1)$$

Figure 2(b) shows the difference variables of the Figure 2(a). The sharp peaks (both positive and negative) imply abrupt and large differences in successive ray magnitudes, and so indicate the points where sweeping rays leave or meet a convex contour on the obstacle boundary. These peaks are detected by applying a notch filter to the plot. If no peaks are found, then the algorithm shifts to the diversification mode in which a random step is taken by the robot.



**Figure 2.** (a) The magnitudes of rays emitted from the robot's position in Figure 1, which was acquired by range sensors. (b) By applying a difference function, obstacle vertices are identified by sharp peaks. Insignificant peaks are omitted by using a notch filter (dashed horizontal lines at  $\pm 0.9$ ).

### Cost Evaluation Component

After determining visible obstacles' extreme vertices, the cost evaluation component associates a value to each tangent ray as a measure for the cost of reaching the goal via the direction of that ray. The criterion by which the cost is specified is a compound function. It incorporates two basic functions related to each ray: 1) distance function and 2) neighborhood function.

The distance function  $f_D(r_i)$  aims to estimate the length of the path connecting the robot center's current configuration to the goal configuration and is defined by

$$f_D(r_i) = \lambda_1 \cdot D(\mathbf{x}_r, \mathbf{x}_i) + \lambda_2 \cdot D(\mathbf{x}_i, \mathbf{x}_g). \quad (2)$$

The first part of (2) is deterministic and is calculated in the perception component. The second term is a heuristic estimation for the length of the free path connecting the  $r_i$ 's endpoint  $\mathbf{x}_i$  and the goal point  $\mathbf{x}_g$ . The weighted linear combination of these two terms (with  $\lambda_1$  and  $\lambda_2$  as weights) provides a heuristic criterion widely used in the  $A^*$  search technique. The thick dotted lines in Figure 1 show the distances involved in building the cost function: the lines originated from the robot's configuration show the tangent rays, i.e., the perceived obstacle vertices, and the lines to the point  $g$  present rough estimations for the distance of the vertices to the goal point, as in the  $A^*$  search.

The neighborhood function  $f_N(r_i)$  measures the degree of change in the magnitudes of neighboring rays occurring at  $r_i$  and is expressed as

$$f_N(r_i) = \alpha \cdot \max\{\hat{\rho}_i, \hat{\rho}_{i-1}\}, \quad (3)$$

where  $\alpha$  is a tuning parameter. A large value of  $f_N(r_i)$  implies that the obstacle (vertex) adjacent to  $r_i$  has a relatively large free space behind it and will possibly lead the robot to a key position in the configuration space, hence offering a better maneuverability for it. Small amounts of  $f_N(r_i)$  indicate cramped areas, narrow passages, or obstacle borders, which generally have lesser priority for navigation.

The overall cost evaluation criterion  $C(r_i)$  is minimizing a blend of the distance and neighborhood functions according to

$$C(r_i) = P_i \cdot f_D(r_i)^\beta \cdot f_N(r_i)^{-\gamma}, \quad (4)$$

in which  $\beta$  and  $\gamma$  are scalars and  $P_i$  is defined as

$$P_i = \begin{cases} e & \text{if } r_i \text{ has the direction of the last move} \\ \nu & \text{if } r_i \text{ points to a visited vertex} \\ t & \text{if } r_i \text{ is a tabu direction.} \end{cases} \quad (5)$$

Through its reducing effect, the parameter  $e < 1$  encourages the robot to continue its navigation along a direction selected in the past few iterations and to be not diverted frequently by every new vertex that appears in its scope. The parameter  $\nu$  increases the cost of a ray pointing to a previously visited vertex, whereas the parameter  $t$  imposes a penalty for directions that are designated as tabu ones. The suggested values for these parameters are shown in Table 3.

After evaluating all rays, the motion planner is able to select the most promising goal-oriented direction to move along, which corresponds to the ray associated with the lowest cost.

Note that since the neighborhood function  $f_N(r_i)$  has a negative exponent in (4), its large values (corresponding to tangent rays) reduce the overall cost dramatically. It follows that the probability of selecting obstacle vertices as next promising destinations is much higher than that of the ordinary rays (which point to obstacle borders), and, therefore, the robot is naturally being attracted to vertices. On the other hand, the distance function  $f_D(r_i)$  in (4) increases the probability of selecting near-to-goal destinations through its positive exponent. In other words, the designed cost evaluation function leads to locally optimal (i.e., shortest) navigations, just as the visibility graph does in offline mode, and, thus, the robot's performance improves significantly.

### Aspiration and Desperation Levels

The aspiration level is a level set to accept a very good move, even if it is tabu. This is an established concept in TS, proposed by Glover. Now, we make the TS metaheuristic more flexible and powerful by introducing a new concept called desperation level.

The desperation level is a level (of cost) beyond which a nontabu move having higher cost values (for minimization problems) or lower cost values (for maximization problems) is rejected and included in the tabu list. It is somehow a counterpart and complementary concept for the aspiration level. The relation of these levels with regard to the tabu or nontabu moves is depicted in Figure 3. The gray sections of the diagram (i.e., II and IV) represent unacceptable moves, since they are either tabu—with costs not better than the aspiration level—or nontabu, but with costs higher than the desperation level.

In our planning context, it frequently happens that there are no nearby nontabu obstacle vertices. Instead, there are some remote vertices with high costs beyond the desperation level. Excluding such vertices from the moves list will make the list empty and limit the search space, which in turn will activate the diversification component and will cause the robot to take a random step toward unexplored areas of the space.

### The Short-Term Tabu List

The notion of a tabu list is critical and fundamental to this approach. The attribute by which we set up tabu lists is the direction of the rays emanated from the robot.

In each iteration, tabu moves are identified based on the robot's direction. A tabu envelope (TE) variable is specified to set the range of tabu directions. It is laid out symmetrically around the reverse of the robot's direction, covering all rays within a  $\pm TE/2$  deviation. For instance, in Figure 4, the robot's initial direction is  $50^\circ$ . By setting  $TE = 90^\circ$ , all the directions included in the area  $(\pi + 50^\circ) \pm 45^\circ$  (i.e.,  $185\text{--}275^\circ$ ) are characterized as tabu moves (the gray sector).

For setting up the short-term tabu list, if its size (STLS) is set to  $k$ , then the tabu directions of the last  $k$  iterations are appended to form the total set of tabu moves. We found  $k = 2$  to be the best size, although  $k = 1$  is also possible, but it leads to more fluctuations in the robot's motion.

### The Long-Term Tabu List

Aside from the short-term tabu list discussed earlier, we set a long-term tabu list by keeping a record of the already visited (i.e., almost touched) vertices. If the robot happens to head for a visited vertex, then since it has been at that location during earlier iterations, it should avoid the point and concentrate on other vertices in view. This is done by a simple checking of the long-term tabu list, with a size of LTLS. The long-term tabu list may also contain a set of nonvertex points that are proven to be ineffective and misleading in directing the robot toward the goal.

### Diversification Component

This component is evoked when there are no admissible nontabu directions. This situation occurs when 1) all the vertices in scope have been previously visited and marked as tabu (i.e., are in tabu list), 2) all nontabu moves have a cost value higher than the desperation level, or 3) the robot is entrapped in a dead end.

The robot will then take a large step with a random direction selected from among rays with big magnitudes (compare with the concept of the diversification phase discussed earlier). The short-term tabu list is cleared after this step, but the long-term tabu list is retained. This action will most likely guide the

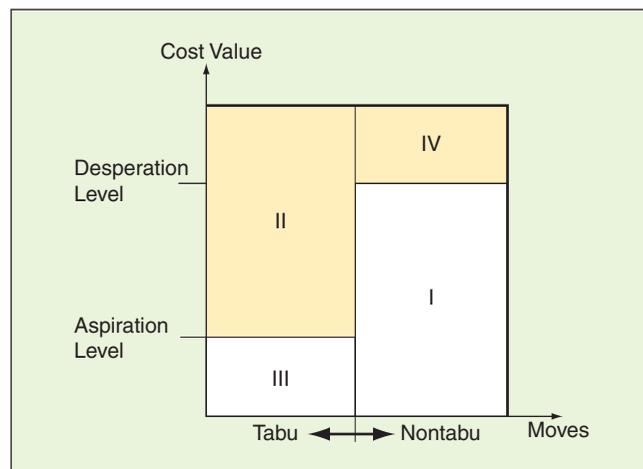


Figure 3. The aspiration and desperation levels for a minimization problem. Region IV represents moves considered unacceptable, although nontabu.

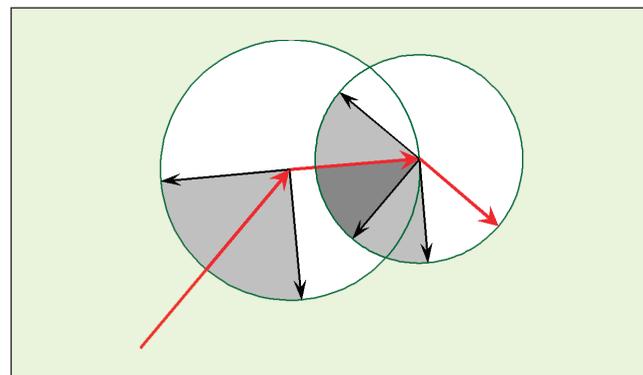


Figure 4. A short list of tabu moves is constructed by appending the tabu envelopes of the last two iterations.

robot toward new and unexplored areas of the searching space. Consequently, all the elements of the short-term tabu list, as well as some older elements of the long-term tabu list, will be eliminated.

In fact, the diversification component provides the planner's probabilistic-completeness property. That is, given sufficient time, the algorithm will eventually reach the goal if there is a valid path. The random steps guarantee that the robot will explore all areas of the workspace.

### Safety Component

When the robot approaches an obstacle border closer than a preset safety radius  $R_s$ , it should take a reflective step away from the obstacle to maintain its safety. This is similar to the behavior of a light beam when reflected from a surface. The reflective step in Figure 5(a) directs the robot toward a safer location via a relatively large and outward movement. It also effectively helps the robot to turn around obstacle vertices and sharp corners [Figure 5(b)].

The length of the reflective step is set to a few times the radius of the LMD, and its direction is determined by a summation of the robot's direction vector and the obstacle border's normal vector.

### Goal Connection Component

If the goal point lies within the sights of the robot, then through a goal connecting operation, the robot's location is connected to the goal point via a straight line. The robot then has to follow that line and terminate its search.

### Algorithm Steps

By integrating the aforementioned components in a single architecture, the TS-based online motion planner follows these steps to produce a goal-driven trajectory:

- Step 1:** The goal connection component checks whether the goal is visible: if it is visible, the current point is connected directly to the goal, and the algorithm is terminated. Otherwise, go to Step 2.
- Step 2:** The robot activates the perception component, including the visibility scan and the discovery of surrounding obstacles' vertices.

**Step 3:** The cost evaluation component evaluates the cost of all directions based on a criterion.

**Step 4:** Update the short-term tabu list (based on the tabu envelopes of the last two moves) as well as the long-term tabu list (based on visited obstacle vertices or misleading configurations).

**Step 5:** If an obstacle border is closer than the safety radius, the robot takes a reflective step away from the obstacle border and goes to Step 1. Otherwise, go to Step 6.

**Step 6:** Construct a list of nontabu directions by excluding tabu moves from the set of all moves and considering the aspiration and desperation levels. If the nontabu list is not empty, select a direction with the lowest cost among the nontabu moves. Otherwise, activate the diversification component and take a large step along a random direction. Go to Step 1.

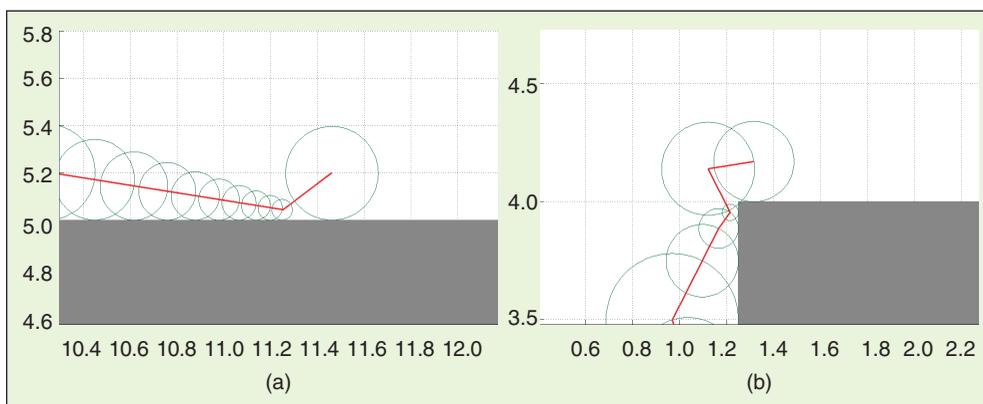
### Experimentation

The algorithm was run for several problems ranging from simple convex to highly concave polygons and mazes and succeeded in performing effectively. Some of the simulations are shown in Figure 6(a)–(h). The running times were within a few seconds using a 2.16 GHz Intel Duo processor.

The robot navigates faster in sparse and uncluttered areas and more cautiously in cluttered and near-to-obstacle regions. The sharp angles in the trajectories are due to the reflective steps. The effect of the diversification component can be seen in Figure 6(h), where the upper-left large step is the random move made after backtracking, hoping for exploring new areas.

To test the efficiency of the proposed method and compare it with other approaches, we designed and solved a number of test problems. The results are shown in Table 1. Compared with global optimum solutions, the paths produced by the TS-based planner had 9.25% average error. Large errors generally occurred in maze-like problems. Because of the vertex attraction fact imposed by the adopted cost function, the planner has a tendency to follow short paths inherited from the visibility graph roadmap (which produces the shortest path in offline mode). Therefore, as an online method, the path quality is quite satisfactory, especially when compared with other offline methods like A\* grid search, PF, or Voronoi diagrams methods (Table 1 and Figure 7). Because grid-based methods (like PF and A\*) examine the neighboring cells of grid points, the resulting path is rough and can only have vertical, horizontal, and diagonal local directions.

Since online methods acquire their knowledge of environment by sensors and plan their path locally, it would be incorrect to compare the processing times of offline and online methods. However, compared with the average



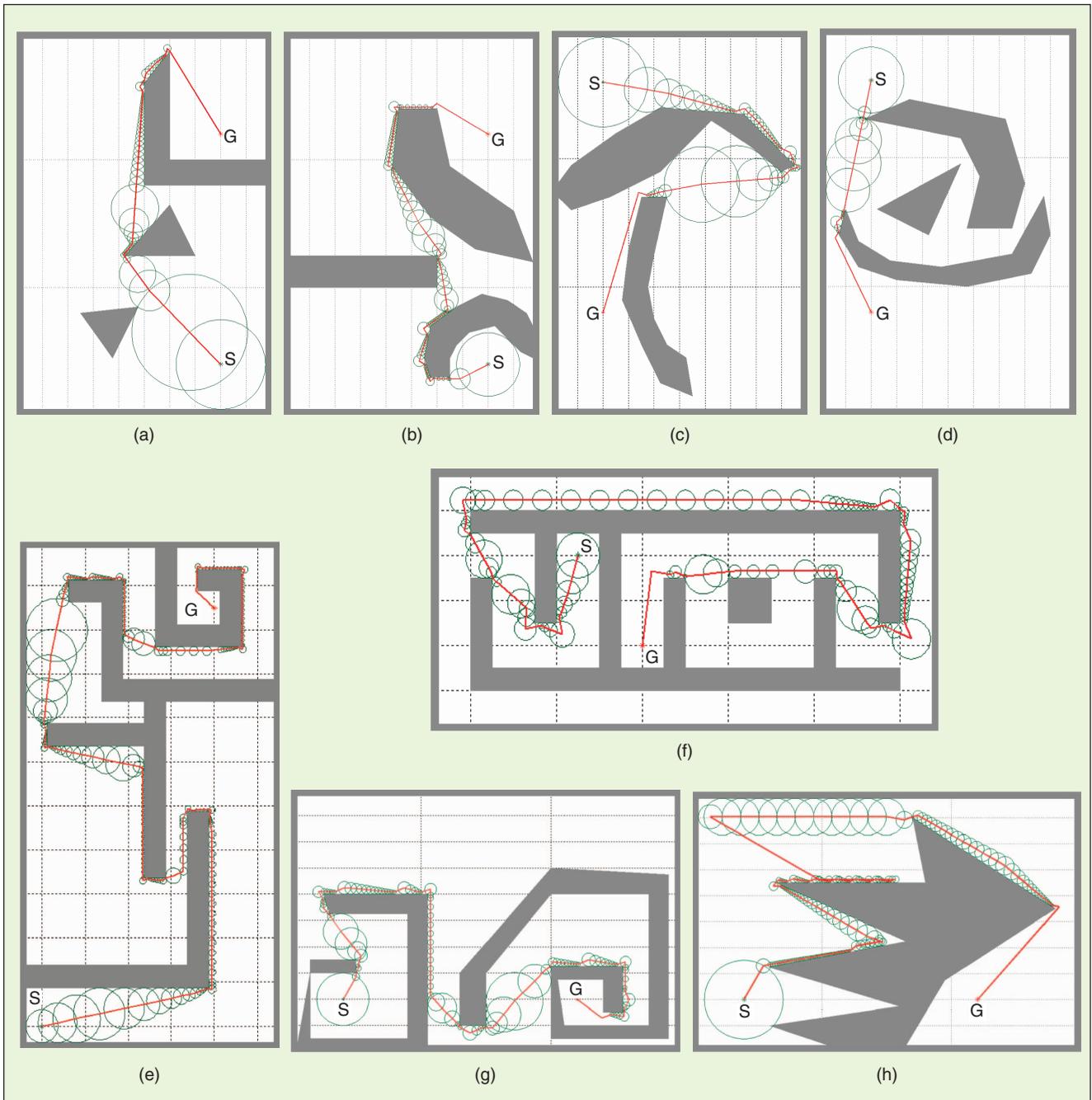
**Figure 5.** (a) The reflective step guarantees that the robot will keep a safe distance from the obstacles. (b) The reflective step enables vertex surmounting.

processing time of 7.46 s for the A\* search and 3.08 s for the PF method (calculated for the 15 problems), the TS-based algorithm performed convincingly well.

We also compared the performance of the proposed new planner with that of an online distance transform method, the incremental construction of generalized Voronoi diagram (GVD). This method builds the GVD incrementally using the information acquired by its sensors [6], [7]. Overall, the path lengths of the TS-based method are shorter (about 25% less in our experiments) than that of the GVD method. This is due to the nature of the Voronoi diagram that keeps the maximum clearance from the obstacles, whereas the TS-based planner is

*The robot navigates faster in sparse and uncluttered areas and more cautiously in cluttered and near-to-obstacle regions.*

attracted to obstacle vertices and thus emulates the visibility graph, which provides the shortest path. Problem 13 in Table 1 was solved by the GVD online method in 24.3 s through 243 iterations and with 24 sensors and a path length of 75.06



**Figure 6.** Some experiments: (a)–(d) convex and concave obstacles, (e)–(g) maze-like obstacles, (h) a random move is made following the activation of the diversification component.

[Figure 7(c)]. The TS-based algorithm's solution is shown in Table 1 and Figure 6(e). Other problems were also compared and gave more or less the same results.

To evaluate the performance of the TS-based planner against one more online motion planner, we selected the sensor-based rapidly exploring random tree (SRT) method [15], which is an online version of LaValle's rapidly exploring

random tree (RRT) method [16]. The SRT-Star method was run for the 15 benchmark problems, and the results are summarized in Table 2.

As its name implies, the SRT method builds a rooted tree from the start point, and at each iteration, by generating neighboring nodes, it extends its branches randomly but toward previously unexplored areas of the C-space. When encountering

**Table 1. A comparison of the path lengths generated by different methods.**

Problem	Number of Convex Obstacles	Work-Space Size	TS-Based Online Planner			Path Lengths by Offline Methods					TS-Based Path Length Error %
			Number of Sensors	Number of Iterations	CPU Time (s)	Path Length	A* Search <sup>ab</sup>	Potential Fields <sup>ac</sup>	Voronoi Diagram <sup>d</sup>	Visibility Graph <sup>e</sup>	
1	1	[10 × 10]	24	7	0.24	10.42	10.61	13.31	14.07	10.33	0.87
2	2	[10 × 10]	24	31	1.37	14.08	14.18	21.74	19.90	13.72	2.62
3	4	[10 × 10]	36	60	4.11	18.85	21.53	24.67	27.71	18.01	4.66
4	5	[15 × 10]	16	36	1.65	14.94	18.27	20.85	21.92	13.92	7.33
5	5	[15 × 10]	16	47	4.09	21.49	22.26	28.77	33.01	19.12	12.39
6	6	[10 × 10]	18	79	7.00	22.64	23.21	27.22	26.82	18.96	19.41
7	7	[15 × 10]	24	40	2.54	17.77	19.53	23.79	26.66	16.95	4.84
8	8	[10 × 10]	24	41	4.20	18.09	18.06	18.81	20.37	15.38	17.62
9	8	[15 × 10]	36	46	5.31	15.74	17.75	21.95	20.47	15.18	3.69
10	7	[10 × 10]	36	83	9.79	25.54	26.25	34.51	31.84	23.73	7.63
11	11	[15 × 10]	20	53	7.36	19.24	18.48	21.94	24.35	16.36	17.60
12	16	[15 × 10]	36	29	10.10	14.07	13.55	18.58	17.40	13.93	1.01
13	11	[13 × 24]	20	183	14.06	64.67	54.50	64.90	74.61	53.60	20.65
14	12	[9 × 10]	24	52	4.62	13.04	19.11	18.46	16.31	12.26	6.36
15	16	[13 × 24]	40	97	23.86	32.43	31.71	36.31	39.46	28.90	12.20
Average				59	6.68	21.53	21.93	26.39	27.66	19.36	9.25

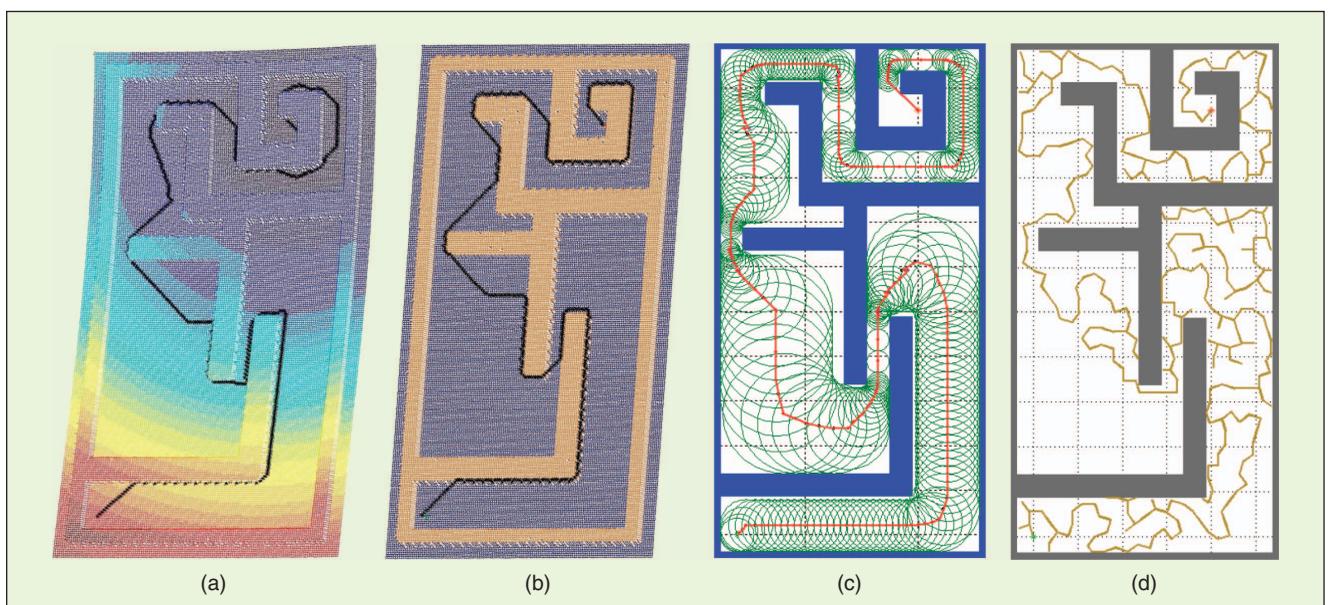
<sup>a</sup>After graduating the workspace with 1/10 of the unit length.

<sup>b</sup>With a best-first search strategy and Euclidean distance heuristic.

<sup>c</sup>With filling-up local minima.

<sup>d</sup>Searched by the Dijkstra's method.

<sup>e</sup>Optimal solution.



**Figure 7.** The problem 13 in Table 1 is solved by (a) PF approach in 3.1 s, (b) A\* search in 41.5 s, (c) online distance transform (GVD) builder in 24.3 s, and (d) sensor-based RRT planner in 14.4 s.

a dead end, it backtracks to its parent node and repeats the procedure, until filling the whole C-space. SRT is essentially developed for workspace exploration, and not for goal reaching. So, we modified it slightly to terminate the search if the goal point is viewed by sensors. A sample output for the problem 13 is illustrated in Figure 7(d).

We observed that SRT fails to reach the goal at some runs, despite calibrating its parameters carefully. This is because it is a resolution- and probabilistic-complete method. Since the time and path length vary greatly because of the SRT's highly stochastic nature, we did several experiments for providing sufficient valid data for our comparison. The failure rate (%), average number of iterations, as well as the mean and standard deviation of the results are also included in Table 2. The last two columns show relative time and path lengths of the TS-based and SRT planners. In both aspects, values less than 1 show the TS-based planner's fine performance.

### Parameter Setting and Tuning

As a metaheuristic approach, the TS technique requires proper setting of its parameters.

A key parameter is the number of sensors: it is assumed that the robot has a circular perimeter equipped with a sensor ring. Practically, the number of sensors is less than 30 or 40. However, for our theoretical investigations and simulations, we selected different sensor numbers such as 12, 18, 24, 36, 40, 60, 72, 90, 120, 180, 240, and 360. The larger is the number of sensors, the more exact is the perception of the environment. On the other hand, large sensor numbers require high computational time and memory.

In the case that the mobile robot (or its sensor) has the ability to rotate about a central axis, then the number of radial sensors becomes less critical, since a disc-shaped robot, for instance, can multiply its environmental perception by  $i$  times

*The tabu search method, which is a well-known metaheuristic technique for solving difficult combinatorial optimization problems, is being applied in robot motion planning for the first time.*

( $i = 2, 3, \dots$ ) via rotating about its center with increments of  $2\pi/(S \times i)$  degrees. A sensitivity analysis for the number of sensors can help users determine the proper degrees of rotation, as well as the hardware requirements for the robot's successful motion planning and navigation. The SRT method proved to be less sensitive to the number of sensors.

A summary of the parameters used in the algorithm, together with our suggested values for them, is presented in Table 3. These values are set through extensive tests and evaluations.

The exploration procedure would be successful if these parameters are set properly suitable for the workspace under navigation. To reduce the risk of improper parameter setting for unknown environments, and to make the planner even more intelligent, we propose a number of rules and guidelines associated with each parameter, such that as the exploration goes on, the robot can adapt itself to different situations by learning more and more about the environment. Thus, it can adopt strategies for reaching the goal, avoiding local minima, and tuning its parameters automatically. We have discussed the effects of tuning each parameter in Table 3. This automatic adaptation makes the TS-based planner very efficient and powerful.

**Table 2. Results of solving the test problems with SRT.**

Problem	Average Number of Iterations	Average Number of Nodes	CPU Time		Path Length		Failure Rate (%)	TS Time Versus SRT	TS Path Versus SRT
			Average	Standard Deviation	Average	Standard Deviation			
1	28.6	23.7	0.94	0.38	32.92	9.74	0.0	0.25	0.32
2	34.9	27.3	1.20	0.23	37.72	5.63	8.3	1.15	0.37
3	67.8	58.5	3.43	0.53	55.74	6.67	9.1	1.20	0.34
4	119.8	84.8	5.46	2.73	81.60	23.18	0.0	0.27	0.18
5	127.4	94.7	7.21	2.32	90.48	19.18	23.1	0.57	0.24
6	78.5	65.0	3.97	0.84	61.94	8.10	9.1	1.76	0.37
7	147.2	98.8	8.84	3.28	93.84	22.16	9.1	0.29	0.19
8	81.9	58.1	4.49	1.04	54.35	9.02	37.5	0.94	0.33
9	121.8	86.8	10.02	5.48	82.74	24.22	8.3	0.53	0.19
10	71.3	60.6	5.22	0.99	56.52	6.94	50.0	1.88	0.45
11	109.0	77.2	8.48	2.79	72.41	16.87	16.6	0.87	0.27
12	105.4	77.9	18.74	7.89	74.73	23.82	0.0	0.54	0.19
13	254.6	203.7	18.39	8.37	183.60	26.21	65.5	0.76	0.35
14	57.9	45.2	3.28	1.51	41.36	15.16	36.8	1.41	0.32
15	249.7	177.3	38.86	17.97	170.60	45.92	7.1	0.61	0.19
Total average			9.24	3.76	79.37	17.52	18.7	0.72	0.27

**The aspiration level is a level of cost set to accept very good moves even if they are tabu, whereas the desperation level is a level of cost beyond which nontabu moves are rejected.**

### Discussion

Two very important issues of a path planning algorithm are its time complexity and completeness.

For determining the time complexity of the algorithm, we should first estimate the number of iterations required to accomplish the path planning task by establishing an upper bound for it. The worst condition occurs when the robot takes smallest possible steps all the time. This happens if the robot moves along the obstacle borders with a minimum clearance, determined by the value of the safety radius, while visiting all

the obstacles in the workspace. Suppose that there are  $m$  disjoint obstacles that are arranged in a regular array. Taking the overall number of obstacle vertices as  $n$ , the total number of obstacle edges would also be  $n$ , and the total border length would be finite times  $n$ , plus the distances of interobstacle traversals, which is finite times  $m$ . Since  $m < n$ , the upper bound of the maximum number of iterations is in  $O(n)$ . Even if this length is navigated with the smallest possible step size ( $R_s$ ), because of the constant number of computations in each iteration, the time complexity of the algorithm would still be in  $O(n)$ .

Unlike distance transform planners (which explore the medial axis of the C-space thoroughly), the TS-based planner does not benefit from a similar connected graph, and so it is not complete. However, because of the large diversifying steps taken on a random basis, the robot can explore all unexplored areas given sufficient time, and so is probabilistically complete, which means it is guaranteed to reach the goal within a long time.

### Conclusions

The new online motion planner developed in this article is based on the tabu search metaheuristic. Various components

**Table 3. Parameters used in the motion planner.**

Symbol	Description	Suggested Range	Conditions for an Increase	Conditions for a Decrease
TE	Tabu envelope	$[\pi/2, \pi]$	The workspace is uncluttered and straight motions would work well	There are many narrow passageways
STLS	Short tabu list size	1 or 2	Must not exceed 2	Cannot find enough vertices
LTLS	Long tabu list size	[5, 20], integer	The number of vertices is large, or the goal is far from current location	Otherwise
AL	Aspiration level	[0.2, 0.8]	Frequent turn-backs are required	A look-forward approach is selected
DL	Desperation level	[5, 15]	Exploring a semiclosed area	Escaping an area by random moves
$\lambda_1, \lambda_2$	Distance function coefficients	(0, 1]	$\lambda_1/\lambda_2 > 1$ : more predictive than realistic, for conventional areas	$\lambda_1/\lambda_2 < 1$ : more realistic than predictive, for unpredictable areas
$\alpha$	Neighborhood function coefficient	(0, 1]	Exploring vast free areas located behind obstacles	Focusing on reaching the goal via narrow openings and passages
$\beta$	Total cost function coefficient	(0, 3]	Moving away from the goal before approaching it (e.g., when getting out of a cul-de-sac)	A more goal-oriented, "greedy" action is required
$\gamma$	Total cost function coefficient	(0, 3]	Exploring vast free areas located behind obstacles	Focusing on reaching the goal via narrow openings and passages
e	Straight move incentive	[0.35, 1]	Reacting to (and approach) new obstacles	The trajectory fluctuates much and is not smooth
v	Vertex revisiting penalty	[6, 10]	Cannot find the goal and want to take more random steps	Reexploring a revisited area
t	Tabu direction penalty	[4, 8]	Exploring unvisited areas	Intensifying exploring an area
$R_s$	Safety radius	$[1.1, 1.5] \times R_{rob}$	More safety is required	Otherwise
$R_f$	Reflective step length	$[4, 8] \times R_s$	Taking more risk of collision	A wall-following behavior is required

## Considering the online and sensor-based nature of the presented model, it can be applied to dynamic environments as well.

of the classic TS have been remodeled and integrated in a single algorithm to craft a motion planner capable of solving varieties of exploration and goal-finding problems. By employing different combinations of a number of parameters, the planner can react intelligently and promptly to the new situations it faces during the robotic navigation. The presented explanations on the parameters' definitions and attributes can help researchers in applying this algorithm to their real-world experiments and applications.

The newly defined concept of desperation level also enriches the still-evolving TS discipline, and together with the aspiration level and the diversification step, it enables the robot particularly to escape from local minima. Numerous experiments and comparisons with offline and online methods showed the algorithm's success and efficiency in coping with different problems, from simple polygons to highly concave obstacles.

Considering the online and sensor-based nature of the presented model, it is believed that it can be applied to dynamic environments (with moving obstacles) as well. In that case, the neighborhood and distance functions must be modified to accommodate some predictive information about the velocity vectors of each obstacle.

### Keywords

Robot motion planning, sensor-based navigation, tabu search metaheuristic.

### References

- [1] V. J. Lumelsky and A. A. Stepanov, "Dynamic path planning for a mobile automation with limited information on the environment," *IEEE Trans. Automat. Contr.*, vol. 31, no. 11, pp. 1058–1063, 1986.
- [2] J. Cox and C. K. Yap, "On-line motion planning: moving a planar arm by probing an unknown environment," Courant Institute of Mathematical Sciences, New York Univ., New York, Tech. Rep., July 1988.
- [3] N. S. V. Rao, S. Kareti, W. Shi, and S. S. Iyengar, "Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms," Oakridge National Lab., Tech. Rep. ORNL/TM-12410, 1993.
- [4] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.
- [5] N. S. V. Rao, N. Stolfus, and S. S. Iyengar, "A 'retraction' method for learned navigation in unknown terrains for a circular robot," *IEEE Trans. Robot. Automat.*, vol. 7, no. 5, pp. 699–707, 1991.
- [6] H. Choset, S. Walker, K. Eiamsa-Ard, and J. Burdick, "Sensor-based exploration: Incremental construction of the hierarchical generalized Voronoi graph," *Int. J. Robot. Res.*, vol. 19, no. 2, pp. 126–148, 2000.
- [7] E. Masehian, M. R. Amin-Naseri, and S. Esmailzadeh Khadem, "Online motion planning using incremental construction of medial axis," in *Proc. IEEE Int. Conf. Robotics and Automation*, Sept. 2003, vol. 3, pp. 2928–2933.
- [8] V. Krishnaswamy and W. S. Newman, "Online motion planning using critical point graphs in two-dimensional configuration space," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 1992, vol. 3, pp. 2334–2339.
- [9] J. B. Oommen, S. S. Iyengar, N. S. V. Rao, and R. L. Kashyap, "Robot navigation in unknown terrains using visibility graphs—Part I: The disjoint convex obstacle case," *IEEE J. Robot. Automat.*, vol. 3, pp. 672–681, Dec. 1987.
- [10] J. M. Ahuactzin, T. El-Ghazali, P. Bessiere, and E. Mazer, "Using genetic algorithms for robot motion planning," in *Proc. Workshop of Geometric Reasoning for Perception and Action*, Sept. 16–17, 1991, pp. 84–93.
- [11] C. Eldershaw and S. Cameron, "Real-world applications: Motion planning using GAs," in *Proc. Genetic & Evolutionary Computation Conf.*, Orlando, FL, July 1999, p. 1776.
- [12] F. Janabi-Sharifi and D. Vinke, "Integration of the artificial potential field approach with simulated annealing for robot path planning," in *Proc. IEEE Int. Symp. Intell. Contr.*, Aug. 1993, pp. 536–541.
- [13] F. Glover, "Tabu search—Part I," *ORSA J. Comp.*, vol. 1, no. 3, pp. 190–206, 1989.
- [14] F. Glover and M. Laguna, *Tabu Search*. Norwell, MA: Kluwer Academic, 1997.
- [15] G. Oriolo, M. Vendittelli, L. Freda, and G. Troso, "The SRT method: Randomized strategies for exploration," in *Proc. IEEE Int. Conf. Robotics and Automation*, New Orleans, LA, Apr. 2004, pp. 4688–4694.
- [16] S. M. LaValle, "Rapidly-exploring random trees; a new tool for path planning," Comp. Sci. Dept., Iowa State Univ., Tech. Rep. TR 98-11, Oct. 1998.

**Ellips Masehian** is an assistant professor at the Faculty of Engineering, Tarbiat Modares University, Tehran, Iran. He received the B.S. and M.S. degrees in industrial engineering, both from Iran University of Science and Technology, Tehran, with honors, and a Ph.D. degree from Tarbiat Modares University. His research is focused on the application of heuristic and intelligent methods to single and multiple robot motion planning problems.

**Mohammad Reza Amin-Naseri** is currently an associate professor of the industrial engineering department at Tarbiat Modares University, Tehran, Iran. He received his B.S. degree in chemical and petrochemical engineering from Amir-Kabir University of Technology (Tehran Polytechnic), an M.S. degree in operations research from Western Michigan University, and a Ph.D. degree in industrial and systems engineering from West Virginia University. His research interests include computational intelligence, artificial neural networks, metaheuristics, and their applications to various optimization problems.

**Address for Correspondence:** Ellips Masehian, Faculty of Engineering, Tarbiat Modares University, Jalale-ale-Ahmad Highway, Tehran, 14115-317, Iran. E-mail: masehian@modares.ac.ir.

# Roadmap-Based Path Planning

## Using the Voronoi Diagram for a Clearance-Based Shortest Path

The path-planning problem was originally studied extensively in robotics, and, through this research, it has gained more relevance in areas such as computer graphics, simulations, geographic information systems (GIS), very-large-scale integration (VLSI) design, and games. Path planning still remains one of the core problems in modern robotic applications, such as the design of autonomous vehicles and perceptive systems [31], [39]. The basic path-planning problem is concerned with finding a good-quality path from a source point to a destination point that does not result in collision with any obstacles. Depending on the amount of the information available about the environment, which can be completely or partially known or unknown, the approaches to path planning vary considerably. Also, the definition of a good-quality path usually depends on the type of a mobile device (a robot) and the environment (space), which has fostered the development of a rich variety of path-planning algorithms, each catering to a particular set of needs. Latombe [28] provides a comprehensive survey of different path-planning algorithms.

Computational geometry plays a special role in path-planning developments. Extensive methodologies that rely on geometric representation of the space, reveal topological properties of the agents (robots and/or obstacles), and allow the efficient dynamic position tracking and updates have been brought forward from computational

geometry to solve a specific set of path-planning problems. Such problems usually have a well-defined and deterministic set of objectives, regular geometric space representation, and specific functions that describe robotic movements. The problems also include planning a path and optimizing it (based on selected criteria such as the length, the smoothness, the cost, etc.) [7], solving problems involving evolutionary algorithms and swarm intelligence [2], and studying the behavior of a network of mobile robot agents [17].

The traditional computational geometry-based approaches to path planning can be classified into three basic categories: the cell decomposition method [35], the roadmap method [1], and the potential field method [40]. If robots are represented by polygonal objects, an approach based on the Minkowski sum is often used [24]. Both the cell decomposition and the roadmap methods along with the Minkowski sum method have their roots in computational geometry.

The cell decomposition method uses nonoverlapping cells to represent the free-space ( $C_f$ ) connectivity. The decomposition can be exact or approximate. An exact decomposition divides  $C_f$  exactly [4]. An approximation scheme Kambhampati discretizes  $C_f$  with cells. It decomposes the free space recursively, stopping when a cell is entirely in  $C_f$  or entirely inside an obstacle. Otherwise, the cell is further divided. Because of memory and time constraints, the recursive process stops when a certain degree of accuracy has been reached. The cell decomposition method, although simple to implement, seldom yields high-quality paths. The exact cell decomposition technique is faster than the



©DIGITAL VISION

Digital Object Identifier 10.1109/MRA.2008.921540

BY PRIYADARSHI BHATTACHARYA AND MARINA L. GAVRILOVA

approximate one, but the path obtained is not optimal. The approximate cell decomposition can yield near-optimal paths by increasing the grid resolution, but the computation time will increase drastically. There is also the known problem of digitization bias associated with using a grid. This stems from the fact that while searching for the shortest path in a grid, the grid distance is measured and not the Euclidean distance.

The roadmap method attempts to capture the free-space connectivity with a graph. This approach has several variations. The probabilistic roadmap method (PRM) [1], [23] represents the free-space connectivity with a graph whose vertices are generated randomly in free space and connected to the  $k$ -nearest neighboring vertices such that the connecting edges do not cross any obstacle. To expedite the graph creation, several sampling-based methods such as Ariadne's Clew algorithm [33], expansive space planner [20], random walk planner [11], rapidly exploring random tree [26], [27] have been proposed. For algorithmic details, refer to [29]. Some of the other popular roadmap-based approaches are based on computational geometry structures such as the visibility graph for the shortest path and the Voronoi diagram for a maximum clearance path.

The idea behind the potential field method is to assign a function similar to the electrostatic potential to each obstacle and then derive the topological structure of the free space in the form of minimum potential valleys. The robot is pulled toward the goal configuration as it generates a strong attractive force. In contrast, the obstacles generate a repulsive force to keep the robot from colliding with them. The path from the start to the goal can be found by following the direction of the steepest descent of the potential toward the goal [9]. The strength of this approach is that path planning can be done in real time by considering only the obstacles close to the robot. Information on the locations of all obstacles is not required beforehand. However, as only local properties are used in planning, the robot may get stuck at local minima and never reach the goal.

In this article, we chose the roadmap approach and utilized the Voronoi diagram to obtain a path that is a close approximation of the shortest path satisfying the required clearance value set by the user. A Voronoi diagram (a fundamental computational geometry structure) is defined as the partitioning of a plane with  $n$  points (generators) into convex polygons such that each polygon contains exactly one generator and every point in a given polygon is closer to its generator than to any other. For a more formal definition and properties, refer to [3]. The Voronoi diagram and its dual structure, the Delaunay triangulation, have been used in a wide variety of applications such as collision detection [14], extraction of crust and skeleton [16], swarm intelligence optimization [2], cluster analysis [7], and mobile robot agent network [17]. The Voronoi diagram is also a well-known roadmap in the path-planning literature, which has edges that provide a maximum clearance path among a set of disjoint polygonal obstacles. However, a path obtained directly from the Voronoi diagram may be far from optimal. It usually has many unnecessary turns, and the length of the path may be undesirably long at regions where the obstacles are far apart. In fact, it is worth noting that minimizing the path length and maximizing the clearance seemingly contradict each other, as increasing the clearance results in a longer path whereas reducing

the path length necessarily reduces the clearance from obstacles. We thought that it would be highly beneficial for many applications if an algorithm could be developed that would accept the minimum clearance required as an input parameter and produce a path that would be shortest while satisfying the minimum clearance requirement. The shortest path problem on its own can be viewed as only a special case when we set the clearance required to zero. The practical usefulness of this kind of an algorithm is apparent for many applications, including marine GIS, ship route planning, VLSI design, oil drill path planning, etc.

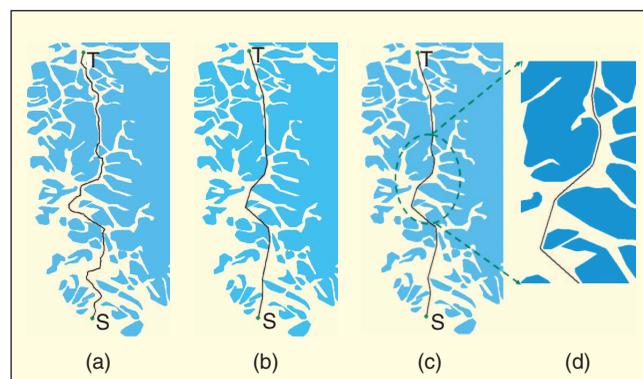
Figure 1 illustrates the output of the proposed algorithm on a spatial dataset. Figure 1(a) is the path obtained directly from the Voronoi diagram. Figure 1(b) shows the shortest path obtained with  $C_{\min} = 0$ . Figure 1(c) shows the final optimal path with a user specified clearance value of two units ( $\approx 20$  m).

The advantage of the proposed technique versus alternative path-planning methods is in its simplicity, versatility, and efficiency. For example, for planning the path for translational robots of varied dimensions, it has the capability to set the minimum clearance to a proportional value and the algorithm will find an optimal path for that minimum clearance, if one exists. The reported path is of more practical value because it is optimal with respect to both length and clearance. A high-quality approximation of the shortest path can also be obtained by the developed algorithm in much less time than by the popular and very well-performing visibility graph approach when we set  $C_{\min} = 0$ . To prove this claim, we experimentally compared the lengths of the paths obtained by these techniques and presented results in the experimental section.

In the next section, we provide an extensive literature review. An outline of the proposed algorithm is provided, followed by the detailed description of Voronoi based methodology. Later, a comprehensive analysis of experimental results is provided. Finally, the last section draws conclusions and outlines future work.

## Literature Review

As established in the previous section, the planning algorithms vary in the nature of the desired path and the information on



**Figure 1.** Shortest path and path with a minimum clearance. The figure depicts a part of the South American coastline from the ESRI world low-resolution layer: (a) shortest path from Voronoi-diagram-based roadmap, (b) shortest path using proposed algorithm ( $C_{\min} = 0$ ), (c) clearance based path using proposed algorithm ( $C_{\min} = 2$ ), and (d) zoomed path.

the environment. Recently, path planning in dynamic and complex environments has received considerable attention from researchers. In a dynamic environment, obstacles are allowed to move, and so, the environment can change dramatically over a period of time. An adaptive path-planning technique that takes cue from the previous situation can be found in [5]. In [10], the authors use an adaptive mesh for dynamic path planning based on a combination of graph- and grid-based representation of the environment. The PRM is very promising for dynamic path planning, as a big advantage of the PRM is that its complexity depends mostly on the difficulty of the path and to a much lesser extent on the global complexity of the environment or the dimension of the configuration space [36]. A recent algorithm based on the PRM for a dynamic environment can be found in [6]. However, in general, PRM-based approaches, being probabilistic in nature, do not meet any optimality criteria.

There has been research on planning algorithms coordinating motion of multiple robots [37]. The idea of this article is to perceive multiple robots as a single composite robot and then to determine a roadmap for this composite robot.

A highly interesting research on the utilization of computational geometry methods for coordination problems in dynamic systems was conducted by Cortés and Bullo [12]. They capitalize on the properties of geometric constructs (disk covering and sphere packing), nonsmooth analysis, and dynamic system approaches, studied in the context of networked robot interactions.

In this article, we focus on the roadmap-based path planning and utilize a powerful computational geometry data structure, the Voronoi diagram, to obtain a clearance-based shortest path. The advantage of using a Voronoi diagram as a roadmap over alternative methods, among which the visibility graph prevails, is its efficiency. The Voronoi diagram can be constructed in just  $O(n \log n)$  time, whereas even the fastest known algorithm for constructing visibility graph [15] can take  $O(n^2)$  time in the worst case when the visibility graph has  $O(n^2)$  edges. Since a Voronoi diagram has  $O(n)$  edges, querying for a path in a Voronoi diagram-based roadmap is also much faster than querying in a visibility graph. However, as mentioned before, the quality of path obtained directly from the Voronoi diagram may be far from optimal. Thus, improving the quality of the path (refining the path) is an important direction of research.

A general method for refining a path obtained from a roadmap based on classical numerical optimization techniques can be found in [25]. The authors apply costs to each edge and use an augmented Dijkstra's algorithm to determine an optimal path. The edges that are nearer to obstacles are assigned higher costs. However, there is no guarantee that the method will generate an optimal path, as the path is constrained to the edges in the roadmap. To improve the smoothness of the path obtained from a roadmap, a B-Spline approximation has been used in [21].

In [32], the authors combine the Voronoi diagram, visibility graph, and potential field approaches to path planning into a single algorithm to obtain a tradeoff between the optimal by safe and the shortest paths. The algorithm is fairly complicated, and although the path length is shorter than those obtained from the potential field method or the Voronoi diagram, it is

still not optimal. The path exhibits bumps and rudimentary turns and is not smooth.

Another recent work on reducing the length of the path obtained from a Voronoi diagram [19] involves constructing polygons at the vertices in the roadmap where more than two Voronoi edges meet. The path is smoother and shorter than that obtained directly from the Voronoi diagram, but there has been no attempt to reach optimality.

In [41], the authors create a new diagram called the  $VV^{(c)}$  diagram (the Visibility-Voronoi diagram for clearance  $c$ ). The motivation behind their work is similar to ours, i.e., to obtain an optimal path for a specified clearance value. The diagram evolves from the visibility graph to the Voronoi diagram as the value of  $c$  increases. Unfortunately, as the method is visibility based, the processing time is  $O(n^2 \log n)$ , which renders it impractical for large spatial datasets.

Our technique is able to generate near-optimal paths in just  $O(n \log n)$  time. The paths are the shortest possible while maintaining just the amount of clearance required. We conjecture that the obtained path is satisfactory in the homotopy sense, i.e., it can be continuously transformed to the true-optimal path without crossing any obstacles. Thus, the obtained approximation can be always refined to a near-optimal path. Experimental results support this conjecture. Another interesting property of the path is that it can be extraordinarily smooth when going around smooth obstacles. The smoothness only seems to help reduce the path length in such cases.

## Outline of the Method

The cornerstone of the methodology is the utilization of a powerful computational geometry data structure: the Voronoi diagram. We start by building the Voronoi diagram of the obstacles. The source and destination are dynamically inserted into the diagram, and they are connected to all Voronoi vertices of their Voronoi cells, respectively. Inserting the source and destination dynamically has two major advantages over simply connecting them to the nearest Voronoi vertex. There is no possibility of the connecting edges crossing an obstacle, as they are contained inside the Voronoi cell. Also, if we want to perform multiple queries, we can do so by simply dynamically deleting the old source and destination from the Voronoi diagram instead of rebuilding the diagram. We next remove all those edges in the resulting diagram that have a clearance less than the minimum clearance required ( $C_{\min}$ ). The resulting graph represents the roadmap used by our algorithm. Any path obtained directly from this roadmap between the source and the destination is guaranteed to have a clearance  $\geq C_{\min}$ . We apply Dijkstra's algorithm to determine the shortest path in the roadmap, but as mentioned before, the path is far from optimal. Our algorithm then refines the obtained path by removing unnecessary turns, so that the path has a minimum number of links but at the same time satisfies the minimum clearance criterion. This path, however, is still not optimal and has sharp corners. We next add Steiner points along the edges of this path and use a novel corner-cutting paradigm to convert it to an optimal path. The flowchart in Figure 2 illustrates the steps involved. We next discuss each stage in detail.

## Methodology in Detail

As established earlier, a powerful computational geometry data structure has been proposed to solve the problem of an optimal path generation between a source and a destination in the presence of simple disjoint polygonal obstacles. This method has a number of unique features, such as a novel application of the Voronoi diagram in the specified clearance context, the iterative refinement technique based on Steiner points for path optimization, and the possibility of performing dynamic updates on the structure during the path computation process. The main steps of the developed path generation algorithm are detailed next.

### Voronoi Diagram Construction

The construction of the Voronoi diagram of a set of polygons in the plane is both complex and time consuming. Instead, we approximate the obstacles with points on the boundary edges and generate the Voronoi diagram of the approximating points. The approximation of generalized Voronoi diagrams was first proposed by Sugihara [38]. Removal of the Voronoi edges that cross obstacles from this diagram provides a nice approximation of the Voronoi diagram of the original obstacles. We first create the Delaunay triangulation and then generate the Voronoi diagram from it in  $O(n)$  time.

We use the winged-edge data structure to represent the triangulation, as we found it more intuitive to use and update than the half-edge or quad-edge data structures. This structure is also more flexible, as the direction of the edges can be fixed arbitrarily. It maintains three collections—one for vertices, one for edges, and one for triangles. Each vertex stores the coordinates and the index of any one edge incident on it. A triangle stores the index of any one of the three bounding edges. Most of the topological information is stored inside an edge. It contains the indices of the end vertices, indices of triangles on left and right, and the clockwise and counter-clockwise predecessors and successors of the edge.

The randomized incremental algorithm for construction of the Delaunay triangulation [18] proved helpful for the path planning problem, as it allows dynamic insertion of new points in  $O(1)$  time without having to rebuild the entire triangulation. The incremental construction starts by creating a triangle that contains all the datapoints inside it. However, just containing the points is not a sufficient condition. The three corner points of this triangle should not lie inside the circumcircle of any of the triangles of the triangulation of the dataset. This is to ensure that the enclosing triangle does not influence the triangulation of the dataset.

We determine the minimum and maximum  $x$  and  $y$  values in the dataset as  $x_{\min}$ ,  $x_{\max}$ ,  $y_{\min}$ , and  $y_{\max}$ . The center point  $(x_0, y_0)$  of the dataset is then calculated as  $x_0 = x_{\min} + (x_{\max} - x_{\min})/2$  and  $y_0 = y_{\min} + (y_{\max} - y_{\min})/2$ . We set  $M$  as  $\text{Max}((x_{\max} - x_{\min})/2, (y_{\max} - y_{\min})/2)$ . Then, the three corner points of the external triangle  $\{p_1, p_2, p_3\}$  can be specified as

$$\begin{aligned} p_1 &= (x_0 + 3 \times M, y_0) \\ p_2 &= (x_0, y_0 + 3 \times M) \\ p_3 &= (x_0 - 3 \times M, y_0 - 3 \times M). \end{aligned}$$

Figure 3 shows the enclosing triangle for a simplified case where the center of the datapoints is  $(0, 0)$  and the span of the data along both  $x$  and  $y$  axes equals  $2 \times M$ . Once the enclosing triangle is constructed, the datapoints are added to the triangulation one by one, and after each insertion, the required topological changes are performed to restore the Delaunay properties. A new datapoint  $p_i$  can lie inside a triangle or on a triangle edge of the current triangulation. On the basis of this, two kinds of topological updates are required. In Figure 4(a), the datapoint lies inside a triangle. The topological update involves the addition of three new edges. In Figure 4(b), the datapoint lies on an edge. In this case, the old edge is deleted, and four new edges are created. The new triangles created as a result of the topology update may not satisfy the empty circle property. If an edge is found to be illegal, it is flipped so that the Delaunay properties are restored.

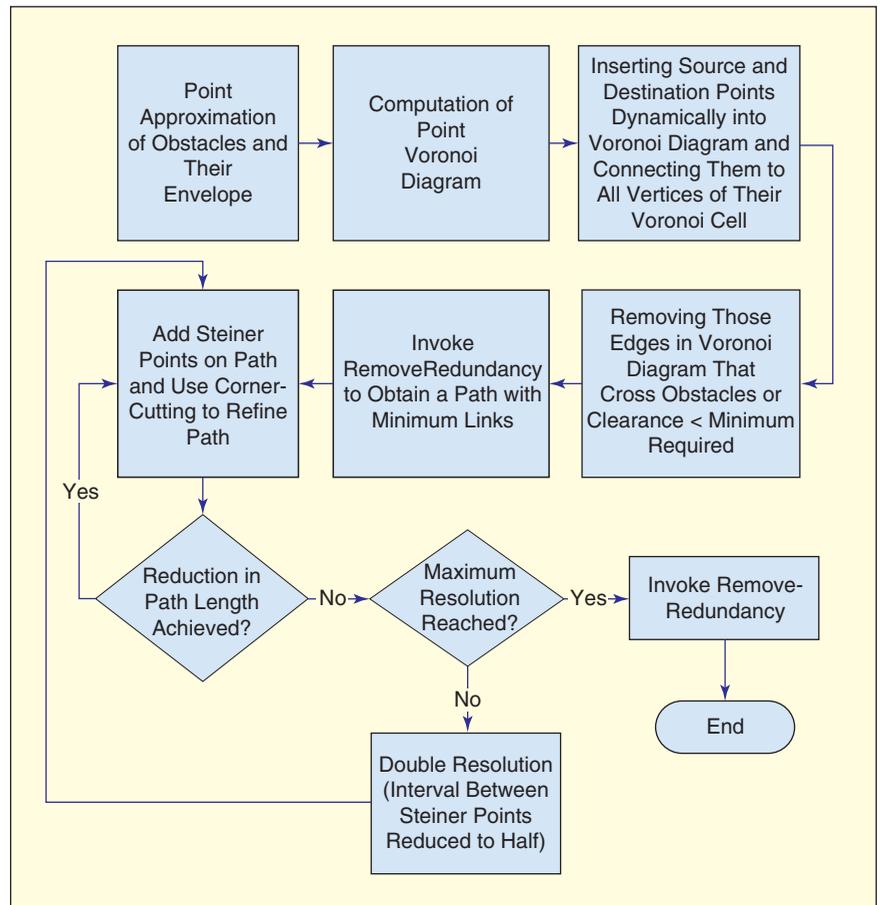


Figure 2. Different stages of the proposed algorithm.

## The advantage of using a Voronoi diagram as a roadmap over alternative methods, among which the visibility graph prevails, is in efficiency.

### Generation of Roadmap

Before generating the roadmap, we add the source and the destination to the Voronoi diagram dynamically. To query, we dynamically delete the old source and destination and insert the new ones. Details on point location and dynamic point deletion are provided in next section. The roadmap is generated by removing the edges from the Voronoi diagram that have an obstacle clearance  $< C_{\min}$ . There is one issue, however. If we consider the obstacles only for the Voronoi diagram construction, the resulting roadmap will not be complete. This is evident in Figure 5(a). This can be resolved by determining the minimum bounding box (mbb) of all obstacle vertices in linear time and expanding the mbb in all four directions

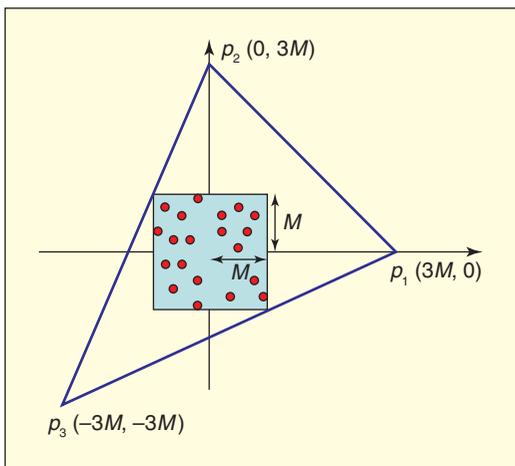


Figure 3. The enclosing triangle containing all datapoints.

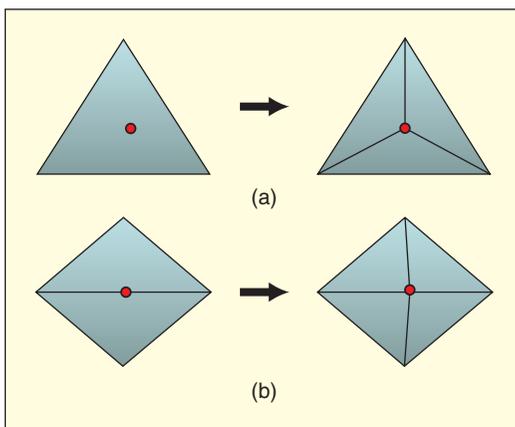


Figure 4. Topological updates associated with insertion of a new datapoint: (a) datapoint lies inside a triangle and (b) datapoint lies on a triangle edge.

by at least twice the minimum clearance required. The Voronoi diagram is then constructed from the points approximating the obstacles and this expanded mbb. The roadmap obtained from such a diagram is shown in Figure 5(b). The quality of the diagram depends on the spacing between approximation points. As can be observed in [30], if the spacing is too large, the Voronoi edges exhibit a zigzag pattern. However, it has been observed in [38] that a finer initial approximation yields results practically indistinguishable from the original Voronoi diagram. The proof was the utilization of the results of the approximating generalized Voronoi diagrams by ordinary Voronoi diagrams in the framework of path planning. Thus, it is possible to find the fine approximation (in our case established experimentally) so that the path obtained for the set of approximation points always satisfies the clearance requirement for the original obstacle set.

In Figure 6(a), the roadmap is generated using an expanded mbb. It can be observed that because of the concavity of the obstacle, there is a large deviation in the shortest path obtained from the roadmap. There are two main disadvantages to this. The first one is that the unnecessarily large deviation may make a shorter path appear longer, and this may result in the selection of a longer path from the roadmap as the shortest path. The second disadvantage is that the optimization step will take longer to execute. To eradicate this problem, we retract the approximation points on the mbb toward the nearest obstacle point so that the distance is a little greater than  $2 \times C_{\min}$ . The Voronoi diagram is then constructed from the obstacle approximation points and the points on this retracted boundary. This results in a much improved shortest path obtained from the roadmap as evident in Figure 6(b).

Figure 7(a) shows the Voronoi diagram obtained from the point approximation of an obstacle and the retracted boundary for a dataset. Figure 7(b) shows the roadmap extracted from the Voronoi diagram. In practice, to reduce the computation time for the clearance check, we first build a quadtree of the mbb's of the obstacle edges. When checking for clearance of a Voronoi edge  $e$ , we first determine in  $O(\log n)$  time all the

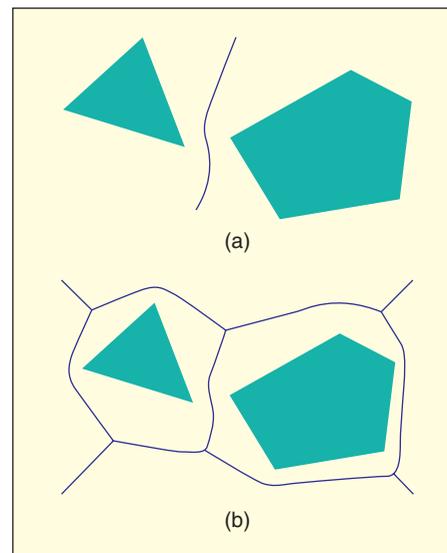


Figure 5. Complete and incomplete roadmaps: (a) without outer boundary and (b) with outer boundary.

obstacle edge indices whose mbbs overlap with the mbb of  $e$ , expanded in all four directions by  $C_{\min}$ . This yields a constant number of edges against which the actual clearance check is carried out. As the expanded mbbs of the Voronoi edges constituting the medial axis of the obstacle and some of the outer edges do not overlap with the mbb of any obstacle edge, they are not removed and remain in the roadmap. These edges do not influence the determination of the shortest path in any way.

### Dynamic Insertion and Deletion of Source and Destination

To insert the source and destination into the triangulation, we perform a walk in the triangulation to locate the triangle containing the point. Algorithmic details about walking in a triangulation have been presented by Devillers et al. [13]. To allow requerying, we delete the old source and destination dynamically from the Voronoi diagram and dynamically insert the new ones. For dynamic deletion of points from the Delaunay triangulation, we followed the algorithm outlined in [34] with a little modification. Because of limitations of the floating-point arithmetic, the number of neighbors of the point ( $P$ ) to be deleted may never get reduced to three, resulting in an infinite loop. To avoid this problem, the authors in [34] consider the circumcircle to be shrunk by a small amount while performing the Incircle test. We adopt a different approach. Even if some of the remaining neighbors of  $P$  fall inside the circumcircle of the potential triangle, we flip the edge. This guarantees that the number of neighboring vertices of  $P$  will

always be reduced to three. We then perform a normal Incircle test on the flipped edges to ensure that the triangulation remains Delaunay. Figure 8 shows a program snapshot of the topological events in deleting the encircled point from a simple dataset.

### Removal of Redundant Vertices and Obtaining a Path with Minimum Number of Links

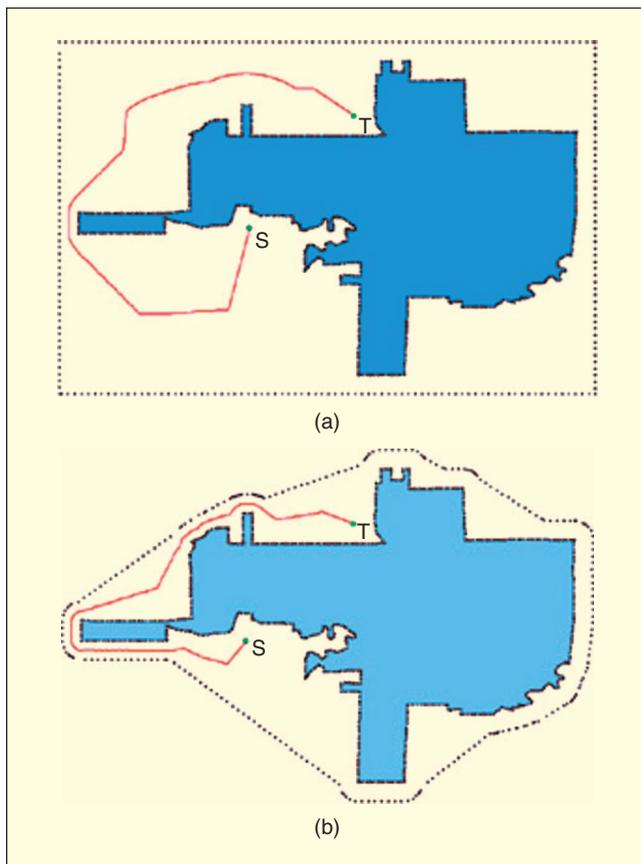
The shortest path obtained in the previous step has many unnecessary turns and redundant vertices. This step removes all redundant vertices and generates a path that has a minimum number of links or edge connections. A minimum number of links ensures that the iterative refinement in next step consumes less time. The method is simple.

For a vertex  $v_i$  on the path ( $i = \{1 \dots n - 2\}$ ), we check whether the line segment  $\overline{v_i v_{i+2}}$  has a clearance  $\geq C_{\min}$ . If so, we remove  $v_{i+1}$  from shortest path and repeat the process from vertex  $v_{i+2}$ . If not, we retain  $v_{i+1}$  and consider it as the next vertex for processing. We provide the pseudocode in Algorithm 1.

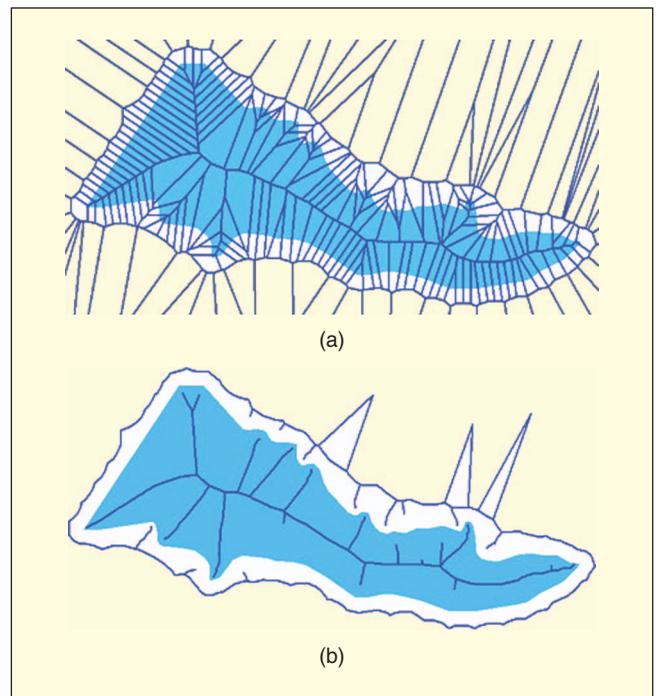
To efficiently determine the clearance of an edge ( $e$ ), we build a quadtree of the mbbs of the obstacle edges. We determine the mbb of  $e$  and expand it by  $C_{\min}$  in all four directions. In  $O(\log n)$  time, it is possible to determine the edges whose mbbs overlap with the expanded mbb of  $e$ , and the clearance check is carried out for only these constant number of edges. This ensures the complexity of this step for all edges is  $O(n \log n)$ .

### Iterative Refinement Using a Corner-Cutting Technique

The main idea behind the iterative refinement of the path is the utilization of the Steiner points. We first add the Steiner points along the edges of the path at regular interval  $\Delta$ . For our



**Figure 6.** Shortest paths obtained from (a) roadmap using expanded mbb and (b) roadmap using retracted boundary.

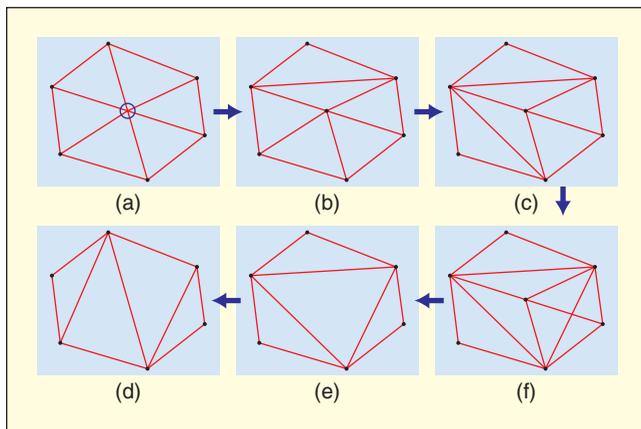


**Figure 7.** Roadmap generation using point Voronoi diagram: (a) Voronoi diagram of points approximating object and retracted boundary and (b) roadmap extracted from Voronoi diagram.

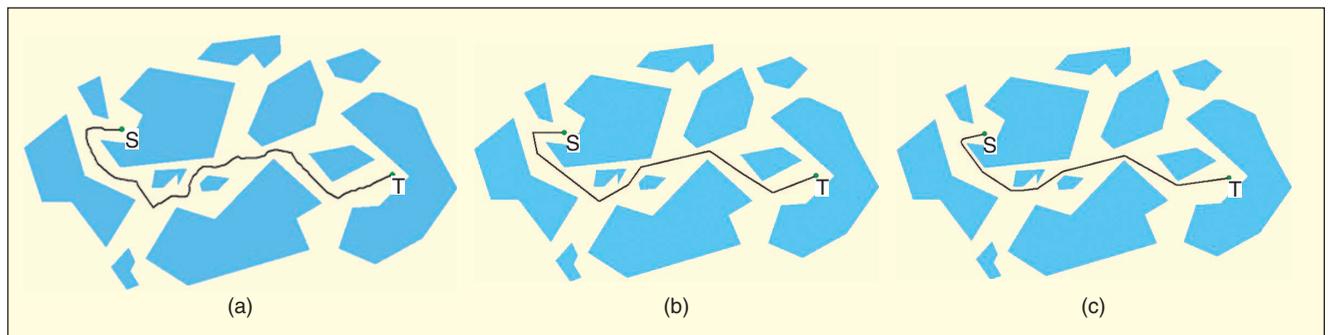
experiments, we set  $\Delta$  to roughly one fourth the average obstacle edge length. This is two times the resolution we used to approximate the obstacles with points. Let  $V$  be a vertex on the shortest path other than source and destination. Let  $e_1$  and  $e_2$  be the two edges incident on  $V$ . We define the first Steiner point along  $e_1$  as a Steiner point that lies on  $e_1$  at  $\Delta$  distance away from  $V$ , the second Steiner point is  $2\Delta$  away from  $V$ , and so on. We try to connect the first Steiner point on  $e_1$  with that on  $e_2$ . If the connecting edge satisfies the minimum clearance, we move to the second Steiner point along both edges and try to connect these. The process continues until an intersection is detected, or the clearance from obstacles falls below the required minimum clearance, or the endpoint of one of the incident edges on  $V$  is reached. We then replace  $V$  with the last Steiner points that we could successfully connect with an edge. If we failed to connect even the first Steiner points along the two incident edges, we retain  $V$ . The path cannot be shortened any further at point  $V$  at this resolution. When no more reduction in path length is possible for any of the vertices, we double the resolution (i.e., set the interval between Steiner points along the edges to  $\Delta/2$ ) and repeat the process. The iteration continues until the resolution reaches a maximum precalculated value (Figure 2). Figure 9 shows a sample path obtained after the different stages.

## Experimental Results

We now demonstrate that the path obtained by our algorithm is optimal with respect to length and clearance from obstacles



**Figure 8.** Dynamic deletion of a point from the Delaunay triangulation.



**Figure 9.** Clearance-based path (different stages): (a) path from roadmap, (b) path after minimizing links in path, (c) path after iterative refinement.

for a specified value of minimum clearance ( $C_{min}$ ). The refined path is also smooth. We also show that the proposed algorithm outperforms the popular (and efficient) visibility graph-based approach with regard to both speed and quality of the path.

We have put our algorithm to test on a real-world data from the large spatial datasets. The spatial datasets considered are mostly represented in the form of Environmental Systems Research Institute (ESRI) shapefiles. When dealing with a spatial data, it is often required to substantially increase the resolution of the data so that extraordinary cases such as polygons degenerating into points can be avoided. After performing the computations, we scale down the result to display scale.

### Algorithm 1. RemoveRedundancy( $C_{min}, S_{old}, S_{new}$ )

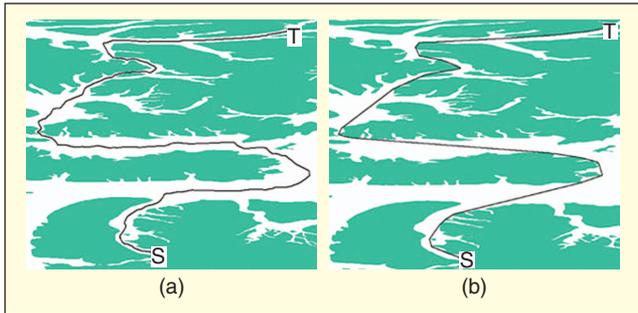
```

1 Input:  $S_{old}$ : Vertices on shortest path (in sequence)
   between source and destination. Obtained by ap-
   plying Dijkstra's algorithm on roadmap.
2  $C_{min}$ : minimum clearance required
3 Output:  $S_{new}$ : Path obtained after processing
4
5  $N \leftarrow \text{NumberOfVertices}(S_{old})$  { $\text{NumberOfVertices}(S)$ :
   number of vertices in  $S$ }
6 if  $N = 2$  then
7    $S_{new} \leftarrow S_{old}$  {trivial case}
8   return
9 end if
10 repeat
11    $N \leftarrow \text{NumberOfVertices}(S_{old})$ 
12    $S_{new} \leftarrow \Phi$ 
13   for  $i = 1$  to  $N - 2$  do
14      $S_{new} \leftarrow S_{new} \cup S_{old}[i]$ 
15     if  $v_i v_{i+2}$  has clearance  $\geq C_{min}$  then
16        $i \leftarrow i + 1$  {skip over next vertex}
17     end if
18   end for
19   for  $j = i$  to  $N$  do
20      $S_{new} \leftarrow S_{new} \cup S_{old}[j]$  {complete path}
21   end for
22    $S_{old} \leftarrow S_{new}$ 
23 until  $\text{NumberOfVertices}(S_{new}) = N$  {path remains
   unchanged}

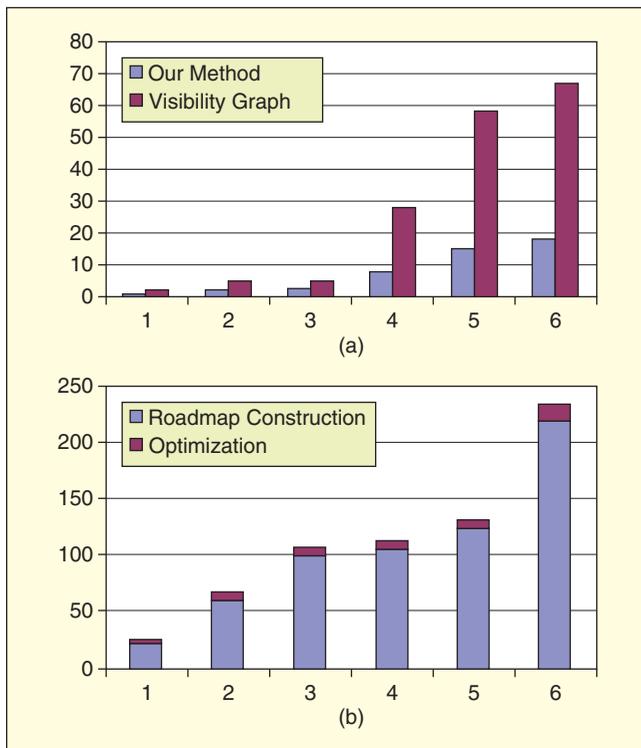
```

Users can zoom in on any part to see magnified view of a selected portion. All experiments have been carried out on a 1.6-GHz Intel Centrino Duo processor with 0.99-GB RAM. The programs have been implemented in Visual C++. The source and target are indicated with S and T in all figures.

In Figure 10, we visually compare the path obtained directly from the Voronoi diagram-based roadmap to that obtained after



**Figure 10.** Shortest path using proposed algorithm (number of vertices in dataset = 11,797): (a) shortest path obtained from Voronoi-diagram-based roadmap, (b) optimal path after iterative refinement ( $C_{\min} = 0$ ).



**Figure 11.** CPU time estimates.  $x$  axis is the shapefile index and  $y$  axis is the time consumed in seconds: (a) CPU time of our method versus visibility graph approach, (b) breakup of CPU time.

**Table 1. Shapefile index versus number of vertices.**

	S1	S2	S3	S4	S5	S6
Vertex cnt. [Fig. 11(a)]	155	338	355	739	939	1,866
Vertex cnt. [Fig. 11(b)]	2,599	4,065	4,769	5,365	8,070	11,797

iterative refinement ( $C_{\min} = 0$ ) on a portion of the real-world data. The shortest path in Figure 10(b) can be observed to wrap around the obstacles very similar to a path obtained from a visibility graph. The result shows that our algorithm is effective even in complicated environments.

We provide time estimates in Figure 11. The number of vertices for the six datasets for Figure 11(a) and 11(b) are mentioned in the first and second rows of Table 1, respectively. Figure 11(a) shows the time consumed by our algorithm and the visibility graph approach on a number of spatial datasets. The time (in seconds) includes the time to build the roadmap and determine the path between the source and the destination. The time estimated for our method includes the time for path optimization. It can be observed that the time difference becomes more pronounced as the number of vertices in the dataset increases. In Figure 11(b), we show the breakup of roadmap construction time and optimization time for some large datasets. The time consumed by the visibility graph approach for these datasets is extremely high, and therefore, we do not mention it here.

## Conclusions

In this article, a computational geometry data structure has been proposed to solve the problem of an optimal path generation between a source and a destination in the presence of simple disjoint polygonal obstacles. The method has a number of unique features, such as a novel application of the Voronoi diagram in the specified clearance context, the iterative refinement technique based on the Steiner points for path optimization, and the possibility of performing dynamic updates on the structure during the path computation process. The obtained path is optimal with respect to length and clearance from the obstacles for a specified value of minimum clearance ( $C_{\min}$ ). The refined path is also observed to be smooth. The proposed algorithm outperforms the popular (and efficient) alternative approach with regard to speed as well as the quality of the path and is flexible to allow various required clearance settings and source/destination location changes.

Future studies will involve porting the algorithm to other platforms and applications (including ArcGIS compatibility and direct shape file visualization options) and extending the algorithm to work with three-dimensional data.

## Acknowledgment

We would like to acknowledge GEOIDE and NSERC grant-ing agencies for continuous support of this project.

## Keywords

Path planning, Voronoi diagram, clearance-based path.

## References

- [1] N. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1996, pp. 113–120.
- [2] R. Apu and M. L. Gavrilova, "Battle swarm: An evolutionary approach to complex swarm intelligence," in *2006 Proc. 9th Int. Conf. Computer Graphics and Artificial Intelligence*, Eurographics, Limoges, France, May 2006, pp. 139–150.
- [3] F. Aurenhammer, "Voronoi diagrams—A survey of a fundamental geometric data structure," *ACM Comput. Surv.*, vol. 23, no. 3, pp. 345–405, 1991.

- [4] F. Avnaim, J. D. Boissonnat, and B. Faverjon, "A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles," in *Proc. IEEE Int. Conf. Robotics and Automation*, Apr. 1988, vol. 3, pp. 1656–1661.
- [5] J. Baltes and N. Hildreth, "Adaptive path planner for highly dynamic environments," *Lect. Notes Comput. Sci.*, vol. 2019, pp. 76–85, 2001.
- [6] K. Belghith, F. Kabanza, L. Hartman, and R. Nkambou, "Anytime dynamic path-planning with flexible probabilistic roadmaps," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2006, pp. 2372–2377.
- [7] P. Bhattacharya and M. L. Gavrilova, "CRYSTAL—A new density-based fast and efficient clustering algorithm," in *Proc. 3rd Int. Symp. Voronoi Diagrams in Science and Engineering*, 2006, pp. 102–111.
- [8] P. Bhattacharya and M. L. Gavrilova, "Geometric algorithms for clearance based optimal path computation," in *Proc. 15th ACM Int. Symp. Advances in Geographic Information Systems (ACM GIS)*, 2007, pp. 208–311.
- [9] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 5, pp. 1179–1187, 1998.
- [10] P. Brož, I. Kolingerová, R. A. Apu, M. L. Gavrilova, and P. Zítka, "Path planning in dynamic environment using an adaptive mesh," in *Proc. Spring Conf. Computer Graphics*, 2007, pp. 172–178.
- [11] S. Carpin and G. Pilonetto, "Robot motion planning using adaptive random walks," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2003, pp. 3809–3814.
- [12] J. Cortés and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM J. Control Optim.*, vol. 44, no. 5, pp. 1543–1574, 2005.
- [13] O. Devillers, S. Pion, and M. Teillaud, "Walking in a triangulation," *Int. J. Found. Comput. Sci.*, vol. 13, no. 2, pp. 181–199, 2002.
- [14] M. L. Gavrilova and J. G. Rokne, "Collision detection optimization in a multi-particle system," *Int. J. Comput. Geometry Appl.*, vol. 13, no. 4, pp. 279–301, 2003.
- [15] S. K. Ghosh and D. M. Mount, "An output-sensitive algorithm for computing visibility graphs," *SIAM J. Comput.*, vol. 20, no. 5, pp. 888–910, 1991.
- [16] C. Gold, "Crust and anti-crust: A one-step boundary and skeleton extraction algorithm," in *Proc. 15th ACM Symp. Computational Geometry*, 1999, pp. 189–196.
- [17] R. Graham and J. Cortés, "Asymptotic optimality of multicenter Voronoi configurations for random field estimation," *IEEE Trans. Automat. Contr.*, submitted for publication.
- [18] L. J. Guibas, D. E. Knuth, and M. Sharir, "Randomized incremental construction of Delaunay and Voronoi diagrams," *Algorithmica*, vol. 7, no. 1, pp. 381–413, 1992.
- [19] D.-H. Yang and S.-K. Hong, "A roadmap construction algorithm for mobile robot path planning using skeleton maps," *Adv. Robotics*, vol. 21, no. 1, pp. 51–63, 2007.
- [20] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," *Int. J. Comput. Geometry Appl.*, vol. 4, no. 4/5, pp. 495–512, 1999.
- [21] J. M. Ibarra-Zannatha, J. H. Sossa-Azuela, and H. Gonzalez-Hernandez, "A new roadmap approach to automatic path planning for mobile robot navigation," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics (Humans, Information, and Technology)*, 1994, vol. 3, pp. 2803–2808.
- [22] S. Kambhampati and L. S. Davis, "Multiresolution path planning for mobile robots," *IEEE J. Robot. Automat.*, vol. 2, no. 3, pp. 135–145, 1986.
- [23] L. E. Kavraki and J.-C. Latombe, *Probabilistic Roadmaps for Robot Path Planning*. New York: Wiley, 1997.
- [24] K. Kedem and M. Sharir, "An efficient motion planning algorithm for a convex rigid polygonal object in 2-dimensional polygonal space," *Discrete Comput. Geom.*, vol. 5, no. 1, pp. 43–75, 1990.
- [25] J. Kim, R. A. Pearce, and N. M. Amato, "Extracting optimal paths from roadmaps for motion planning," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2003, pp. 2424–2429.
- [26] J. Kuffner, Jr., and J.-C. Latombe, "Interactive manipulation planning for animated characters," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2000, pp. 417–418.
- [27] J. Kuffner, Jr., and S. LaValle, "RRT-Connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2000, vol. 2, pp. 995–1001.
- [28] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.
- [29] S. M. LaValle. (2005). *Planning algorithms* [Online]. Available: <http://msl.cs.uiuc.edu/planning>
- [30] R. Mahkovic and T. Slivnik, "Generalized local Voronoi diagram of visible region," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1998, vol. 1, pp. 349–355.
- [31] S. Martínez, J. Cortés, and F. Bullo, "Motion planning and control problems for underactuated robots," in *Control Problems in Robotics* (Springer Tracts in Advanced Robotics Series, vol. 4). New York: Springer-Verlag, 2003, pp. 59–74.
- [32] E. Masehian and M. R. Amin-Naseri, "A Voronoi diagram-visibility graph-potential field compound algorithm for robot path planning," *J. Robot Syst.*, vol. 21, no. 6, pp. 275–300, 2004.
- [33] E. Mazer, J. M. Ahuactzin, and P. Bessière, "The Ariadne's clew algorithm," *J. Artif. Intell. Res.*, vol. 9, pp. 295–316, 1998.
- [34] M. A. Mostafavi, C. Gold, and M. Dakowicz, "Delete and insert operations in Voronoi/Delaunay methods and applications," *Comput. Geosci.*, vol. 29, pp. 523–530, 2003.
- [35] H. Noliborio, T. Naniwa, and S. Arimoto, "A quadtree-based path-planning algorithm for a mobile robot," *J. Robot Syst.*, vol. 7, no. 4, pp. 555–574, 1990.
- [36] M. H. Overmars, "Recent developments in motion planning," *Lect. Notes Comput. Sci.*, vol. 2331, pp. 3–13, 2002.
- [37] P. Švestka and M. Overmars, "Coordinated path planning for multiple robots," *Robot Autom. Syst.*, vol. 23, no. 3, pp. 125–152, 1998.
- [38] K. Sugihara, "Approximation of generalized Voronoi diagrams by ordinary Voronoi diagrams," *CVGIP: Graph. Models Image Process.*, vol. 55, no. 6, pp. 522–531, 1992.
- [39] G. Taylor and L. Kleeman, "Robust range data segmentation using geometric primitives for robotic applications," in *Proc. 5th IASTED Int. Conf. Signal and Image Processing*, 2003, pp. 467–472.
- [40] C. W. Warren, "Global path planning using artificial potential fields," in *Proc. IEEE Conf. Robotics and Automation*, 1989, pp. 316–321.
- [41] R. Wein, J. P. Van den Berg, and D. Halperin, "The Visibility-Voronoi complex and its applications," in *Proc. 21st Annu. Symp. Computational Geometry*, 2005, pp. 63–72.

**Priyadarshi Bhattacharya** is a senior software developer at Intermap Technologies Corporation, Calgary, Canada, developing software for GIS and 3-D visualization. He graduated with M.Sc. from the Department of Computer Science, University of Calgary, in 2007, where he worked in Spatial Analysis in Computational Sciences (SPARCS) Laboratory on projects related to optimal path planning in marine environments.

**Marina L. Gavrilova** is with the Department of Computer Science, University of Calgary. She is a founder and codirector of the SPARCS Laboratory and the Biometric Technologies Laboratory (BTLab). She has published over 100 journal and conference articles, books and book chapters, in addition to editing a number of journal special issues. In 2007, she became an editor-in-chief for *LNCS Transactions on Computational Science Journal*, Springer-Verlag. She currently serves on the Editorial Board for *International Journal of Computational Sciences and Engineering*, *Computer Graphics and CAD/CAM Journal*, and *Journal of Biometrics*. She is a Member of the IEEE.

**Address for Correspondence:** Priyadarshi Bhattacharya, SW Dev, Intermap Technologies Corp., 1074 Northmount Dr. NW, Calgary, AB T2L 0C2, Canada. E-mail: [pbhattacharya@intermap.com](mailto:pbhattacharya@intermap.com).

# Online Algorithms with Discrete Visibility

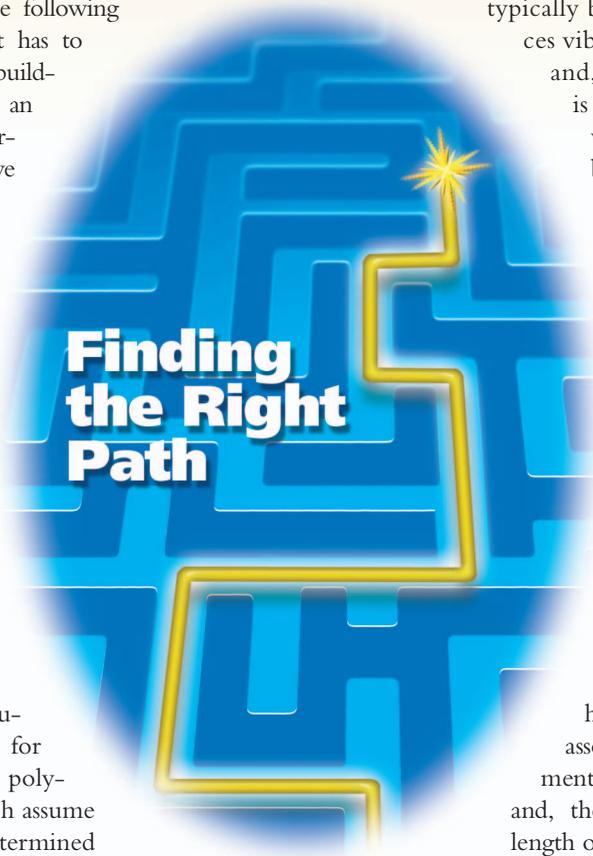
## Exploring Unknown Polygonal Environments

The context of this work is the exploration of unknown polygonal environments with obstacles. Both the outer boundary and the boundaries of obstacles are piecewise linear. The boundaries can be nonconvex. The exploration problem can be motivated by the following application. Imagine that a robot has to explore the interior of a collapsed building, which has crumbled due to an earthquake, to search for human survivors. It is clearly impossible to have a knowledge of the building's interior geometry prior to the exploration. Thus, the robot must be able to see, with its onboard vision sensors, all points in the building's interior while following its exploration path. In this way, no potential survivors will be missed by the exploring robot. The exploratory path must clearly reflect the topology of the free space, and, therefore, such exploratory paths can be used to guide future robot excursions (such as would arise in our example from a rescue operation).

There are several online computational geometry algorithms for searching or exploring unknown polygons with or without holes, which assume that the visibility region can be determined in a continuous fashion from each point on the path of a robot. Although this assumption is reasonable in the case of a human watchman, it may not be practical in the robotic case for several reasons. First, autonomous robots can carry only a limited amount of onboard computing capability. In the current state of the art, computer vision

algorithms that could compute visibility polygons are time consuming. The computing limitations suggest that it may not be practically feasible to continuously compute the visibility polygon along the robot's trajectory. Second, for good visibility, the robot's camera will typically be mounted on a mast. Such devices vibrate during the robot's movement, and, hence, for good precision (which is required to compute an accurate visibility polygon) the camera must be stationary at each view. Therefore, it seems feasible to compute visibility polygons only at a discrete number of points. Hence, in this article, we assume that the visibility polygon is computed at a discrete number of points.

Although the earlier discussion suggests that a robot can only compute visibility polygons at discrete points on its path, it is not clear whether the total cost for a robotic exploration is dominated by the number of visibility polygons that it computes or the length of the path it travels. The computational geometry literature has typically assumed that the cost associated with a robot's physical movement dominates all other associated costs, and, therefore, minimizing the Euclidean length of the path of a robot is considered as the main criterion or the sole criterion for designing motion-planning algorithms. The essential components that contribute to the total cost required for a robotic exploration can be analyzed as follows. Each move will have two associated costs. First, there is the time required to physically execute the move. If we crudely assume that the robot moves at a constant rate,  $r$ , during a move, the total time required for motion will be  $d/r$ , where  $d$  is the total path length



©DIGITAL VISION

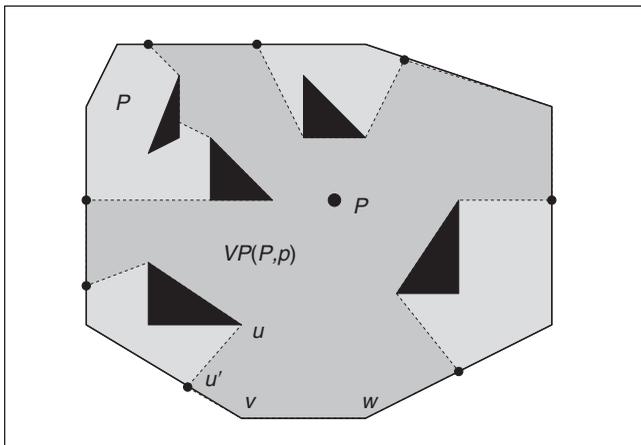
Digital Object Identifier 10.1109/MRA.2008.921542

BY SUBIR KUMAR GHOSH, JOEL WAKEMAN BURDICK,  
AMITAVA BHATTACHARYA, AND SUDEEP SARKAR

followed by the robot during the exploration. Second, in an exploratory process where the robot has no a priori knowledge of the environment's geometry, each move must be planned immediately prior to the move so as to account for the most recently acquired geometric information. The robot will be stationary during this process, which we assume to take time  $t_M$ . Because straight-line paths are the easiest to plan, and since any curvilinear path can be well approximated by straight-line segments, we assume that each move consists of a straight segment. For the reasons outlined previously, we assume that the robot is stationary during each sensing operation, which we assume takes time  $t_S$ . Let  $N_M$  and  $N_S$  be the number of moves and the number of sensor operations, respectively, which are required to complete the exploration of  $P$ . Hence, the total cost of an exploration is equated to the total time,  $T$ , which is required to explore  $P$ :  $T(P) = t_M N_M + t_S N_S + d/r$ . Now,  $(t_M N_M + t_S N_S)$  can be viewed as the time required for computing and maintaining visibility polygons, and it is, indeed, a significant fraction of  $T(P)$  because computer vision algorithms consume significant time on modest computers in a relatively cluttered environment. Thus, the goal in this article is to develop an exploration algorithm for a robot that minimizes the number of visibility polygons computed during the exploration of  $P$ .

Here, it is assumed that a vision sensor can see any object in  $P$ , irrespective of its distance from the sensor, which is not the case in practice. Computer vision range sensors or algorithms, such as stereo or structured-light range finder, can reliably compute scene locations only up to a depth  $R$ . The reliability of depth estimates is inversely related to the distance from the camera. Thus, the range measurements from a vision sensor for objects that are far away are not at all reliable. This suggests that it is necessary to restrict visibility polygons by a range distance  $R$ . Therefore, only the region of  $P$  within  $R$  is considered to be visible from the camera of the robot. We refer to the visibility polygon under this range restriction as the restricted visibility polygon. Observe that restricted visibility polygons need not always be closed regions. For exploring  $P$ , an online algorithm with a range restriction needs more views than an online algorithm without any range restriction.

This article is organized as follows. In the next section, we discuss the background for our problem on robotic exploration and



**Figure 1.** Visibility polygon of  $P$  from a point  $p$ .

establish its relation to the art gallery problem. In the section “An Exploration Algorithm,” we present an online exploration algorithm without a range restriction and show that the algorithm computes at most  $r + 1$  visibility polygons, where  $r$  is the total number of reflex vertices in  $P$ . The competitive ratio of the algorithm is  $(r + 1)/2$ . We also show that  $r + 1$  visibility polygons are sometimes necessary to see the entire  $P$ . In the “An Exploration Algorithm with Range Restriction” section, we present an online exploration algorithm by restricting visibility polygons by a range distance  $R$ . The maximum number of visibility polygons that may be needed by this algorithm to explore  $P$  of  $n$  vertices with  $h$  holes is bounded by

$$\left\lceil \frac{8 \times \text{Area}(P)}{3 \times R^2} \right\rceil + \left\lceil \frac{\text{Perimeter}(P)}{R} \right\rceil + r + h + 1.$$

The competitive ratio of the algorithm is

$$\left\lceil \frac{8\pi}{3} + \frac{\pi R \times \text{Perimeter}(P)}{\text{Area}(P)} + \frac{(r + h + 1) \times \pi R^2}{\text{Area}(P)} \right\rceil.$$

Finally, we conclude the article with a few remarks.

## Background for Robotic Exploration

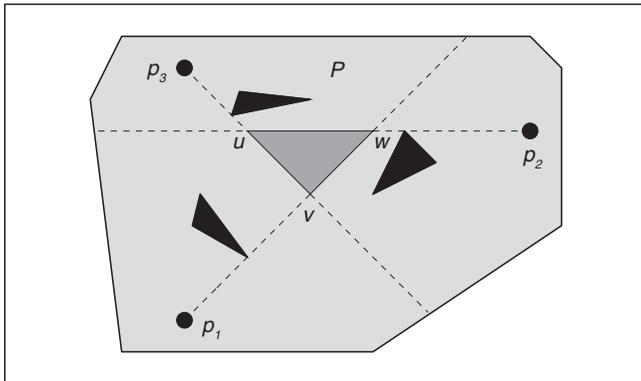
In this article, the robot is assumed to be a point. The polygonal environment  $P$  is assumed to consist of a boundary polygon,  $\mathcal{B}$ , populated with  $h$  polygonal obstacles, or holes with a total of  $n$  vertices. The assemblage of  $\mathcal{B}$  and the obstacles is the polygon  $P$  (see Figure 1). The free space,  $\mathcal{F}$  is the complement of the holes in the interior of  $P$ . Prior to the exploration of  $P$ , the robot has no knowledge of the geometry of  $P$ , the number of edges, or the number of holes. However, we make three assumptions. 1) We assume that the robot can locate its current location relative to a fixed reference configuration. Denote  $p = (x, y)$  as the coordinates of the robot relative to this fixed reference. 2) The robot can compute the visibility polygon,  $VP(P, p)$  from a viewing point  $p \in P$ . The visibility polygon  $VP(P, p)$  is the set of all points of  $P$  visible from  $p$  [9] and is bounded by some combination of polygonal edges, partial polygonal edges, and constructed edges as shown in Figure 1. For example,  $uu'$  is a constructed edge,  $u'v$  is a partial polygonal edge, and  $vw$  is a polygonal edge. One of the endpoints of a constructed edge is always a reflex vertex. 3) We assume that the viewing point  $p$  and two endpoints of every constructed edge in  $VP(P, p)$  are collinear. For example,  $p, u,$  and  $u'$  are collinear (see Figure 1).

To explore an unknown polygonal environment  $P$ , the robot starts from a given position and sees all points of the free space  $\mathcal{F}$  incrementally. It may appear that it is enough to see all vertices and edges of  $P$  to see the entire free space. However, this is not the case, as shown by the example in Figure 2. In this figure, three views from  $p_1, p_2,$  and  $p_3$  are sufficient to see all vertices and edges of  $P$ , but the shaded region  $uvw$  of  $\mathcal{F}$  cannot be seen from these views. This suggests that to see the entire free space the algorithm must ensure that all triangles in the triangulation of  $P$  have been explored. Once  $P$  is known and triangulated, triangulation of the free space becomes a map of  $\mathcal{F}$ ,

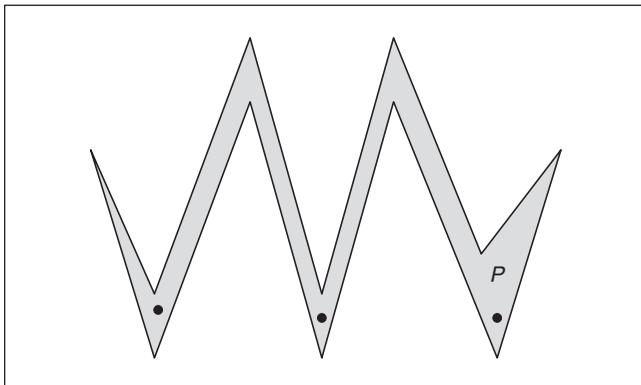
and it can then be used by the robot for its movement between any two locations in  $\mathcal{F}$  for future operations.

Suppose  $p_1, p_2, \dots, p_k$  be the viewing points in  $P$  such that an optimal exploration algorithm for a point robot has computed visibility polygons from these points, where  $p_1$  is the starting position of the robot. For safety reasons, the next viewing point is always chosen within the region of  $P$  that has so far been explored by the robot. Therefore, 1)  $\bigcup_{i=1}^k VP(P, p_i) = P$ , 2)  $p_i \in VP(P, p_j)$  for some  $j < i$ , and 3)  $k$  is minimum. So,  $P$  can be guarded by placing stationary guards at  $p_1, p_2, \dots, p_k$ . So, the exploration problem for a point robot is the art gallery problem for stationary guards [14], with the additional constraint 2). Hence, our exploration algorithms for a point robot are approximation algorithms for this variation of the art gallery problem (when  $P$  is not known a priori), which also seems to be nondeterministic polynomial time-hard. For the standard art gallery problem (i.e.,  $P$  is known a priori and constraint 2) is omitted), Ghosh [8] proposed approximation algorithms for minimum vertex and edge-guard problems for polygons with or without holes that run in  $O(n^5 \log n)$  time and yield solutions at most  $O(\log n)$  times the optimal.

In the standard art gallery problem, guards can be placed anywhere inside  $P$ , and, therefore, there may be guards that cannot be seen by any other guard (see Figure 3). We know that  $\lfloor n/3 \rfloor$  stationary guards are sufficient and sometimes necessary for guarding  $P$ , which contains no holes [14]. Suppose



**Figure 2.** Three views are enough to see all vertices and edges of  $P$  but not the entire region.



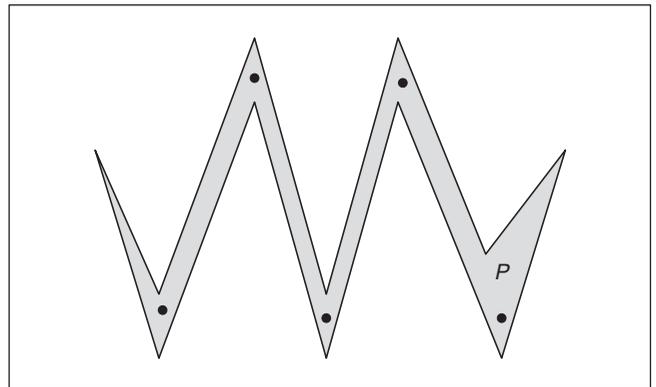
**Figure 3.** Three guards can see the entire  $P$ , but they are not mutually visible.

the guards  $g_1, g_2, \dots, g_k$  are placed in  $P$  for security reasons in a such way that each guard  $g_i$  for  $i > 1$  is visible at least from one other guard  $g_j$  for  $i < j$ , then  $\lfloor n/3 \rfloor$  guards are not sufficient as shown in Figure 4. This figure also shows that  $\lfloor n/2 \rfloor - 1$  guards are necessary. It has been proved by Hernandez-Penalver [11] that  $\lfloor n/2 \rfloor - 1$  guards are not only necessary but also sufficient. If  $P$  contains  $h$  holes, we know that  $\lfloor n+h/3 \rfloor$  stationary guards are sufficient and sometimes necessary for the standard art gallery problem [14] (see Figure 5). If the guards also have to satisfy the visibility constraint between them as stated previously, then  $\lfloor n+h/3 \rfloor$  guards are not sufficient as shown in Figure 6. We feel that  $\lfloor n+2h/3 \rfloor$  guards are sufficient for this problem.

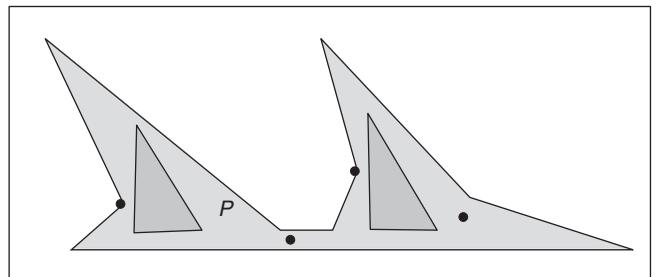
A watchman route in a polygon is a polygonal path such that every point of the polygon is visible from some point on the path [13], [14]. It can be seen that the path of a robot produced by our exploration algorithm is a watchman route inside  $P$ . This path can also be used as an inspection path for autonomous inspection of subsequent traversal [4], [5]. Note that after the viewing points are chosen by our exploration algorithm, the length of the inspection path can be made shorter by connecting these viewing points through shorter paths [5], [15].

## An Exploration Algorithm

In this section, we describe an algorithm that a point robot can use to explore an unknown polygonal environment and guarantee that the entire free space  $\mathcal{F}$  has been seen by the robot [10]. We show that the algorithm computes at most  $r+1$  visibility polygons, where  $r$  is the total number of



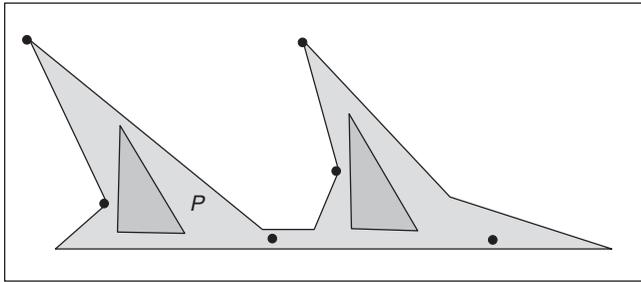
**Figure 4.** Two more guards are required to ensure visibility between guards.



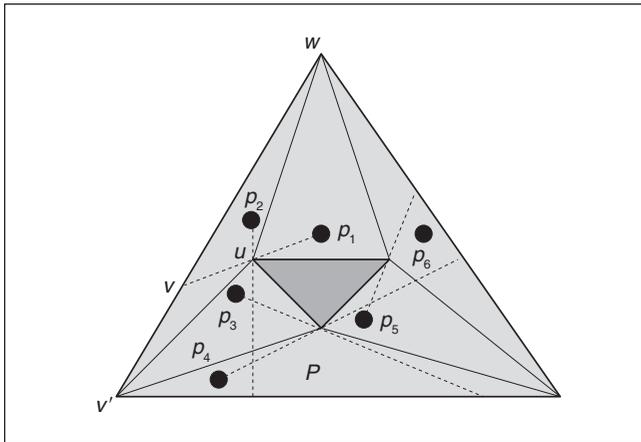
**Figure 5.** Four guards can see the entire  $P$ , but they are not mutually visible.

reflex vertices in  $P$  containing  $h$  holes. We also show that  $r + 1$  visibility polygons are sometimes necessary to see the entire  $\mathcal{F}$ .

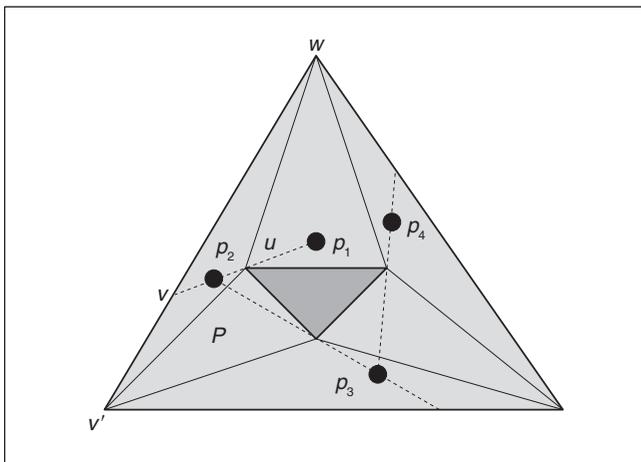
The algorithm proceeds as follows. The robot starts at any initial location,  $p_1$ , where it determines  $VP(P, p_1)$ . Using the visible vertices of  $P$  in  $VP(P, p_1)$ , the robot triangulates as much of the  $VP(P, p_1)$  as possible. Let this triangulation be denoted  $T_1$ . The robot then executes a forward move to the next viewing point  $p_2$ . For safety reasons, forward moves are always restricted to the current visibility polygon. Then, it computes the next visibility polygon  $VP(P, p_2)$ . The region common to



**Figure 6.** Two more guards are required to ensure visibility between guards.



**Figure 7.** The algorithm needs  $n + 2h - 2$  views; i.e., six views.



**Figure 8.** The algorithm needs  $r + 1$  views; i.e., four views.

## The robot must be able to see, with its onboard vision sensors, all points in the building's interior while following its exploration path.

$VP(P, p_2)$  and  $T_1$  is removed. The remaining free space in  $VP(P, p_2)$  is triangulated, and the total triangulation  $T_2$  is updated. So,  $T_2$  represents the map of  $\mathcal{F}$  explored so far.

To describe the algorithm in more detail, assume that the robot is beginning the  $i$ th step of the exploration procedure. Let  $T_{i-1}$  denote the cumulative triangulation that has been established prior to step  $i$ . Let  $uv$  be a constructed edge of  $VP(P, p_{i-1})$ , where  $u$  is the vertex of  $P$  and  $v$  is a point on a polygonal edge of  $P$ . The next viewing point  $p_i$  can be chosen from the triangle  $uvw$  formed by the constructed edge  $uv$ , the boundary edge  $uw$  of  $T_{i-1}$ , and the partial polygonal edge  $uv$ . Figure 7 shows that  $p_2$  is chosen from the triangle  $uvw$ . If there is no constructed edge of  $VP(P, p_{i-1})$ , the robot executes a backward move. The backward move is repeated until a constructed edge  $uv$  is located for some  $VP(P, p_j)$ . A point from the triangle  $uvw$  can be chosen as  $p_i$ , as stated earlier. Note that if both  $u$  and  $v$  of a constructed edge happen to be the vertices of  $P$ , then  $w$  and  $v$  become the same and, therefore,  $p_i$  is a point on  $uw$ .

Once  $p_i$  is chosen, the robot computes  $VP(P, p_i)$  and then removes the region common to  $VP(P, p_i)$  and  $T_{i-1}$  from  $VP(P, p_i)$ . If this operation splits  $VP(P, p_i)$  in several disjoint parts, the part containing  $p_i$  is chosen as  $VP(P, p_i)$ . As a result,  $VP(P, p_i)$  contains a portion of  $\mathcal{F}$ , which is yet to be triangulated. It can be viewed as if  $VP(P, p_i)$  has been computed by treating  $T_{i-1}$  as an opaque region. From now on, whenever we refer to  $VP(P, p_i)$ , it means that  $VP(P, p_i)$  is the connected unexplored region of  $\mathcal{F}$  that is visible from  $p_i$  and contains  $p_i$ .

After computing  $VP(P, p_i)$ , the interior of  $VP(P, p_i)$  is again triangulated by connecting only the vertices of  $P$  that lie in  $VP(P, p_i)$ . The vertices of the triangulation may also include vertices from previously triangulated regions. Let  $T'_i$  denote the triangulation of the newly viewed free space in  $VP(P, p_i)$ . Hence, at the end of step  $i$ , the total free space triangulated is  $T_i = T_{i-1} \cup T'_i$ . In the following lemma, we show that  $T'_i$  is not empty.

**Lemma 1:** *There exists at least one triangle in  $T'_i$ .*

**Proof:** By definition, the next viewing point  $p_i$  lies on or outside the boundary of  $T_{i-1}$ , and it belongs to a previously computed visibility polygon. Without loss of generality, we assume that  $p_i$  is a point of  $VP(P, p_{i-1})$ . We know that  $p_i$  is chosen from the partial visible triangle  $uvw$  formed by a constructed edge  $uv$  of  $VP(P, p_{i-1})$ , the corresponding boundary edge  $uw$  of  $T_{i-1}$ , and the partial polygonal edge  $uv$ . Figures 7 and 8 show that  $p_2$  is chosen from the triangle  $uvw$ . So, the boundary of  $VP(P, p_i)$  consists of the edge  $uw$  and a chain of polygonal and constructed edges connecting  $u$  and  $w$ . Since

this chain makes a turn of  $180^\circ$  or more with respect to  $p_i$ , there exists a vertex  $v'$  of  $P$  on the chain such that  $uv'w$  is a triangle inside  $VP(P, p_i)$ . Hence,  $uv'w$  belongs to  $T'_i$ . The same argument holds if  $p_i \in uw$ . ■

**Corollary 1:** *The viewing point  $p_i$  belongs to  $T'_i$  and  $VP(P, p_i)$  removes at least one constructed edge of  $VP(P, p_{i-1})$ .*

The above lemma suggests that every time a view is taken, at least one new triangle is explored. From the acquired view  $VP(P, p_i)$ , the robot establishes a list of constructed edges  $(C_{i,1}, C_{i,2}, \dots, C_{i,c_i})$  on the boundary of the current visibility polygon. These edges help to define the partial visible triangle  $uvw$  as stated earlier. Observe that the choice of the viewing point inside such triangles can play a major role in deciding the number of visibility polygons that are required to see the unexplored free space as shown in Figures 7 and 8. Therefore, for subsequent forward exploratory moves, choose a point  $z_{i,j}$  ( $j = 1, \dots, c_i$ ) on  $C_{i,j}$ . Choose one of the  $z_{i,j}$ , for example,  $z_{i,1}$ , to be the next viewing point,  $p_{i+1}$ , and recursively apply this procedure. Hence, the algorithm is a depth-first search of the unexplored free space. When the entire unexplored territory associated with  $z_{i,1}$  has been explored, then choose another viewing point at level  $i$ , say  $z_{i,2}$ , and continue. Note that while choosing  $z_{i,2}$  as the next viewing point, the corresponding constructed edge  $C_{i,2}$  must lie, partially or totally, on or outside the boundary of the free space triangulated so far (called an unexplored constructed edge). Otherwise, it is considered that  $C_{i,2}$  has been explored. The algorithm terminates when all the constructed edges of  $VP(P, p_1)$  have been explored recursively. In the following lemma, we show that the algorithm has explored  $\mathcal{F}$  completely.

**Lemma 2:** *When all the constructed edges of  $VP(P, p_1)$  have been explored recursively, the algorithm has explored the entire free space  $\mathcal{F}$ .*

**Proof:** Assume on the contrary that all the constructed edges of  $VP(P, p_1)$  have been explored recursively, but there exists a point  $z \in \mathcal{F}$ , which has not been seen by the algorithm. Consider a path  $Q$  inside  $P$  connecting  $p_1$  to  $z$ . If no such path exists, then  $z$  does not lie in the free space  $\mathcal{F}$ , which contradicts the assumption that  $z \in \mathcal{F}$ . So, we assume that such a path  $Q$  exists. Since  $p_1$  lies in the triangulated region, starting from  $p_1$ ,  $Q$  must intersect at least one boundary edge  $uv$  of the triangulation before reaching  $z$ . It means that there exists an unexplored constructed edge bounded by  $u$  or  $v$ , which contradicts the fact that all the constructed edges of  $VP(P, p_1)$  have been explored recursively. Therefore, the entire path  $Q$  lies inside the triangulation of  $P$ . Hence,  $z$  has been seen by the algorithm. ■

In the following, we present the major steps of the exploration algorithm for computing the set  $S$  of viewing points.

**Step 1:**  $i := 1$ ;  $T(P) := \emptyset$ ;  $S := \emptyset$ ; Let  $p_1$  denote the starting position of the robot.

**Step 2:** Compute  $VP(P, p_i)$ ; Construct the triangulation  $T'(P)$  of  $VP(P, p_i)$ ;  $T(P) := T(P) \cup T'(P)$ ;  $S := S \cup p_i$ ;

**Step 3:** While  $VP(P, p_i) - T(P) = \emptyset$  and  $i \neq 0$  then  $i := i - 1$ ;

**Step 4:** If  $i = 0$  then goto Step 7;

**Step 5:** If  $VP(P, p_i) - T(P) \neq \emptyset$  then choose a point  $z$  on any constructed edge of  $VP(P, p_i)$  lying outside  $T(P)$ ;

**Step 6:**  $i := i + 1$ ;  $p_i := z$ ; goto Step 2;

**Step 7:** Output  $S$  and  $T(P)$ ;

**Step 8:** Stop.

In the following lemma, we prove the upper bound on the number of views that may be required by our exploration algorithm to see the entire free space.

**Lemma 3:** *The exploration algorithm computes at most  $r + 1$  views, where  $r$  is the total number of reflex vertices in  $P$ .*

**Proof:** Let  $u_1v_1, u_2v_2, \dots, u_jv_j$  be the constructed edges generated by the algorithm during exploration. Since one endpoint of every constructed edge is a reflex vertex, we assume that  $u_1, u_2, \dots, u_j$  are reflex vertices. If  $u_1, u_2, \dots, u_j$  are different vertices, then  $j \leq r$ . So, the exploration algorithm can take at most  $r$  views after the initial view at the starting position. Consider the other situation when  $u_1, u_2, \dots, u_j$  are not different vertices. Assume that  $u_1, u_2, \dots, u_{i-1}$  are different vertices, and  $u_i$  is the same as  $u_k$  where  $i < j$  and  $1 \leq k \leq i - 2$ . We know that the algorithm will choose a viewing point on  $u_i v_i$  before any viewing point is chosen on  $u_k v_k$ . Let  $T(P)$  denote the region of the free space triangulated so far by the algorithm before a viewing point (say,  $z$ ) is chosen on  $u_i v_i$ . Note that  $u_1, u_2, \dots, u_{i-1}$  are vertices of  $T(P)$ . Since the exploration algorithm uses the depth-first search method, the algorithm explores the entire free space recursively lying outside  $T(P)$  that can be reached from  $z$ . Let  $T'(P)$  be the triangulated region of the newly explored region starting from  $z$ . Since  $u_i v_i$  and  $u_k v_k$  share the same vertex by assumption, there exists a path in the free space from  $z$  to every point of  $u_k v_k$  and, therefore, entire  $u_k v_k$  lies inside  $T(P) \cup T'(P)$ . So, at most one viewing point can be chosen by the exploration algorithm from those constructed edges that share the same reflex vertex. Hence, the exploration algorithm can take at most  $r$  views after the initial view at the starting position.

The correctness of the exploration algorithm and the completeness of the exploration follow from Lemma 1 and Lemma 2, respectively. Lemma 3 provides the upper bound on the computational complexity of the exploration. We summarize the result in the following theorem. ■

**Theorem 1:** *The exploration algorithm correctly explores the entire polygonal environment by computing at most  $r + 1$  views, where  $r$  is the total number of reflex vertices in  $P$ .*

Let us now compare the performance of our algorithm with that of an optimal exploration algorithm. Consider a spiral polygon without any hole as shown in Figure 9. It can be seen from the figure that no exploration algorithm, starting from  $p_1$ , can explore the entire spiral polygon in less than  $r + 1$  views. Figure 10 also shows that  $r + 1$  views are necessary to explore the entire polygon, which contains one hole. On the

## There are several online computational geometry algorithms for searching or exploring unknown polygons with or without holes.

other hand, consider a star-shaped polygon without any hole as shown in Figure 11. In this figure, two views are enough to see the entire star-shaped polygon because the robot takes the first view at the given starting position  $p_1$ , and then it takes the second view from the star-point  $p_2$ . On the other hand, our algorithm first takes a view from  $p_1$  and, then, takes  $r$  views to remove all constructed edges of  $VP(P, p_1)$ . This example shows that this is the worst performance of our algorithm (i.e., the competitive ratio is  $(r + 1)/2$ ) with respect to the performance of an optimal exploration algorithm. We have the following theorem.

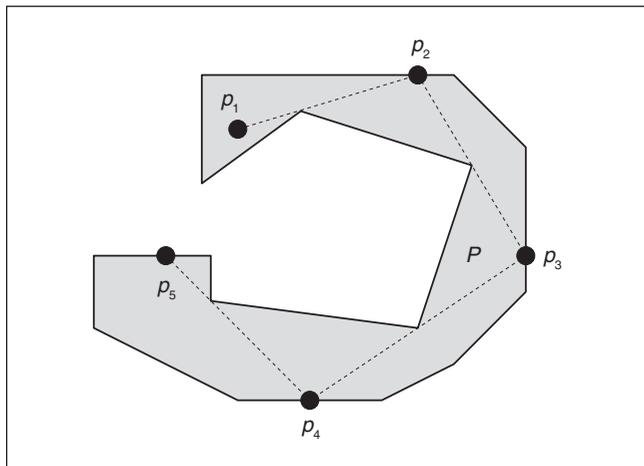


Figure 9. Any exploration algorithm needs at least  $r + 1$  views for exploring the spiral polygon.

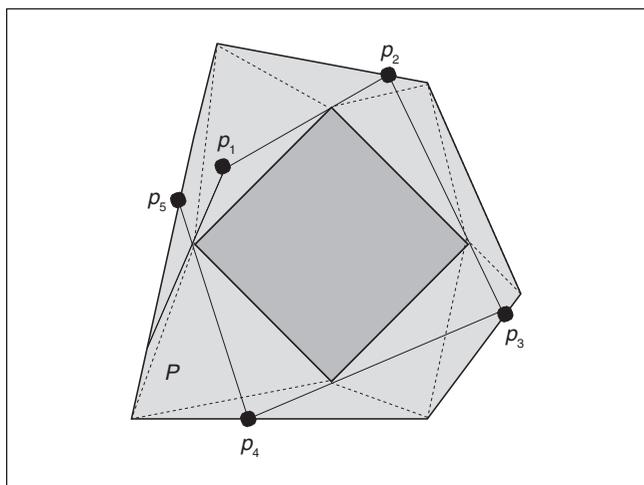


Figure 10. Any exploration algorithm needs at least  $r + 1$  views to explore the polygon with one hole.

**Theorem 2:** The competitive ratio of the exploration algorithm is  $(r + 1)/2$ , where  $r$  is the total number of reflex vertices in  $P$ .

## An Exploration Algorithm with Range Restriction

In this section, we present our exploration algorithm by restricting the visibility polygons by a range distance  $R$  (see Figure 12) and establish an upper bound for its competitive ratio [2]. This algorithm follows the same approach as in the previous algorithm. We start with the following observation.

**Theorem 3:** Let  $D$  be the longest line segment that can lie completely inside a polygonal environment  $P$ . If  $R \geq D$ , then the exploration algorithm in the previous section can be used to explore  $P$  using restricted visibility polygons.

**Proof:** The proof follows from the fact that if  $R \geq D$  any visibility polygon computed by the exploration algorithm is the same as the restricted visibility polygon. ■

Let us consider the other case when  $R < D$ . Let  $RVP(z)$  denote the restricted visibility polygon computed from a point  $z$ . Observe that a restricted visibility polygon may not

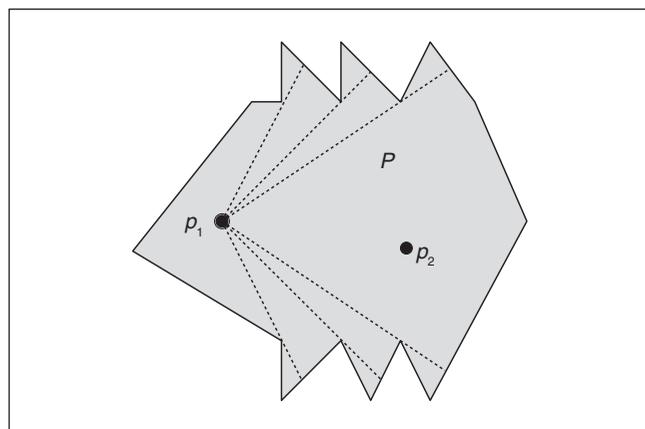


Figure 11. The star-shaped polygon can be explored in two views.

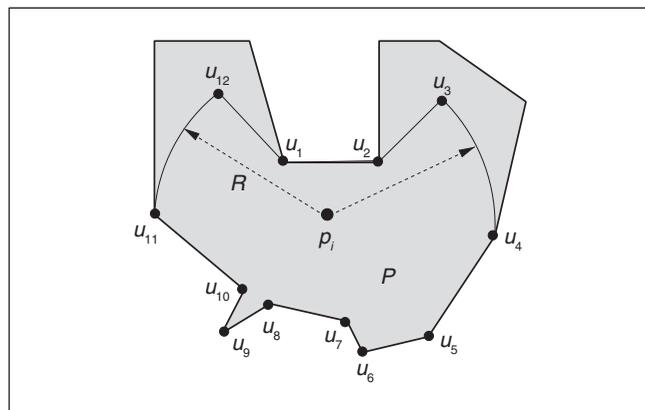


Figure 12. Vertices of restricted visibility polygon from  $p_i$  with range  $R$  are  $u_1, u_2, \dots, u_{12}$ .

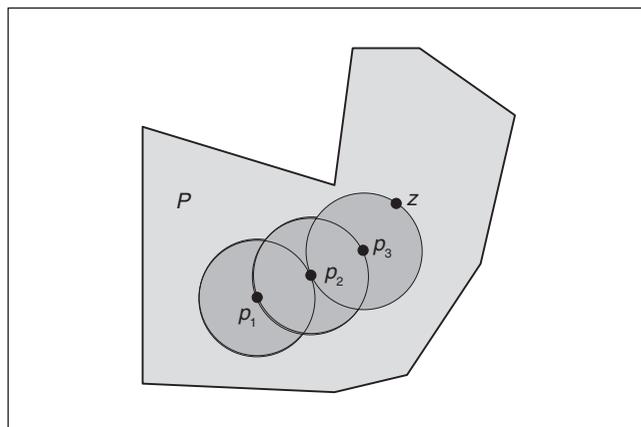
be closed (see Figure 13), and its boundary can have circular arcs in addition to constructed and polygonal edges. So, it is necessary to take another view from some point in the current restricted visibility polygon to see more of  $P$ . The process can be repeated until the union of these restricted visibility polygons covers  $P$ .

In the following, we present an algorithm that a point robot can use to explore  $P$  by starting from an internal point  $p_1$  and using restricted visibility polygons (see Figure 13). Our algorithm is somewhat similar to the Bug1 algorithm [3], [12] and the coverage algorithm [1], where a point robot moves straight to the boundary of  $P$  and then follows the boundary of  $P$ . However, our algorithm chooses viewing points arbitrarily close to the boundary of  $P$  rather than on the polygonal edges to avoid collision. In the sequel, whenever a viewing point is said to be on a polygonal edge, it implies that the viewing point is arbitrarily close to the polygonal edge.

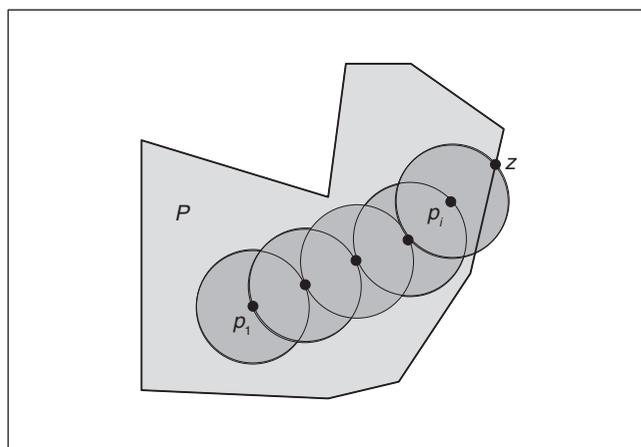
Let  $CP_i$  denote the region of  $P$  so far visible from the robot. The robot initializes its polar coordinate system by setting its origin at  $p_1$ . So, the coordinates of the boundary points of  $CP_i$  are with respect to  $p_1$ . Initially, all edges of  $P$  are unmarked to indicate that they are not yet explored by the robot. Starting from  $p_1$ , the algorithm chooses viewing points on circular arcs until a point  $u$  on the boundary of  $P$  is visible from the robot. Assume that  $u$  belongs to a hole  $H_i$  in  $P$ . Then, the robot chooses viewing points along the boundary of  $H_i$  until all edges of  $H_i$  are visible from the robot. Similarly, the robot explores the remaining holes and the outer boundary of  $P$  one by one. Finally, the algorithm terminates when the entire free space of  $P$  is explored.

- Step 1:** Compute  $RVP(p_1)$ ;  $i := 1$ ;  $CP_i := RVP(p_i)$ ; If the boundary of  $CP_i$  consists of only polygonal edges then goto Step 8;
- Step 2:** While the boundary of  $CP_i$  consists of only circular arcs do (see Figure 13)
- Step 2a:** Choose a point  $z$  on any circular arc of  $CP_i$ ;  $i := i + 1$ ;  $p_i := z$ ;
- Step 2b:** Compute  $RVP(p_i)$ ;  $CP_i := CP_{i-1} \cup RVP(p_i)$ ;
- Step 3:** If  $p_i$  is not a point of a polygonal edge do (see Figure 14)
- Step 3a:** Let  $z$  be the furthest point of  $p_i$  on the boundary of  $CP_i$  that belongs to a polygonal edge;  $i := i + 1$ ;  $p_i := z$ ;
- Step 3b:** Compute  $RVP(p_i)$ ;  $CP_i := CP_{i-1} \cup RVP(p_i)$ ;
- Step 3c:** Mark those edges of  $P$  that are totally inside  $CP_i$ ;
- Step 4:** While  $p_i$  is a point of an unmarked polygonal edge do (see Figure 15)
- Step 4a:** Starting from  $p_i$ , traverse the boundary of  $RVP(p_i) - CP_{i-1}$  along polygonal edges until a point  $z$  is located, which is a starting point of a circular arc or a constructed edge;  $i := i + 1$ ;  $p_i := z$ ;
- Step 4b:** Compute  $RVP(p_i)$ ;  $CP_i := CP_{i-1} \cup RVP(p_i)$ ;
- Step 4c:** Mark those edges of  $P$  that are totally inside  $CP_i$ ;
- Step 5:** If the boundary of  $CP_i$  contains a part  $uu'$  of a polygonal edge then (see Figure 16)
- Step 5a:** Take a point  $z$  from  $uu'$ ;  $i := i + 1$ ;  $p_i := z$ ; goto Step 4;
- Step 6:** While the boundary of  $CP_i$  consists of only circular arcs and marked edges do (see Figure 17)

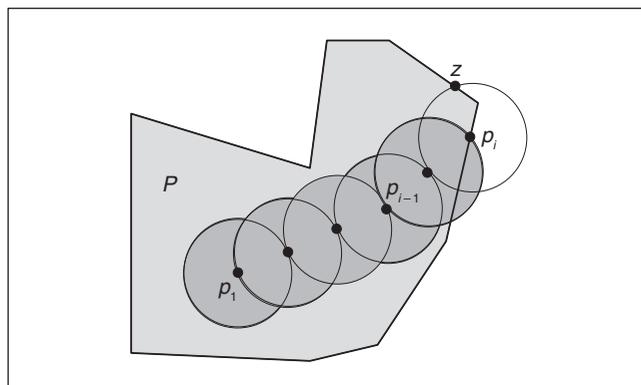
- Step 6a:** Choose a point  $z$  on any circular arc of  $CP_i$ ;  $i := i + 1$ ;  $p_i := z$ ;
- Step 6b:** Compute  $RVP(p_i)$ ;  $CP_i := CP_{i-1} \cup RVP(p_i)$ ;
- Step 6c:** Mark those edges of  $P$  that are totally inside  $CP_i$ ;
- Step 7:** If there is a circular arc or a constructed edge on the boundary of  $CP_i$  then goto Step 3;
- Step 8:** Report viewing points  $p_1, p_2, \dots, p_i$  and Stop.



**Figure 13.** Restricted visibility polygons are computed in  $P$  starting from the initial position  $p_1$ .



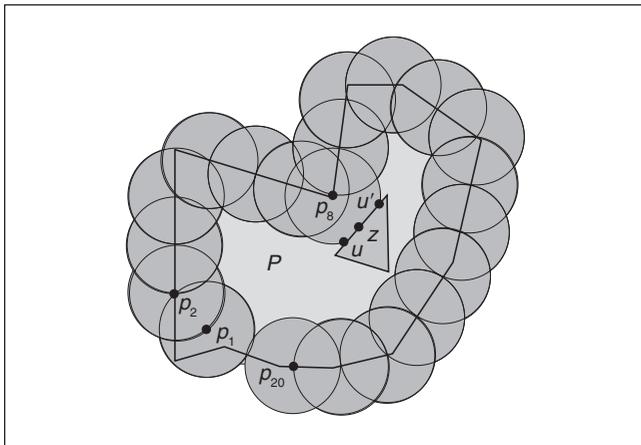
**Figure 14.** The current restricted visibility polygon has intersected a polygonal edge for the first time.



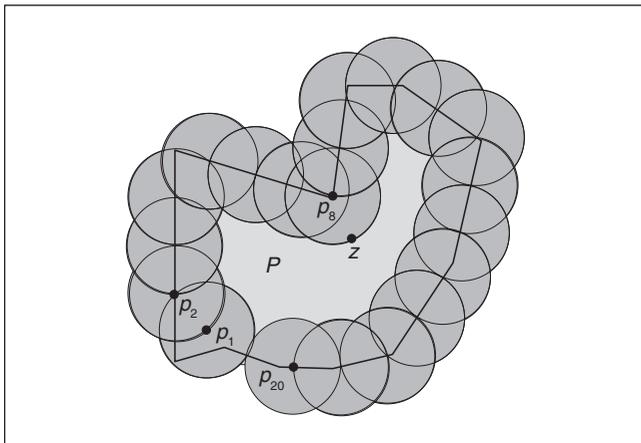
**Figure 15.** The robot moves along the boundary of  $P$  from  $p_i$  until the next viewing point  $z$  is located.

# It is open whether an upper bound on the number of views for exploring $P$ under translation can be derived for a convex robot.

Let us prove the correctness of the exploration algorithm. We show that  $CP_i = P$  when the algorithm terminates. Assume on the contrary that  $P \not\subset CP_i$ . Then, there exists a point  $p \in P$  and  $p \notin CP_i$ . If there exists a path from  $p$  to  $p_1$  lying inside  $CP_i$ , then,  $p$  belongs to  $CP_i$ , which is a contradiction. Otherwise, any path from  $p$  to  $p_1$  must intersect the boundary of  $CP_i$ . Since every edge on the boundary of  $CP_i$  is a polygonal edge at the time of termination, every path from  $p$  to  $p_1$  must intersect an edge of  $P$ . So,  $p$  does not belong to  $P$ , which is a contradiction. Hence,  $CP_i = P$  when the algorithm terminates. We have the following theorem.



**Figure 16.** The robot moves to a point  $z$  of  $uu'$  to explore all edges of that hole.



**Figure 17.** All boundary edges of  $P$  are already explored (i.e., marked).

**Theorem 4:** Using restricted visibility polygons, the exploration algorithm explores the entire polygon  $P$  correctly.

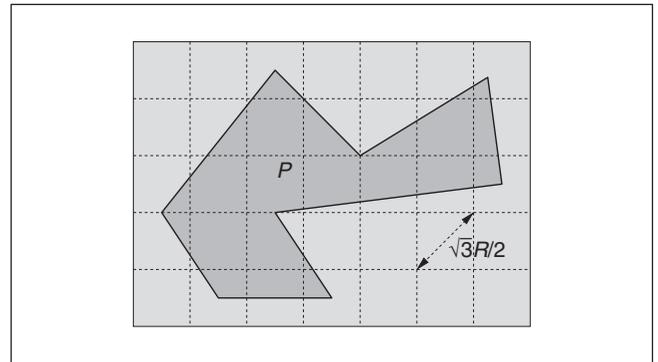
In the following lemma, we establish an upper bound on the number of views required by a robot to explore the region  $P$  using this exploration algorithm.

**Lemma 4:** If the area, perimeter, number of holes, and number of reflex vertices of  $P$  are  $\text{Area}(P)$ ,  $\text{Perimeter}(P)$ ,  $h$  and  $r$ , respectively, then the number of viewing points required by the exploration algorithm is bounded by

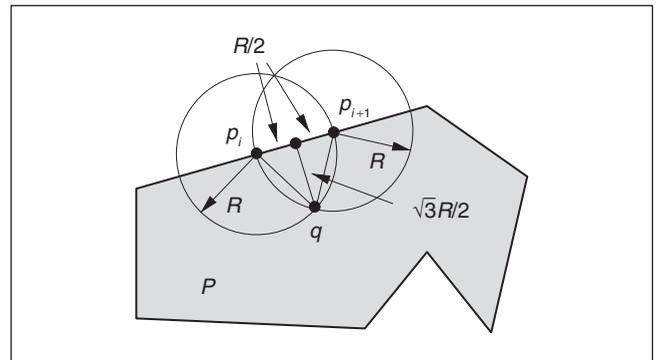
$$\left\lceil \frac{8 \times \text{Area}(P)}{3 \times R^2} \right\rceil + \left\lceil \frac{\text{Perimeter}(P)}{R} \right\rceil + r + h + 1.$$

**Proof:** Place  $P$  on a grid where the diagonals of squares is of length  $\sqrt{3}R/2$  (see Figure 18). Observe that if the robot takes a view inside a square, it can cover the entire square as every point of the square is within the distance of  $R$  from the viewing point. So, the number of squares lying partially and totally inside  $P$  gives an upper bound on the number of views required. ■

Consider the squares that lie totally inside  $P$ . We know that the area of a square is  $(\sqrt{3}R/2\sqrt{2}) \times (\sqrt{3}R/2\sqrt{2}) = 3 \times R^2/8$ . Clearly the number of such squares in  $P$  is at most  $\text{Area}(P)/(3 \times R^2/8) = 8 \times \text{Area}(P)/3 \times R^2$ . Since the robot can take at most one view in each square,  $8 \times \text{Area}(P)/3 \times R^2$  views can be taken from such squares in  $P$ .



**Figure 18.**  $P$  is placed on a grid, where the length of diagonals of squares is  $\sqrt{3}R/2$ .



**Figure 19.** Viewing points  $p_i$  and  $p_{i+1}$  are on the same edge of  $P$ .

Consider the squares that are intersected by edges of  $P$ . Instead of counting the number of squares intersected by the edges of  $P$ , we show that such intersected squares are covered by views that are taken along the boundary of  $P$ . Let  $p_i$  and  $p_{i+1}$  be two consecutive viewing points on the boundary of  $P$ . Assume that both  $p_i$  and  $p_{i+1}$  lie on the same edge of  $P$  (see Figure 19). We know that the distance between them is  $R$  if they are intermediate points on the edge. If the distance between them is less than  $R$ , then  $p_{i+1}$  is a reflex (see Step 4). Consider the case where  $p_i$  and  $p_{i+1}$  are intermediate points on the same edge. Draw two circles of radius  $R$  with  $p_i$  and  $p_{i+1}$  as centers. It can be seen that any point  $q$ , at a distance of at most  $\sqrt{3}R/2$  for the segment  $p_i p_{i+1}$ , lies within one of the two circles. On the other hand, every point of any square intersected by the segment  $p_i p_{i+1}$  lies at a distance of at most  $\sqrt{3}R/2$ . Therefore, all points of the squares intersected by the segment  $p_i p_{i+1}$  lie within  $RVP(p_i)$  or  $RVP(p_{i+1})$ . The same arguments hold if  $p_{i+1}$  is a reflex vertex, or  $p_i$  and  $p_{i+1}$  do not belong to the same edge (see Figure 20). Hence, all points in the squares intersected by edges of  $P$  can be seen by at most  $\lfloor \text{Perimeter}(P)/R \rfloor + r$  views, where  $r$  represents the additional views taken at reflex vertices.

Now, consider the view that sees the boundary of  $P$  for the first time (see Step 3). If the view is taken from a square intersected by an edge of  $P$ , then we have to add 1 to the bound. Since there can be one such view for every hole and for the outer boundary of  $P$ , there can be  $h + 1$  additional views in the squares intersected by edges of  $P$ . Hence, the maximum number of views that the robot can take to explore  $P$  is bounded by  $\lfloor (8 \times \text{Area}(P)) / (3 \times R^2) \rfloor + \lfloor \text{Perimeter}(P)/R \rfloor + r + h + 1$ .

Let us derive the competitive ratio of the exploration algorithm. Since a robot can see in each view at most  $\pi R^2$  area of  $P$ , any exploration algorithm must take at least  $\lceil \text{Area}(P) / \pi R^2 \rceil$  views to see the entire  $P$ . Hence, the competitive ratio is

$$\frac{\left\lfloor \frac{8 \times \text{Area}(P)}{3 \times R^2} \right\rfloor + \left\lfloor \frac{\text{Perimeter}(P)}{R} \right\rfloor + r + h + 1}{\left\lceil \frac{\text{Area}(P)}{\pi R^2} \right\rceil},$$

which is upper bounded by

$$\left\lceil \frac{8\pi}{3} + \frac{\pi R \times \text{Perimeter}(P)}{\text{Area}(P)} + \frac{(r + h + 1) \times \pi R^2}{\text{Area}(P)} \right\rceil.$$

It can be seen that the worst case arises when the number of reflex vertices is large for a given  $P$  and  $R$ . On the other hand, if  $R$  is sufficiently large, the number of views required is  $r + h + 1$ . We have the following theorem.

**Theorem 5:** *The competitive ratio of the exploration algorithm for a point robot with restricted visibility polygons is bounded by*

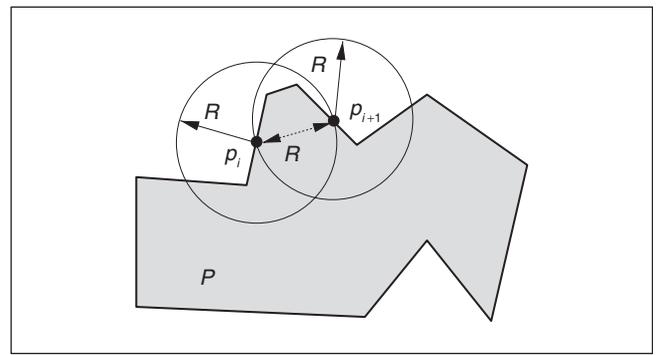
$$\left\lceil \frac{8\pi}{3} + \frac{\pi R \times \text{Perimeter}(P)}{\text{Area}(P)} + \frac{(r + h + 1) \times \pi R^2}{\text{Area}(P)} \right\rceil.$$

## Conclusions

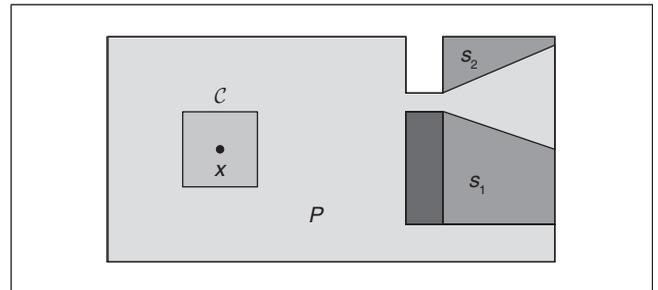
Our exploration algorithm for a point robot in the “An Exploration Algorithm” section chooses the next viewing point on

an unexplored constructed edge. It can be seen from Figure 8 that the viewing point  $p_4$  has been chosen, as the triangle containing  $p_4$  is not totally visible from  $p_1$  or  $p_3$ . However, this triangle is totally visible jointly from  $p_1$  and  $p_3$ . This observation suggests that if partially visible triangles associated with unexplored constructed edges are taken into consideration while triangulating the current visibility polygon, it may be possible to reduce the number of viewing points by one for every hole that gives the tighter bound  $r + 1 - h$ . However, it is not clear how to combine these partially visible triangles correctly to generate a triangulation connecting only the vertices of the polygon.

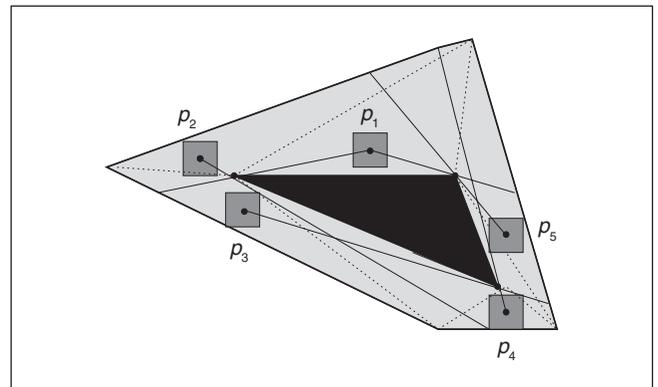
Suppose we wish to design an algorithm that a convex robot  $\mathcal{C}$  can use to explore an unknown polygonal environment  $P$  (under translation) following a similar strategy of the point robot as stated in the “An Exploration Algorithm” section: let  $x$  be the point of  $\mathcal{C}$  corresponding to the position of the visual sensor of the robot (see Figure 21). This means that the region



**Figure 20.** Viewing points  $p_i$  and  $p_{i+1}$  are on different edges of  $P$ .



**Figure 21.** Two parts  $s_1$  and  $s_2$  of  $P$  cannot be explored by  $\mathcal{C}$ .



**Figure 22.** More than  $r + 1$  views are required for exploration.

visible from the current position of  $\mathcal{C}$  is always the visibility polygon computed from  $x$ . Observe that since all points of  $P$  may not be reachable by  $\mathcal{C}$ , some parts of  $P$  may remain unexplored as shown in Figure 21.

Let  $p_1$  be the starting position of  $x$  (see Figure 22). Initially,  $\mathcal{C}$  computes  $VP(P, p_1)$ . Then, it computes the Minkowski sum of  $C$  and  $VP(P, p_1)$  under translation of  $C$ , taking  $x$  as the reference point [6], [7]. From the Minkowski sum, constructed edges of  $VP(P, p_1)$  are located that can be touched by  $C$ , which can be called *eligible* constructed edges. Then,  $\mathcal{C}$  moves toward an eligible constructed edge until it touches some point on that edge. Let  $p_2$  be the corresponding position of  $x$ . Then,  $\mathcal{C}$  computes  $VP(P, p_2)$ . Using this strategy,  $\mathcal{C}$  can explore  $P$  but that requires more than  $r + 1$  views as shown in Figure 22, because the same reflex vertex is an endpoint of more than one eligible constructed edge. So, the upper bound on the number of views for a convex robot is more than  $r + 1$ . It is open whether an upper bound on the number of views for exploring  $P$  under translation can be derived for a convex robot.

## Acknowledgments

The extended abstract of this article was reported in two parts [2]. In [10], we claimed that the point robot computes at most  $r + 1 - h$  views to explore the entire free space. However, it has been pointed out that our bound mentioned in [10] is not correct. We have now modified the algorithm by positioning the next viewing point on a constructed edge. This restriction ensures that the algorithm computes at most  $r + 1$  views.

The authors would like to thank Sudeb Prasant Pal and Partha Goswami for their helpful comments.

## Keywords

Polygon exploration, online algorithms, point robot, competitive ratio, art gallery problem, discrete visibility, visibility polygon.

## References

- [1] E. U. Acar and H. Choset, "Sensor-based coverage of unknown environments: Incremental construction of morse decompositions," *Int. J. Robot. Res.*, vol. 21, no. 4, pp. 345–366, 2002.
- [2] A. Bhattacharya, S. K. Ghosh, and S. Sarkar, "Exploring an unknown polygon environment with limited visibility," in *Proc. Int. Workshop Computational Geometry and Applications*, May 2001, pp. 640–648.
- [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*. Cambridge, MA: MIT Press, 2005.
- [4] P. Colley, H. Meijer, and D. Rappaport, "Motivating lazy guards," in *Proc. 7th Canadian Conf. Computational Geometry*, 1995, pp. 121–126.
- [5] T. Danner and L. E. Kavraki, "Randomized planning for short inspection paths," in *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, 2000, pp. 971–976.
- [6] P. K. Ghosh, "A solution of polygon containment, spatial planning, and other related problems using minkowski operations," *Comput. Vision, Graphics Image Process.*, vol. 49, no. 1, pp. 1–35, 1990.
- [7] P. K. Ghosh, "A unified computational framework for minkowski operations," *Comput. Graph.*, vol. 17, no. 4, pp. 357–378, 1993.
- [8] S. K. Ghosh, "Approximation algorithms for art gallery problems," in *Proc. Canadian Information Processing Society Congr.*, 1987, pp. 429–434.

- [9] S. K. Ghosh, *Visibility Algorithms in the Plane*. Cambridge, UK: Cambridge Univ. Press, 2007.
- [10] S. K. Ghosh and J. W. Burdick, "An online algorithm for exploring an unknown polygonal environment by a point robot," in *Proc. 9th Canadian Conf. Computational Geometry*, 1997, pp. 100–105.
- [11] G. Hernandez-Penalver, "Controlling guards," in *Proc. 6th Canadian Conf. Computational Geometry*, 1994, pp. 387–392.
- [12] V. Lumelsky and A. Stepanov, "Path planning strategies for point automation moving amidst unknown obstacles of arbitrary shape," *Algorithmica*, vol. 2, no. 1, pp. 402–430, 1987.
- [13] B. J. Nilsson, "Guarding art galleries—Methods for mobile guards," Ph.D. dissertation, Lund Univ., Sweden, 1994.
- [14] J. Urrutia, "Art gallery and illumination problems," *Handbook of Computational Geometry*, J.-R. Sack and J. Urrutia, Eds. Amsterdam, The Netherlands: North-Holland, 2000, pp. 973–1023.
- [15] P. Wang, "View planning with combined view and travel cost," Ph.D. dissertation, Simon Fraser Univ., Canada, 2007.

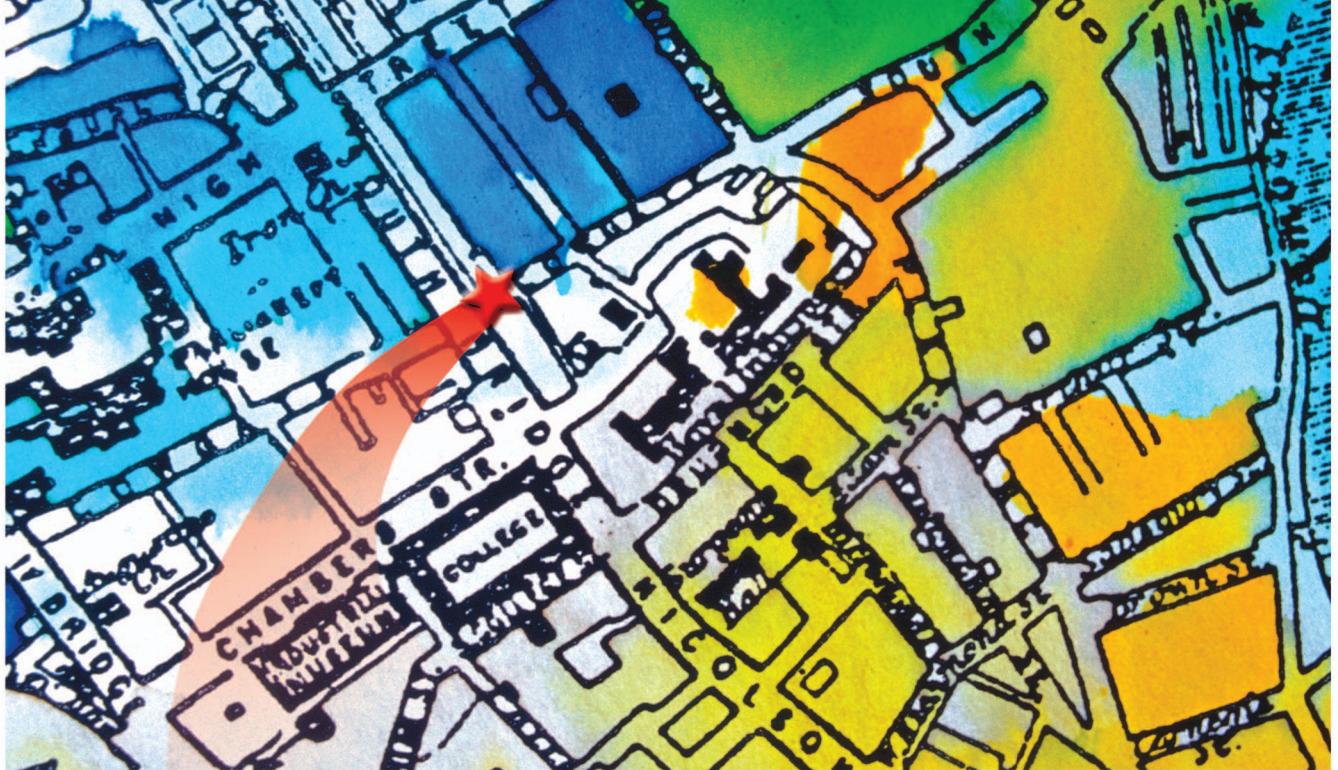
**Subir Kumar Ghosh** is a professor of computer science at the Tata Institute of Fundamental Research, Mumbai, India, and he is a fellow of the Indian Academy of Sciences. He has authored several papers on computational geometry, robot path planning, and geometric graph theory, and he has also written a book *Visibility Algorithms in the Plane*, which was published by Cambridge University Press in 2007. He has worked as a visiting scientist in many reputable universities and research institutes around the world.

**Joel Wakeman Burdick** is a professor in the departments of mechanical engineering and bioengineering at the California Institute of Technology. His current research interests include sensor-based robot motion planning, robotic grasping and fixturing, neural prosthetics, and rehabilitation of spinal cord injuries. He is a past recipient of the NSF Presidential Young Investigator award, the Office of Naval Research Young Investigator award, and the Feynman fellowship. He has been a finalist for the best paper award in the IEEE International Conference on Robotics and Automation in 1993, 1999, 2000, and 2005, and he was appointed an IEEE Robotics and Automation Society Distinguished Lecturer in 2003.

**Amitava Bhattacharya** is currently a research assistant professor in the Department of Mathematics, Statistics, and Computer Science, at the University of Illinois, Chicago. His main area of research is polyhedral combinatorics. He has published papers in many different areas of discrete mathematics.

**Sudeep Sarkar** is a professor in computer science and engineering at the University of South Florida, Tampa, where he is an Ashford Distinguished Scholar. His research interests include computer and robot vision, biometrics, and nano-computing. He is the coauthor of the book *Computing Perceptual Organization in Computer Vision*, which was published by World Scientific in 1994. He is also the coeditor of the book *Perceptual Organization for Artificial Vision Systems*, which was published by Kluwer Publishers in 2000.

**Address for Correspondence:** Subir Kumar Ghosh, School of Computer Science, Tata Institute for Fundamental Research, Mumbai 400005, India. E-mail: ghosh@tifr.res.in.



©ARTVILLE

# Probabilistic Self-Localization and Mapping

## *An Asynchronous Multirate Approach*

BY LEOPOLDO ARMESTO, GIANLUCA IPPOLITI,  
SAURO LONGHI, AND JOSEP TORNERO

One of the most common techniques for state estimation of nonlinear discrete-time dynamic systems is the extended Kalman filter (EKF) [1] or, more recently, unscented Kalman filter (UKF) [2]. Kalman filter gives a robust, optimal, and recursive state estimation to fuse redundant sensor information. However, both approaches assume that the probability distribution function (PDF) is Gaussian, which is not true for most signals found in practice. Other recent filtering methods are particles filters (PFs) [3], [4] in which the main advantage is that the PDF can be accurately approximated with a large number of particles. The most common approach of PF is the

sampling importance resampling [5], [6] that provides a weight (importance factor) to each particle. The resampling step selects those particles with higher weights and removes those particles with lower weights. Another well-known approach is the Rao-Blackwellized PF [7], which uses a PF for some variables of the state and a Kalman filter for other variables.

In robotics, these estimation methods are commonly used to determine the robot pose with respect to its environment (map); see, for instance, [1], [8]–[12]. The problem of determining the robot pose is commonly known as self-localization. The map can be assumed to be known or unknown. In the former, the EKF, UKF, and also the Monte Carlo localization (MCL) methods [11], [12] have been widely used. In the latter, the SLAM (simultaneous localization and map building)

Digital Object Identifier 10.1109/M-RA.2007.907355

problem arises, which consists of estimating the robot pose and map simultaneously. Classical approaches to solve this problem are based on EKF by joining the robot state and the map [1], [13], [14]. The problem of this approach is that the dimension of the covariance grows to  $(N + 3) \times (N + 3)$ , with  $N$  being the number of map features.

Recently, a solution of the SLAM problem with Rao-Blackwellized PFs has been given, which is known as FastSLAM (Factored Solution to SLAM) [15]. The key idea behind the FastSLAM is that the problem can be divided into  $N + 1$  separated problems: one for estimating the robot pose with a PF and, therefore, with low state dimension, and  $N$  low dimension separate problems for updating the map features.

One of the main contributions of this article is related to the multirate asynchronous filtering approach for the SLAM problem based on PFs. Previous multirate filter contributions are mainly for linear systems. In [16] and [17], a Kalman filter is applied for linear quadratic regulator (LQG) control, while in [18] a Kalman filter is developed using lifting techniques [19]. The problem of multirate filtering arises from the fact that sensors and actuators of robots are sampled at different sampling rates due to technological limitations, communication channels, processing time, etc.

In this article, significant improvements for robot pose estimation are obtained when introducing multirate techniques to FastSLAM. In particular, it is shown that multirate fusion aims to provide more accurate results in loop-closing problems in SLAM (localization and map building problems with closed paths).

Additionally, in this article a pose estimation algorithm based on least squares (LS) fitting of line features is proposed. Since the complexity of LS fitting is linear to the number of features, this implies a low computational cost than other techniques. Therefore, methods based on PFs such as MCL and FastSLAM that require a large number of particles may benefit from this fact. In particular, this provides an accurate approximation of the posterior PDF for FastSLAM 2.0 [20].

Moreover, we developed an asynchronous filtering method to deal with measurements of sensors at different sampling rates. One of the key ideas of the method is to use an asynchronous hold to extrapolate inputs of the system. Another key point is the asynchronous execution of prediction and update steps in the filtering method, which aims to maintain a good system performance. The prediction step is executed within at least a prespecified sampling period (generally at a fast sampling rate to reduce discretization errors), and the update step is executed only when measurements are asynchronously received.

Experimental tests have been performed on a powered wheelchair equipped with a fiber-optic gyroscope (FOG), a laser scanner, and two optical encoders connected to the axes of the driving wheels. These applications are of interest in the emerging area of assistance technologies where powered wheelchairs can be used to strengthen the residual abilities of users with motor disabilities [21]–[23].

The proposed approach results in a computationally efficient solution to the localization problem and may really represent a basic step towards the proper design of a navigation system aimed at enhancing the efficiency and the security of commercial powered wheelchairs.

## Object Detection

In this section, a method for object detection is described, based on laser ranger measurements. This method is specifically recommended for indoor applications, since it assumes certain geometric constraints about the environment.

Firstly, we define an object as a set of points  $\{x_i, y_i\}$ , representing a given landmark of the environment (walls, shelves, columns, corners, persons, etc.). In particular, for localization purposes, the map contains lines representing walls, shelves, and doors, etc. Moving objects such as persons are not included in the map, since it is assumed to be static.

### Segmentation and Single Line Fitting

Laser points are segmented using an adaptive breakpoint detector [24] and the well-known “split and merge” algorithm [25]. Initially, a standard LS fitting is used for each segmented line to estimate distance ( $\rho$ ) and angle ( $\varphi$ ) parameters, where the well-known solution is

$$\hat{\rho} = \bar{x} \cos \hat{\varphi} + \bar{y} \sin \hat{\varphi}, \quad \hat{\varphi} = \frac{1}{2} \arctan \left( \frac{-2\sigma_{xy}^2}{\sigma_y^2 - \sigma_x^2} \right),$$

where  $\bar{x}$ ,  $\bar{y}$ ,  $\sigma_x^2$ ,  $\sigma_y^2$ , and  $\sigma_{xy}^2$  are means and covariances of the dataset.

### Multiple Line Fitting

Since our object detection algorithm is designed for structured indoor environments, it is reasonable to make some assumptions about the properties of the environment. In particular, walls are usually perpendicular or parallel to each other. With that in mind, we present a method for fitting lines by considering geometric constraints of the environment.

The goal of this method is the minimization of the sum of squared distances for all lines (global multiple line fitting). This basic idea allows us to improve typically noisy estimations of lines with few points. In general, the method may not only improve parameter estimation but also data association and map building. In addition to this, corners can easily be estimated as intersection points of contiguous lines.

By performing a previous single estimation (standard LS fitting), we can select those lines whose angle difference is closed to  $\varphi_k - \varphi_0 \approx \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$  with respect to a reference line. The idea is to force the estimation to generate  $\varphi_k - \varphi_0 = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$ . The reference line should be taken to be the most reliable one; i.e., the largest line or the line with the greatest number of points. Obviously, the method is not limited to rectangular angles and any angle might be forced. If a line does not satisfy the constrained condition, it is not forced and the single line fitting estimation is used instead.

The performance index for minimizing is as follows:

$$E_{ML} = \frac{1}{2} \sum_{k=1}^{N_l} \sum_{i=1}^{N_{i,k}} (\rho_k - x_{i,k} \cos \varphi_k - y_{i,k} \sin \varphi_k)^2$$

constrained to  $\varphi_k = \varphi_0 + \varphi_{c,k}$ , where  $\varphi_{c,k}$  is the constrained angle with respect to the reference line (for convenience the

first line).  $N_l$  is the number of lines and  $N_{l,k}$  the number of points for each segmented line.

The distances of lines can be obtained as follows:

$$\frac{\partial E_{ML}}{\partial \rho_k} = 0 \Rightarrow \hat{\rho}_k = \bar{x}_k \cos \hat{\varphi}_k + \bar{y}_k \sin \hat{\varphi}_k.$$

Replacing these variables and angle constraints into the original performance index and performing some standard algebraic manipulations, we get to the optimal estimate of  $\varphi_0$ :

$$\frac{\partial E_{ML}}{\partial \varphi_0} = \tilde{B} \sin(2\hat{\varphi}_0) - \tilde{A} \cos(2\hat{\varphi}_0) = 0 \Rightarrow \tan(2\hat{\varphi}_0) = \frac{\tilde{A}}{\tilde{B}},$$

with

$$\tilde{A} = - \sum_{k=0}^{N_{l,k}} 2\sigma_{xy,k}^2 \cos(2\varphi_{c,k}) + (\sigma_{y,k}^2 - \sigma_{x,k}^2) \sin(2\varphi_{c,k}) \quad \text{and}$$

$$\tilde{B} = \sum_{k=0}^{N_{l,k}} -2\sigma_{xy,k}^2 \sin(2\varphi_{c,k}) + (\sigma_{y,k}^2 - \sigma_{x,k}^2) \cos(2\varphi_{c,k}).$$

It is interesting to remark that data statistics, that is,  $\bar{x}$ ,  $\bar{y}$ ,  $\sigma_x^2$ ,  $\sigma_{xy}^2$ , and  $\sigma_y^2$ , were initially computed for single fitting and, therefore, they are reused for saving computational resources.

#### Algorithm 1: Multiple Line Fitting

- 1 **MultipleLineFitting**( $\mathbf{r}_t, \alpha_t$ )
- 2 remove those measurements from  $\mathbf{r}_t$  and  $\alpha_t$  higher than a maximum distance;
- 3 compute break points based on discontinuities using ( $\mathbf{r}_t, \alpha_t$ );
- 4 compute Cartesian points ( $\mathbf{x}_t, \mathbf{y}_t$ ) from ( $\mathbf{r}_t, \alpha_t$ );
- 5 segment Cartesian points with Split & Merge algorithm;
- 6 estimate lines  $\mathbf{z}_t$  with LS fitting, one for each segmented set of points (single fitting);
- 7 retrieve  $\hat{\varphi}_0$  from the largest line of  $\mathbf{z}_t$ ;
- 8  $\tilde{A} = 0$ ;  $\tilde{B} = 0$ ;
- 9 **for**  $i = 1$  **to**  $\text{length}(\mathbf{z}_t)$  **do**
- 10 retrieve  $\langle \hat{\varphi}_i, \sigma_{x,i}^2, \sigma_{xy,i}^2, \sigma_{y,i}^2 \rangle$  from  $\mathbf{z}_t^i$ ;
- 11 **if**  $\hat{\varphi}_i - \hat{\varphi}_0 \simeq 0$  **then**  $\hat{\varphi}_{c,i} = \hat{\varphi}_0$ ;
- 12 **else if**  $\hat{\varphi}_i - \hat{\varphi}_0 \simeq \frac{\pi}{2}$  **then**  $\hat{\varphi}_{c,i} = \hat{\varphi}_0 + \frac{\pi}{2}$ ;
- 13 **else if**  $\hat{\varphi}_i - \hat{\varphi}_0 \simeq \pi$  **then**  $\hat{\varphi}_{c,i} = \hat{\varphi}_0 + \pi$ ;
- 14 **else if**  $\hat{\varphi}_i - \hat{\varphi}_0 \simeq \frac{3\pi}{2}$  **then**  $\hat{\varphi}_{c,i} = \hat{\varphi}_0 + \frac{3\pi}{2}$ ;
- 15 **else**  $\hat{\varphi}_{c,i} = \hat{\varphi}_i$ ;
- 16  $\tilde{A} = \tilde{A} - 2\sigma_{xy,i}^2 \cos(2\hat{\varphi}_{c,i}) - (\sigma_{y,i}^2 - \sigma_{x,i}^2) \sin(2\hat{\varphi}_{c,i})$ ;
- 17  $\tilde{B} = \tilde{B} + (\sigma_{y,i}^2 - \sigma_{x,i}^2) \cos(2\hat{\varphi}_{c,i}) - 2\sigma_{xy,i}^2 \sin(2\hat{\varphi}_{c,i})$ ;
- 18 **end**
- 19  $\hat{\varphi}_{c,0} = \frac{1}{2} \arctan(\frac{\tilde{A}}{\tilde{B}})$ ;
- 20 **for**  $i = 1$  **to**  $\text{length}(\mathbf{z}_t)$  **do**
- 21 retrieve  $\langle \hat{\rho}_i, \bar{x}_i, \bar{y}_i \rangle$  from  $\mathbf{z}_t^i$ ;
- 22  $\hat{\rho}_i = \hat{\rho}_{c,i} + \hat{\rho}_{c,0}$ ;
- 23  $\hat{\rho}_i = \bar{x}_i \cos \hat{\varphi}_i + \bar{y}_i \sin \hat{\varphi}_i$ ;
- 24 replace  $\langle \hat{\rho}_i, \hat{\varphi}_i \rangle$  of  $\mathbf{z}_t^i$ ;
- 25 **end**
- 26 **return**  $\mathbf{Z}_t$ ;

Algorithm 1 describes the implementation of the proposed method. The input of the algorithm is raw polar data of the laser ranger ( $\mathbf{r}_t, \alpha_t$ ), while the output of the algorithm is the vector of detected lines  $\mathbf{z}_t$ .

A comparison between the single and multiple line fitting methods is shown in Figure 1. The multiple line fitting method gives a better estimation in the sense that lines are correctly estimated and the corners are better defined. Therefore, this will improve global estimation and map building, without additional efforts, since the cost of this multiple line fitting is always of the order  $\mathcal{O}(N_l)$  as in the single-line fitting.

## LS Pose Estimation

In this section, we present a simple but effective method of estimating the pose of a robot based on line features. The method assumes that a data preprocess has been previously done so that detected line features of the environment have been obtained  $\mathbf{z}_t^i = \langle \rho_d^i, \varphi_d^i \rangle$ . It also assumes that the map, containing line features  $\mathbf{m}_t^j = \langle \rho_m^j, \varphi_m^j \rangle$ , is known or it is being estimated.

The main issue of this method is that it estimates the robot pose in the LS sense, and it can be applied for global localization under the assumption of known data association. Since this is not a realistic approach in most real applications, the method requires a previous association between the detected and map features, which is given by the hypothesis  $\mathcal{H}$ , a list that relates each detected  $i$  feature with its corresponding feature on the map  $j = \mathcal{H}(i)$ . If a feature is not associated, then we set  $\mathcal{H}(i) = 0$ .

Based on the well-known line model,

$$\rho_d^i = \rho_m^{\mathcal{H}(i)} - x \cos \varphi_m^{\mathcal{H}(i)} - y \sin \varphi_m^{\mathcal{H}(i)} + \epsilon_{\rho,i} \quad \text{and} \quad (1)$$

$$\varphi_d^i = \varphi_m^{\mathcal{H}(i)} - \theta + \epsilon_{\varphi,i}, \quad (2)$$

where  $\mathbf{p} = [x \ y]^T$  is the Cartesian robot position and  $\theta$  the orientation.  $\epsilon_{\rho,i}$  and  $\epsilon_{\varphi,i}$  are distance and angle errors between the

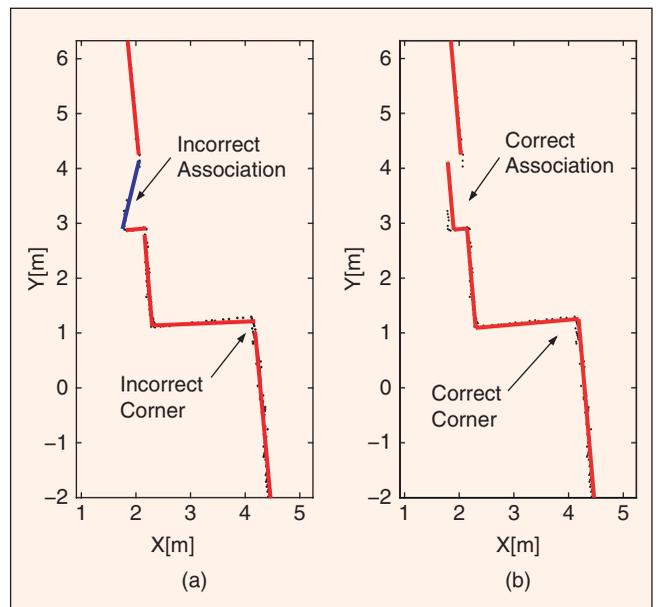


Figure 1. Comparative between (a) single and (b) multiple line fit.

detected lines and predicted ones. Error covariances are assumed to be  $\Sigma_\rho = \sigma_\rho^2 \mathbf{I}$  and  $\Sigma_\varphi = \sigma_\varphi^2 \mathbf{I}$ , respectively.

The pose estimation problem can be easily separated because (1) is affected only by the Cartesian pose, while (2) is affected only by the orientation. In addition to this, we can easily see that both equations are linear with respect to the pose variables, and they can be rewritten in matrix form as

$$\begin{aligned}\rho_e &= -\mathbf{X} \cdot \mathbf{p} + \epsilon_\rho, \\ \varphi_e &= [1 \dots 1]^T \theta + \epsilon_\varphi,\end{aligned}$$

where

$$\rho_e = [\rho_d^1 - \rho_m^{\mathcal{H}(1)}, \dots, \rho_d^{N_l} - \rho_m^{\mathcal{H}(N_l)}]^T,$$

$$\mathbf{X} = \begin{bmatrix} \cos \varphi_m^{\mathcal{H}(1)} & \sin \varphi_m^{\mathcal{H}(1)} \\ \vdots & \vdots \\ \cos \varphi_m^{\mathcal{H}(N_l)} & \sin \varphi_m^{\mathcal{H}(N_l)} \end{bmatrix},$$

$$\varphi_e = [\varphi_d^1 - \varphi_m^{\mathcal{H}(1)}, \dots, \varphi_d^{N_l} - \varphi_m^{\mathcal{H}(N_l)}]^T, \text{ and}$$

$$\epsilon_\rho = [\epsilon_{\rho,1}, \dots, \epsilon_{\rho,N_l}]^T, \epsilon_\varphi = [\epsilon_{\varphi,1}, \dots, \epsilon_{\varphi,N_l}]^T.$$

### Algorithm 2: Line-Based Pose Estimation

```

1  PoseEst( $\mathbf{z}_t, \mathbf{m}_t, \mathcal{H}_t, \mathbf{x}_{t-1}$ )
2   $A = B = C = D = E = \varphi_e = N_l = 0$ ;
3  for  $i = 0$  to length( $\mathcal{H}_t$ ) do
4    if  $\mathcal{H}_t(i) > 0$  then
5      retrieve  $\langle \rho_m^j, \varphi_m^j \rangle$  from  $\mathbf{m}_t^{\mathcal{H}_t(i)}$ ;
6      retrieve  $\langle \rho_d^i, \varphi_d^i \rangle$  from  $\mathbf{z}_t^i$ ;
7       $A = A + \cos^2 \varphi_m^j$ ;
8       $B = B + \sin^2 \varphi_m^j$ ;
9       $C = C + \sin \varphi_m^j \cos \varphi_m^j$ ;
10      $D = D + (\rho_d^i - \rho_m^j) \cos \varphi_m^j$ ;
11      $E = E + (\rho_d^i - \rho_m^j) \sin \varphi_m^j$ ;
12      $\varphi_e = \varphi_e + \varphi_d^i - \varphi_m^j$ ;
13      $N_l = N_l + 1$ ;
14   end
15 end
16 retrieve  $\langle x_t, y_t, \theta_t \rangle$  from  $\mathbf{x}_{t-1}$ ;
17 if  $j > 0$  and  $A \cdot B - C^2 \neq 0$  then
18    $x_t = \frac{C \cdot E - B \cdot D}{A \cdot B - C^2}$ ;
19    $y_t = \frac{C \cdot D - A \cdot E}{A \cdot B - C^2}$ ;
20    $\Sigma_{xy} = \sigma_\rho^2 \begin{bmatrix} A & C \\ C & B \end{bmatrix}^{-1}$ ;
21 else  $x_t = x_{t-1}$ ;  $y_t = y_{t-1}$ ;  $\Sigma_{xy} = \mathbf{0}$ ;
22 if  $j > 0$  then  $\theta_t = -\frac{\varphi_e}{N_l}$ ;  $\Sigma_\theta = \frac{\sigma_\varphi^2}{N_l}$ ;
23 else  $\theta_t = \theta_{t-1}$ ;  $\Sigma_\theta = 0$ ;
24  $\mathbf{x}_t = [x_t \ y_t \ \theta_t]^T$ ;
25  $\Sigma_{\mathbf{x}_t} = \begin{bmatrix} \Sigma_{xy} & \mathbf{0} \\ \mathbf{0}^T & \Sigma_\theta \end{bmatrix}$ ;
26 return  $\mathbf{x}_t$  and  $\Sigma_{\mathbf{x}_t}$ ;
```

Therefore, the LS fitting provides the following estimations for the Cartesian pose and orientation:

$$\hat{\mathbf{p}} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = -(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \rho_e, \quad \hat{\theta} = -\bar{\varphi}_e,$$

and the covariance estimation error is as follows:

$$\Sigma_{xy} = \sigma_\rho^2 (\mathbf{X}^T \mathbf{X})^{-1}, \quad \Sigma_\theta = \frac{\sigma_\varphi^2}{N_l}.$$

Matrix  $\mathbf{X}^T \mathbf{X}$  is singular when all lines considered have the same orientation. However, because the method estimates the Cartesian positions separately from the orientation, and in such situations, a valid estimation of the orientation can still be provided.

Algorithm 2 describes the implementation of the method, where the following considerations must be taken into account:

$$\mathbf{X}^T \mathbf{X} = \begin{bmatrix} A & C \\ C & B \end{bmatrix}, \quad \mathbf{X}^T \rho_e = \begin{bmatrix} D \\ E \end{bmatrix},$$

with

$$A = \sum_{i=1}^{N_l} \cos^2 \varphi_m^{\mathcal{H}(i)}, \quad B = \sum_{i=1}^{N_l} \sin^2 \varphi_m^{\mathcal{H}(i)},$$

$$C = \sum_{i=1}^{N_l} \cos \varphi_m^{\mathcal{H}(i)} \sin \varphi_m^{\mathcal{H}(i)},$$

$$D = \sum_{i=1}^{N_l} (\rho_d^i - \rho_m^{\mathcal{H}(i)}) \cos \varphi_m^{\mathcal{H}(i)}, \text{ and}$$

$$E = \sum_{i=1}^{N_l} (\rho_d^i - \rho_m^{\mathcal{H}(i)}) \sin \varphi_m^{\mathcal{H}(i)}.$$

A similar approach is followed by Araujo and Aldon in [26], where the main difference is given in the definition of the performance index. In [26], the proposed index is based on the minimization of Cartesian projections between detected lines and map lines with respect to robot pose. This representation introduces the advantage of including points and lines under the same performance index, but the analytic solution to this problem leads to solve a fourth order polynomial equation derived from a trigonometric rotation equation. In addition, it presents four possible solutions (real and/or complex), and therefore, a study on the Jacobian is required to discard local maxima.

To analyze the performance of the proposed method, a simulation, using a mobile robot moving through a regular environment with walls and corridors, has been considered. Obviously, the line detection method proposed in the previous section has been taken into account for laser-ranger measurements, where a noise has been added to them to simulate more realistic data. Figure 2 shows the pose estimation error between ground truth and estimated pose with LS and Araujo's methods. Crosses

depicted on Figure 2(a) and (b) represent iterations where  $\mathbf{X}^T\mathbf{X}$  becomes singular and, therefore, they should not be considered. It can be appreciated that Araujo's method presents more than three times higher cumulative computational time ( $T_c$ ) than that of LS pose estimation. In fact, the mean computational times of LS and Araujo's methods are  $4.92 \cdot 10^{-4}$  and  $18 \cdot 10^{-4}$  ms, respectively. This aspect is particularly relevant, since each particle of the FastSLAM algorithm will correct its pose using this method (see the next section for more details).

Moreover, consider the mean estimation error (MSE) of Cartesian positions and orientations:

$$\text{MSE}_{xy} = \frac{1}{h} \sum_{k=0}^h (x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2, \quad \text{and}$$

$$\text{MSE}_\theta = \frac{1}{h} \sum_{k=0}^h (\theta_k - \hat{\theta}_k)^2,$$

where  $h$  is the number of iterations of the simulation. The performance in terms of  $\text{MSE}_x$  and  $\text{MSE}_\theta$  of LS and Araujo's methods is shown in Table 1, where  $N_l$  is the minimum number of lines required to produce a valid estimation. Araujo's method gives more accurate results than those of LS in the cases where more than two lines have been detected ( $N_l \geq 3$  or  $N_l \geq 4$ ). However, it is clear that the accuracy of both the methods is good enough for many practical situations.

## Asynchronous Multirate Techniques Applied to SLAM Problem

### Probabilistic Robot Localization and Map Building

The key idea of PFs is to represent the posterior distribution of a signal  $\mathbf{x}_t$  by a set of random samples (particles) drawn from this posterior. Such a set of samples is an approximation of the real distribution and, in general, PFs can represent much broader spaces of distributions, rather than Gaussian distributions such as EKF or UKF.

We denote the set of particles describing the posterior distribution as

$$\mathcal{X}_t = \{\mathbf{x}_t^{[1]}, \mathbf{x}_t^{[2]}, \dots, \mathbf{x}_t^{[M]}\},$$

where  $M$  is the number of particles and each particle  $\mathbf{x}_t^{[k]}$  is drawn from the posterior distribution

$$\mathbf{x}_t^{[k]} \sim p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}), \quad (3)$$

where  $\mathbf{z}_{1:t}$  represents the set of whole measurements and  $\mathbf{u}_{1:t}$  the set of whole inputs. The approximation is valid if the number of samples is large enough, generally  $M \geq 100$ . Similar to most popular filters, the PF uses a recursive estimation for approximating the posterior distribution. Therefore, (3) simplifies to

$$\mathbf{x}_t^{[k]} \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_t, \mathbf{u}_t),$$

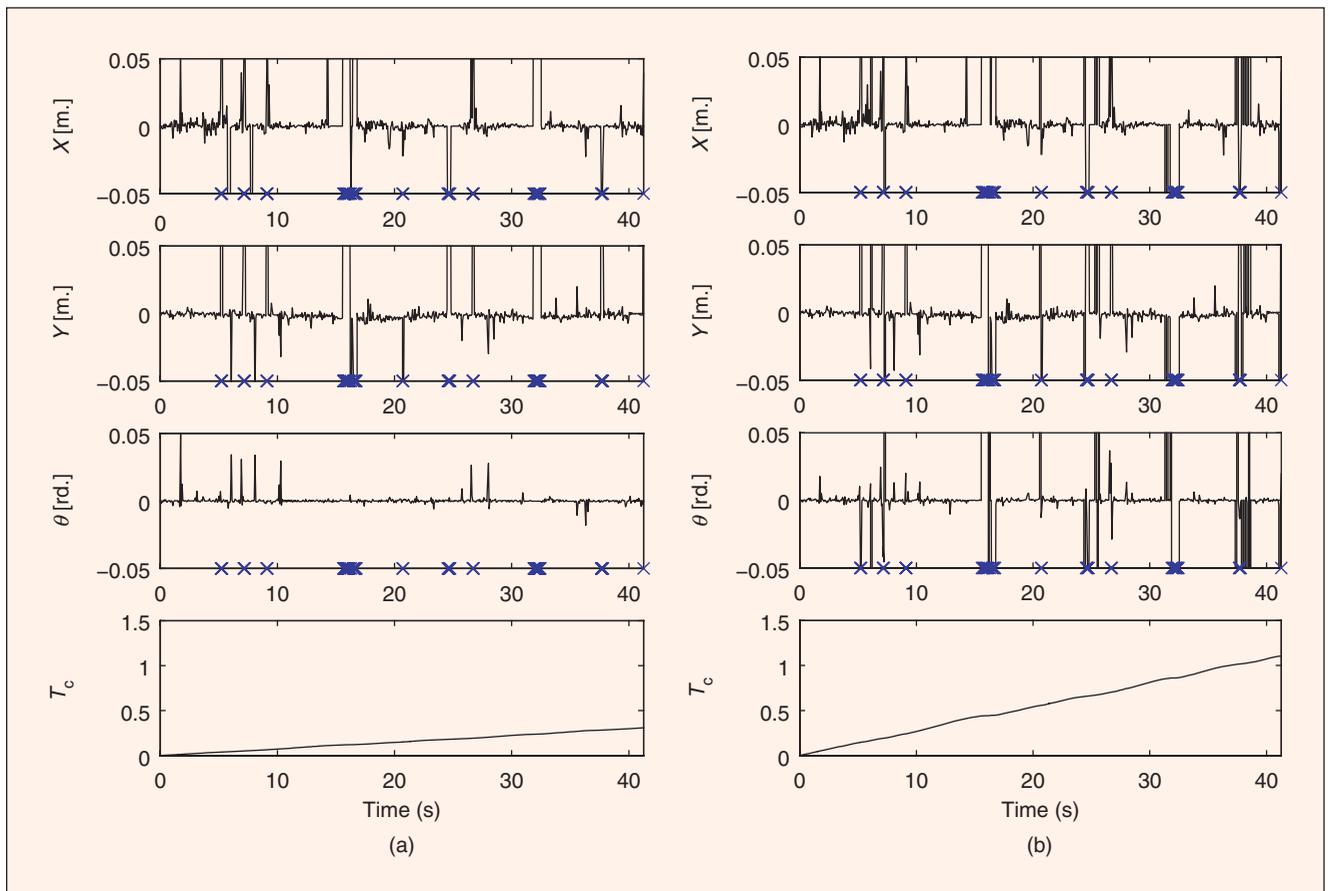


Figure 2. Pose estimation error and computational time with (a) LS pose estimation and (b) Araujo's methods.

where only inputs and measurements at the present time instant are used. The main problem is that the posterior distribution is not known, and therefore a proposal distribution has to be used based on the importance factor. For each particle, a weight obtained from  $w_t^{[k]} = p(\mathbf{z}_t | \mathbf{x}_t^{[k]})$  is computed, where  $p(\mathbf{z}_t | \mathbf{x}_t)$  is the PDF of the sensor model. The set of weighted particles represents (an approximation) of the posterior. This method is known as sequential importance resampling, where a resampling step and normalization must be done to avoid the algorithm degeneracy [6], [27].

FastSLAM is the PF approach to SLAM. Unfortunately PF in SLAM is also affected by the *curse of dimensionality*, with exponential growth ( $M \times N$ ). For that reason, Rao-Blackwellized PF [28] is applied, where the pose of the robot is estimated with a standard PF and the map is estimated by an EKF. In FastSLAM, each particle contains an estimated robot pose  $\mathbf{x}_t^{[k]}$  and a set of Kalman filters with  $\mu_t^{[k]}$  and covariance  $\Sigma_t^{[k]}$ , one for each feature in the map. Thus, the problem is decomposed in the  $N + 1$  problem: 1 for estimating the robot pose and  $N$  for estimating the features in the map. A key advantage of this solution is that maps between particles are not correlated, and therefore errors in the map are filtered through the resampling process, where only the best particles (with best maps) will survive.

The FastSLAM described so far is known as FastSLAM 1.0 [15], while in FastSLAM 2.0 [20] the proposal distribution takes the measurements  $\mathbf{z}_t$  into account when sampling the pose  $\mathbf{x}_t^{[k]}$  based on an EKF approach. This modification follows a similar approach to that of [29] and [30] for PF.

In mobile robots, inputs and outputs have different sampling rates. For instance, odometry sensors, such as encoders, gyros, and accelerometers, are typically sampled faster than

external sensors such as lasers [31], sonars, or vision systems [32], [33]. This is a problem that arises from the inherent technological limitations of sensors, communication channels, and processing cost, etc., which can not be neglected. A typical solution to overcome this problem is to reduce the overall sampling period to the slowest one. However, it is well known that this approach may decrease the overall system performance. In addition to this, discretization errors will also increase with higher sampling periods. Therefore, faster dynamics require fast sampling rates.

In this article, we describe an asynchronous filtering method to deal with measurements of sensors at different sampling rates. One of the key ideas of the method is to use an asynchronous hold to extrapolate the inputs of the system. Another key point is the asynchronous execution of prediction and update steps in the filtering method, which maintains a good system performance. The prediction step is executed within at least a prespecified sampling period (generally at a fast sampling rate to reduce discretization errors), and the update step is executed only when measurements are asynchronously received.

### Asynchronous Holds

A multirate hold is a hybrid device for generating, from a sequence of inputs sampled at a slow sampling rate, a continuous signal that may be discretized at a high sampling rate. The mathematical background of multirate high-order holds (MR-HOH) and samplers is described in [34]–[36]. Initially, multirate holds were introduced as a generalizations of classical holds such as zero-, first-, and second-order holds (ZOH, FOH, and SOH). Later on, in [37], a wide variety of holds were obtained from general primitive functions. The idea behind general primitive functions is to generate an extrapolated continuous signal based on input samples  $\{\mathbf{u}_{t_j}, \mathbf{u}_{t_{j-1}}, \dots, \mathbf{u}_{t_{j-n}}\}$  uniformly distributed at a low frequency (dual-rate sampling) [37].

In this article, we extend the formulation for the asynchronous case, where input samplings are not uniformly distributed:

$$\hat{\mathbf{u}}_t = \sum_{l=0}^n \mathbf{f}_l(t, \mathbf{u}_{t_{j-l}}, t_{j-l}), \quad (4)$$

where  $\mathbf{u}_{t_j}$  denotes an input that has been sampled at time instant  $t_j$  being  $t_{j-n} \leq \dots \leq t_j \leq t$ . The primitive function  $\mathbf{f}_l(t)$  generates the continuous signal  $\hat{\mathbf{u}}_t$ , which can be computed at any desired time instant. Thus, the asynchronous hold is in charge of generating a “continuized” signal regardless of when the inputs were sampled. Asynchronous holds may be used in two different situations: for estimating signals in between samples as well as for overcoming the data-missing problem.

Algorithm 3 implements the asynchronous hold of order  $n$ , based on a general primitive function. To implement the asynchronous hold, a shift register  $\mathcal{U}_j = \{\mathbf{u}_{t_j}, \dots, \mathbf{u}_{t_{j-n}}\}$  is required to log the signal. Table 2 summarizes some primitive functions that can be used in asynchronous holds. In spline holds, a set of coefficients  $c_{j,l}$  are obtained when a spline curve is adjusted to the previous inputs. In Taylor holds, input derivatives are obtained using the backward approximation.

Table 1. MSE performance.

MSE	Method	$N_1 \geq 2$	$N_1 \geq 3$	$N_1 \geq 4$
$MSE_x [m^2]$	LS	$1.8 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$
	Araujo	27.85	$7.42 \cdot 10^{-5}$	$6.26 \cdot 10^{-5}$
$MSE_\theta [rd^2]$	LS	$1.75 \cdot 10^{-5}$	$1.93 \cdot 10^{-5}$	$1.53 \cdot 10^{-5}$
	Araujo	0.32	$7.7 \cdot 10^{-6}$	$4.74 \cdot 10^{-6}$

#### Algorithm 3: Asynchronous Hold

```

1  Asynchronous-Hold( $\mathbf{u}_t, \mathcal{U}_j, t, j$ )
2  If  $\mathbf{u}_t$  is updated then
3    shift out  $\mathbf{u}_{t_{j-n}}$  and shift in  $\mathbf{u}_t$  from  $\mathcal{U}_j$ ;
4     $\hat{\mathbf{u}}_t = \mathbf{u}_t$ ;
5     $j = t$ ;
6  else
7     $\hat{\mathbf{u}}_t = \mathbf{0}$ ;
8    for  $l = 0$  to  $n$  do
9      retrieve  $\mathbf{u}_{t_{j-l}}$  from  $\mathcal{U}_j$ ;
10      $\hat{\mathbf{u}}_t = \hat{\mathbf{u}}_t + \mathbf{f}_l(t, \mathbf{u}_{t_{j-l}}, t_{j-l})$ ;
11   end
12 end
13 return  $\hat{\mathbf{u}}_t, \mathcal{U}_j$  and  $j$ ;

```

An experiment has been performed on a mobile robot moving in a real environment, where signal profiles of linear and angular velocities are shown in Figure 3. This figure also shows the MSE of the extrapolated signals when using a, MR-FOH (first order Lagrange hold), MR-SOBH (second-order Bezier hold), MR-SOTH (second-order Taylor hold) and MR-SOSH (second-order spline hold) (first order cases of spline, Bezier, and Taylor functions are the same as that of the FOH.). The sampling period of signals is  $T_s = 100$  ms,  $i$  is the number of iterations that signals are extrapolated, and  $n$  is the order of the hold. Obviously, stability cannot be guaranteed for any arbitrary extrapolation time interval, since MSE increases exponentially with  $i$ . However, it can be seen that for all cases with  $i \leq 5$  the results of MR-FOH, MR-SOBH, MR-SOTH, and MR-SOSH significantly improve the results of the naive MR-ZOH solution.

### Multirate Asynchronous FastSLAM

The proposed multirate asynchronous FastSLAM (MR-FastSLAM) has been

implemented in the two different versions FastSLAM 1.0 and FastSLAM 2.0, described in Algorithm 4, where  $M$  is the number of particles. In FastSLAM 1.0, pseudo-code lines 11 and 12 are not used. The pose estimation method used in line 2 is the one described in the previous section. This modifies the original contribution of FastSLAM [20] in the sense that

Table 2. Primitive functions.	
Type	Function
Interpolation (Lagrange)	$\hat{\mathbf{u}}_t = \sum_{l=0}^n \left[ \prod_{\substack{q=0 \\ q \neq l}}^n \frac{t - t_{j-q}}{t_{j-l} - t_{j-q}} \right] \mathbf{u}_{t_{j-l}}$
Interpolation (Spline)	$\hat{\mathbf{u}}_t = \sum_{l=0}^n c_{j,l} (t - t_{j-l})^l \text{ with } \{c_{j,-n+1,0}, \dots, c_{j,0}, \dots, c_{j,-n+1,m}, \dots, c_{j,m}\}$ = spline( $\mathcal{U}_t$ )
Approximation (Bezier)	$\hat{\mathbf{u}}_t = \sum_{l=0}^n \left[ \frac{n!}{l!(n-l)!} \left(1 + \frac{t - t_j}{t_j - t_{j-n}}\right)^{n-l} \left(\frac{t - t_j}{t_j - t_{j-n}}\right)^l \right] \mathbf{u}_{t_{j-l}}$
Approximation (Taylor)	$\hat{\mathbf{u}}_t = \sum_{l=0}^n \frac{(t - t_j)^l}{l!} \mathbf{u}_{t_j}^{(l)} \text{ with } \mathbf{u}_{t_j}^{(l)} = \frac{\mathbf{u}_{t_j}^{(l-1)} - \mathbf{u}_{t_{j-1}}^{(l-1)}}{t_j - t_{j-1}}$

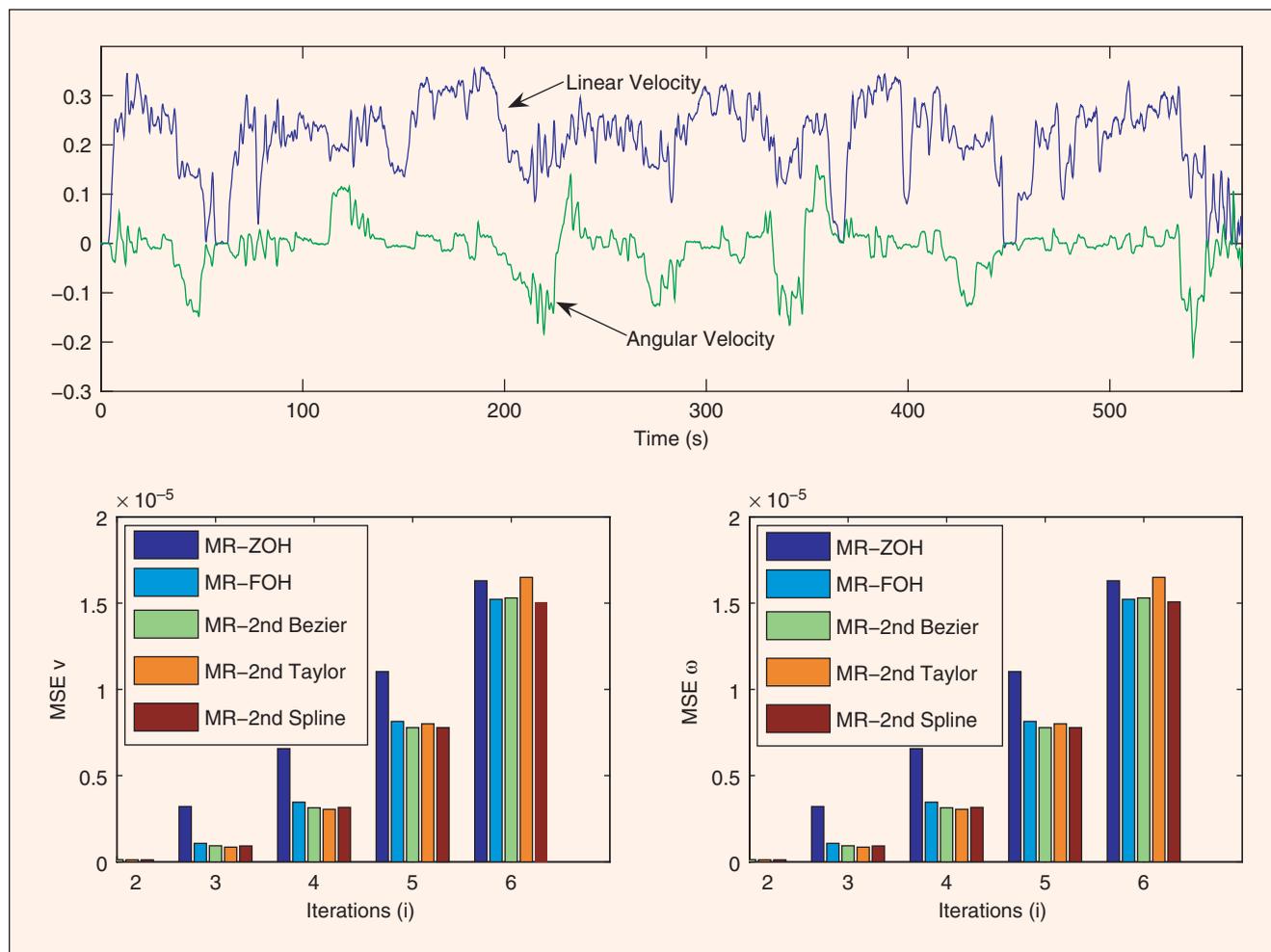


Figure 3. Input profile and MSE for different holds.

## The Kalman filter gives a robust, optimal, and recursive state estimation to fuse redundant sensor information.

an LS approach is used instead of an EKF approach. The main advantage of this new approach is the reduced computational cost of LS.

The steps are similar to the conventional FastSLAM algorithms, where differences lie in the inclusion of asynchronous holds and asynchronous prediction and update steps. In that sense, asynchronism is due to a time-varying execution time period  $t$ .

- ◆ If the time  $T_s$  has elapsed since the last execution ( $t - 1$ ) period of the algorithm and no measurement has been received yet, then only the prediction step is executed with  $t = (t - 1) + T_s$  to maintain a good system performance with low discretization error.
- ◆ Whenever a measurement is received, the prediction and update steps are executed with  $t = (t - 1) + t_z$ ,

### Algorithm 4: Multirate FastSLAM 2.0

```

1  MR-FastSLAM 2.0( $\mathcal{X}_{t-1}, \mathbf{u}_t, \mathbf{z}_t, \mathcal{U}_j, t, j$ )
2   $\bar{\mathcal{X}}_t = \mathcal{X}_t = \mathbf{z}_t^s = \emptyset$ ;
3   $[\hat{\mathbf{u}}_t, \mathcal{U}_j, j] = \text{Asynchronous-Hold}(\mathbf{u}_t, \mathcal{U}_j, t, j)$ ;
4   $[\hat{\mathbf{u}}_t, \mathcal{U}_t] = \text{MR-Hold}(\mathbf{u}_t, \mathcal{U}_t, t)$ ;
5  for  $k = 1$  to  $M$  do
6    retrieve  $\langle \mathbf{x}_{t-1}^{[k]}, \mathbf{m}_{t-1}^{[k]}, w_{t-1}^{[k]} \rangle$  from  $\mathcal{X}_{t-1}$ ;
7     $\mathbf{x}_t^{[k]} = \text{MotionModel}(\hat{\mathbf{u}}_t, \mathbf{x}_{t-1}^{[k]}, t)$ ;
8    If  $\mathbf{z}_t$  has been sampled then
9       $\hat{\mathbf{z}}_t = \text{MeasurementModel}(\mathbf{m}_{t-1}^{[k]}, \mathbf{x}_t^{[k]})$ ;
10      $[\mathcal{H}_t, w_t^{[k]}] = \text{DataAssociation}(\mathbf{z}_t, \hat{\mathbf{z}}_t)$ ;
11      $[\hat{\mathbf{x}}_t^{[k]}, \Sigma_{\mathbf{x}_t}^{[k]}] = \text{PoseEst}(\mathbf{m}_{t-1}^{[k]}, \mathcal{H}_t, \mathbf{z}_t, \mathbf{x}_t^{[k]})$ ;
12     sample  $\mathbf{x}_t^{[k]} \sim \mathcal{N}(\hat{\mathbf{x}}_t^{[k]}, \Sigma_{\mathbf{x}_t}^{[k]})$ ;
13     for  $i = 1$  to  $\text{length}(\mathbf{z}_t)$  do
14       If  $\mathcal{H}_t(i) = 0$  then
15         // New feature
16          $\tilde{\mathbf{m}}_t^{[k]} = \text{InvMeasModel}(\mathbf{z}_t^i, \mathbf{x}_t^{[k]})$ ;
17         add  $\tilde{\mathbf{m}}_t^{[k]}$  to  $\mathbf{m}_t^{[k]}$ ;
18       else
19         // Update feature
20          $\mathbf{m}_t^{[k]} = \text{EKF}(\mathbf{m}_{t-1}^{[k]}, \mathbf{z}_t^i, \mathbf{x}_t^{[k]})$ ;
21       end
22     end
23     remove unstable features of  $\mathbf{m}_t^{[k]}$ ;
24   else  $w_t^{[k]} = w_{t-1}^{[k]}$ ;  $\mathbf{m}_t^{[k]} = \mathbf{m}_{t-1}^{[k]}$ ;
25   add  $\langle \mathbf{x}_t^{[k]}, \mathbf{m}_t^{[k]}, w_t^{[k]} \rangle$  to  $\bar{\mathcal{X}}_t$ ;
26 end
27 if  $\mathbf{z}_t$  has been sampled then
28   normalize  $w_t^{[k]}$  of  $\bar{\mathcal{X}}_t$ ;
29    $\mathcal{X}_t = \text{Resampler}(\bar{\mathcal{X}}_t)$ ;
30 else  $\mathcal{X}_t = \bar{\mathcal{X}}_t$ ;
31 return  $\mathcal{X}_t, \mathcal{U}_t$  and  $j$ ;

```

with  $t_z$  being the elapsed time between the last execution and the measurement sampling. Note that the prediction step must necessarily be performed to coherently fuse measurements and states at the same time instant.

Figure 4 shows an example of a tasks chronogram for the proposed algorithm. In this case, four different tasks have been considered: encoder, laser, prediction, and update, each of them running at different frequencies. The encoder task is executed approximately at  $T_{\text{inc}} \approx 100$  ms, each time an encoder measurement is received, while Laser Task is executed at  $T_{\text{las}} \approx 400$  ms. The laser task has been subdivided into more subtasks to indicate when each of the detected lines have been processed. In Figure 4, several lines are typically processed for a specific laser scan. Prediction and update tasks are executed according to the conditions mentioned previously. In this sense, the update step is not performed until the processing of the laser scan has been completed. In Figure 4, it can also be seen that the prediction task is regularly executed with a sampling period of  $T_s = 100$  ms. However, whenever a laser measurement is received, the prediction task is executed again, immediately followed by the update task. In addition to this, it is interesting to observe the effect of the asynchronous hold. Despite the asynchronous encoder samplings, the prediction step, which uses these measurements, is executed at its own sampling frequency (also asynchronous).

## Experimental Results

Experimental tests have been performed in an indoor environment, on the Explorer powered wheelchair developed by the company TGR (Italy). This vehicle has two driving wheels. The odometric system consists of two incremental encoders connected to independent passive wheels aligned with the axes of the driving wheels. An FOG HITACHI module, HOFG-1, has also been used to accurately measure the angular velocity. The laser scanner measurements have been acquired by the SICK LMS200 installed on the vehicle. A resolution of  $0.5^\circ$  and a spectrum of  $180^\circ$  have been chosen to have a number of measurements that could simultaneously guarantee good map

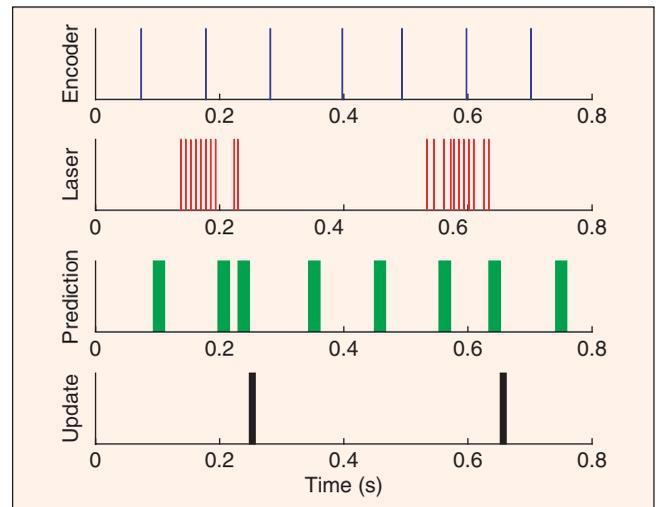


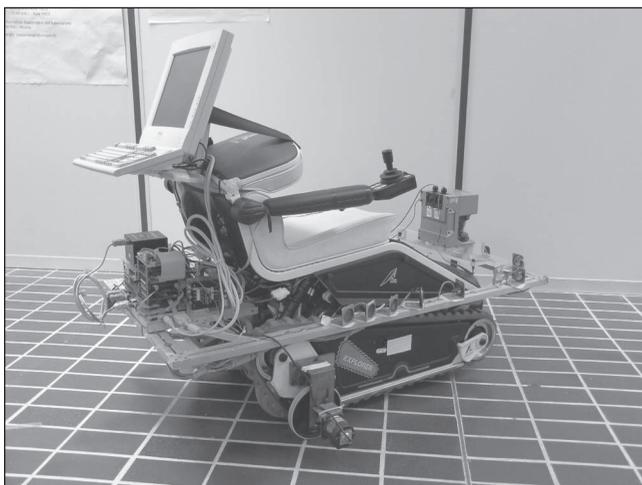
Figure 4. Task chronogram example.

building and real-time implementation. The TGR Explorer powered wheelchair with the data acquisition system is shown in Figure 5.

Different tests have been performed at the Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione (DIIGA). Department for analyzing the performance of the proposed MR-FastSLAM algorithm. In this section, two significant tests are introduced and discussed: a loop-shape and an L-shape experiment. Figure 6 shows the DIIGA Department taken from a computer aided design (CAD) plan as well as pose estimations of both experiments acquired with classical MCL and assuming that the map is known. The purpose of the first experiment is to investigate whether the multirate sampling aims to solve the loop-closing problem. The starting and ending pose of this experiment is  $x = 38.4$  m,  $y = 21.8$  m, and  $\theta = \pi$  rd. In the second experiment of an L-shaped corridor, doors 1 and 2 were closed. The main difficulty lies on the U-turn done in the middle of the experiment, where the robot has the greatest difficulty in estimating the orientation due to the lack of visible walls and the rate of turn. The starting and ending position of this experiment is  $x = 23.5$  m,  $y = 45$  m, and  $\theta = 0$  rd.

Figure 7(a) and (d) shows the results obtained when using the single-rate FastSLAM 1.0 for both experiments (running at the laser sampling frequency), in which the orientation estimation clearly fails when turning. In this case, the number of samples used for the FastSLAM estimation is 100, which is probably not enough, and therefore the results may also be improved by increasing the number of particles, which is associated with a higher computational cost. On the other hand, Figure 7(b) and (e) shows the results for the multirate FastSLAM 1.0, where it can be seen that it closes the loop and also improves the results of the the corridor experiment but still has some difficulties in the estimation of the orientation.

In addition to this, the map building and also pose estimation robustness can be significantly improved when using the MR-FastSLAM 2.0 instead of MR-FastSLAM 1.0. Figure 7(c) and (f) shows the estimation results for the MR-FastSLAM 2.0 case, where the map estimation is much more accurate (corners with forms of  $90^\circ$ ). In this sense, as seen in



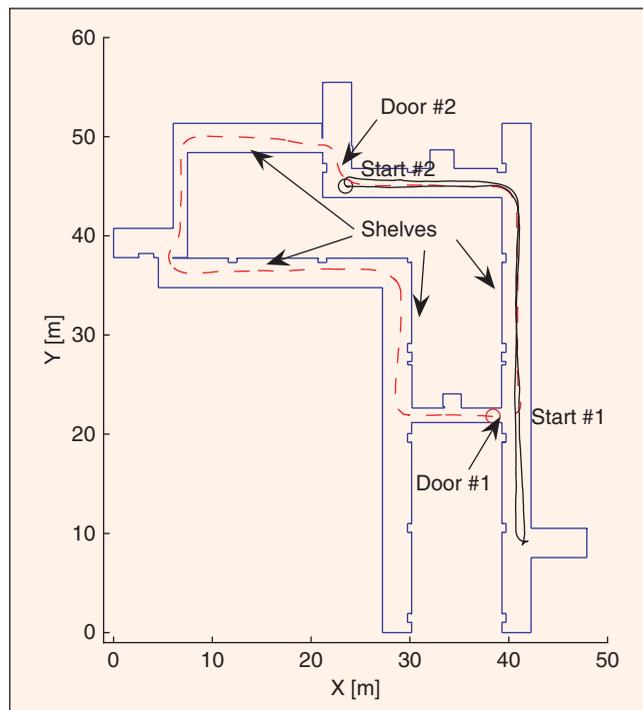
**Figure 5.** TGR explorer with data acquisition system for FOG sensor, incremental encoders, and laser scanner.

**The goal of this method is the minimization of the sum of squared distances for all lines.**

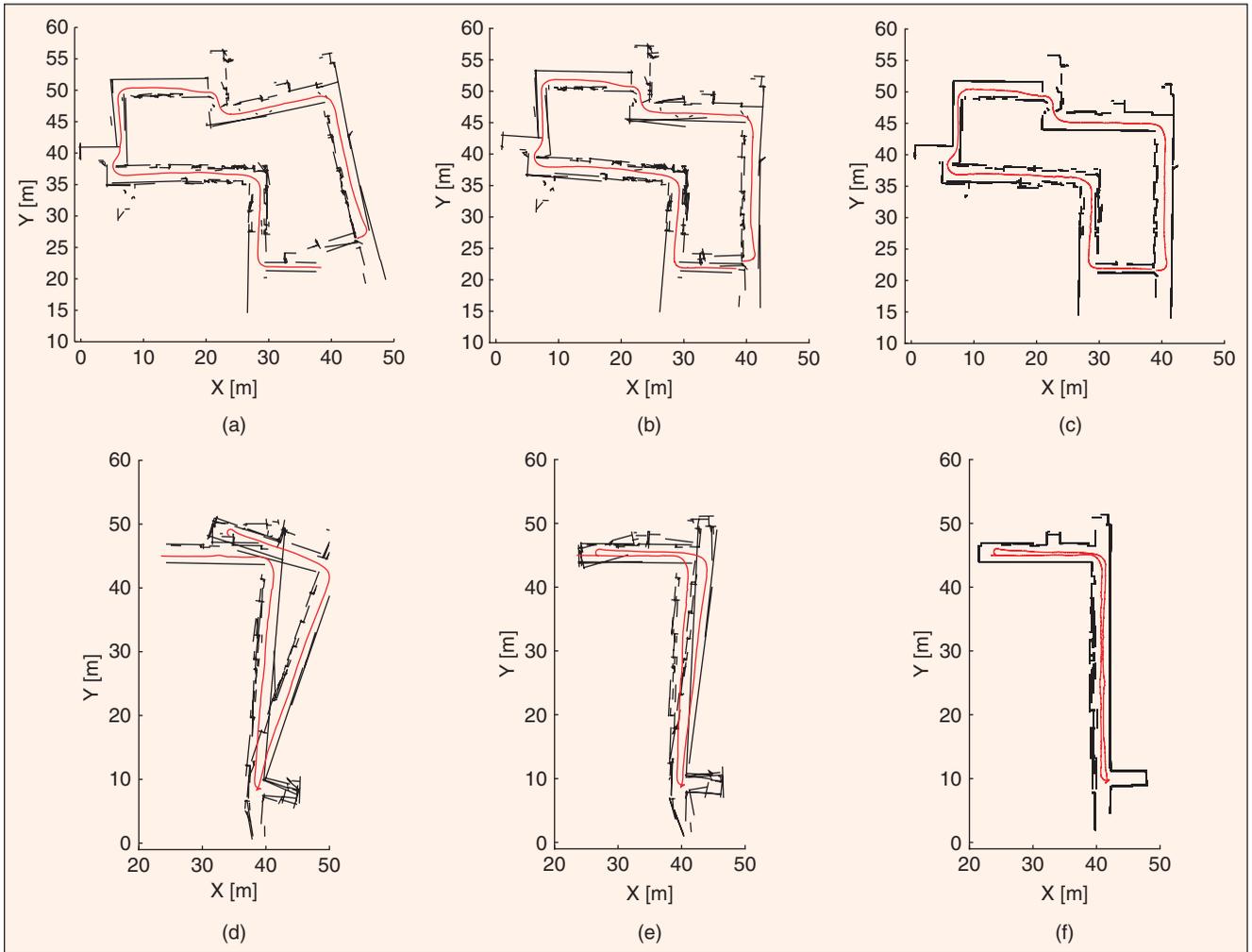
Figure 8, MR-FastSLAM 2.0 uses less number of features, which is also an indicator of its good accuracy, decreasing the computational cost, but without a detriment on pose and map estimation. The same performance has been also obtained in the other developed experiments that are characterized by simple paths along the corridors and laboratories of the DIIGA Department. Moreover, no specific assumptions have been considered on the structure and on the parameters of the considered mobile base. Therefore, the same performance of the proposed MR-FastSLAM algorithm can be obtained on different mobile bases with different kinematic models and parameters.

## Conclusions

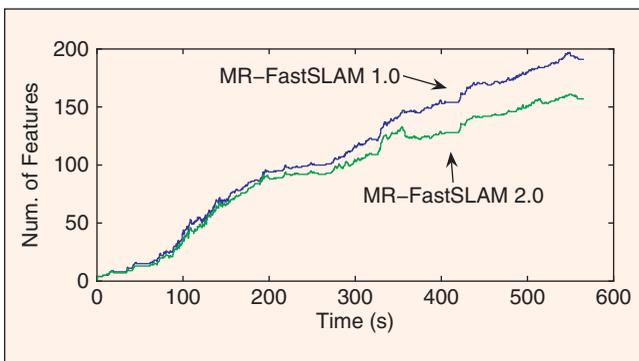
In this article, a new approach for multirate fusion, probabilistic self-localization, and map building for mobile robots has been developed. To overcome the filtering problem of dynamic systems with inputs and outputs sampled at a different rates, asynchronous multirate holds have been used. In addition to this, an asynchronous fusion method (with asynchronous prediction-update steps) is also proposed. The prediction step is executed at least within a maximum period, which ensures a low discretization error and stability, while an update step fuses measurements as they are received. Although in this



**Figure 6.** CAD plane with loop-shape (dashed line) and L-shape (continuous line) experiments pose estimations with MCL.



**Figure 7.** Experiment results: (a) loop experiment with FastSLAM 1.0 at single-rate; (b) loop experiment with FastSLAM 1.0 at multirate; (c) loop experiment with FastSLAM 2.0 at multirate; (d) corridor experiment FastSLAM 1.0 at single-rate; (e) corridor experiment with FastSLAM 1.0 at multirate; (f) corridor experiment with FastSLAM 2.0 at multirate.



**Figure 8.** The number of estimated features for MR-FastSLAM 1.0 and 2.0.

article the multirate asynchronous structure has been applied to PFs in FastSLAM, it is clear that it can be used with many other filters such as EKF, UKF, or discrete-Bayes filters, since they also have the prediction-update steps.

Multirate asynchronous holds are hybrid systems that generate continuous signals from discrete sequences of inputs that

may be uniformly distributed in time or not. The key idea is to use interpolation and approximation functions, or in general any primitive function, to extrapolate continuous signals. Thus, multirate holds act as interfaces for signals with different sampling rates, providing signals properly adapted to the required sampling period (discretization of the continuous signal). In this article, it has been shown that general multirate holds reduce extrapolation errors in comparison to the naive solution of keeping the last updated value (ZOH). This is specially interesting in data-missing problems such as communication failures.

These ideas have been successfully applied to the multirate fusion of laser ranger and odometry measurements, by considering each sensor at its own sampling frequency rate. In this sense, the laser ranger is sampled at a slow sampling rate, while the odometry is sampled several times faster. These ideas can be also extended to other sensor fusion applications, such as vision and inertial fusion. In fact, in [31] and [33], preliminary ideas of multirate (synchronous) EKF and UKF were successfully applied to the mobile robot localization problem (laser and encoder fusion) and three-dimensional tracking problem (vision and inertial fusion).

## We developed an asynchronous filtering method to deal with measurements of sensors at different sampling rates.

This article shows that a significant improvement can be obtained on the localization and map-building problem when the proposed multirate filtering approach is applied. The results show that multirate fusion improves the estimation with respect to the single-rate one for the same number of particles, since the multirate filter has lower discretization errors. Alternatively, similar errors are obtained with less number of particles. Therefore, reducing the number of particles reduces the computational cost.

This article also provides an LS pose-estimation method based on the detected lines in indoor environments. It has been shown that the method is robust and accurate enough with a lower computational cost than that of the Araujo's method [24]. The LS method can successfully be applied to the FastSLAM 2.0 as demonstrated through experimental results.

Finally, this article describes a novel object detection method that is mainly based on multiple line fitting of landmarks (walls) with regular constrained angles. This method is particularly indicated for indoor structured environments and represents an improvement with respect to the standard LS line fitting method without significantly incrementing the computational cost.

To conclude, methods for object detection, LS pose-estimation, and asynchronous multirate filtering are combined to produce a robust and efficient overall method for mobile robot localization. These methods have been validated with experimental real data in a moving mobile robot in an unknown environment for solving the SLAM problem.

### Acknowledgments

This work has been supported by the Spanish Government (MCyT) research project BIA2005-09377-C03-02 and by the Italian Government (MIUR) research project PRIN 2005097207.

### Keywords

multirate fusion, probabilistic, localization, mapping, FastSLAM.

### References

- [1] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, pp. 229–241, 2001.
- [2] S. Julier and J. Uhlmann, "Reduced sigma points filters for the propagation of means and covariances through nonlinear transformations," in *Proc. American Control Conf.*, 2002, vol. 2, pp. 887–892.
- [3] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEE Proc. F*, vol. 140, no. 2, pp. 107–113, 1993.
- [4] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump markov linear systems," *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 613–624, 2001.
- [5] A. Smith and E. Gelfand, "Bayesian statistics without tears: A sampling-resampling perspective," *Am. Stat.*, no. 2, pp. 84–88, 1992.
- [6] J. Carpenter, P. Clifford, and P. Fernhead, "An improved particle filter for non-linear problems," Dept. Math., Imperial College, Tech. Rep., 1997.
- [7] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic bayesian networks," presented at the *Uncertainty in Artificial Intelligence 2000*.
- [8] P. Jensfelt and H. Christensen, "Pose tracking using laser scanning and minimalistic environmental models," *IEEE Trans. Robot. Automat.*, vol. 17, no. 2, pp. 138–147, 2001.
- [9] G. Zunino and H. Christensen, "Simultaneous localization and mapping in domestic environments," presented at the *Int. Conf. Multisensor Fusion and Integration For Intelligent Systems*, 2001.
- [10] J. Castellanos, J. Neira, and J. Tardos, "Multisensor fusion for simultaneous localization and map building," *IEEE Trans. Robot. Automat.*, vol. 17, no. 6, pp. 908–914, 2001.
- [11] S. Thrun, D. Fox, and W. Burgard, "A probabilistic approach to concurrent mapping and localization," *Mach. Learn. Auton. Robots*, vol. 31, pp. 29–53, 1998.
- [12] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Proc. Int. Conf. Robotics and Automation*, 1999, pp. 1322–1328.
- [13] A. Bonci, G. Ippoliti, L. Jetto, T. Leo, and S. Longhi, "Methods and algorithms for sensor data fusion aimed at improving the autonomy of a mobile robot," in *Advances in Control of Articulated and Mobile Robots*. Berlin, Germany: Springer, 2004, vol. 10 pp. 191–222.
- [14] A. Bonci, G. Ippoliti, A. L. Manna, S. Longhi, and L. Sartini, "Sonar and video data fusion for robot localization and environment feature estimation," presented at the *44th IEEE Conference on Decision and Control and European Control Conference*, Dec. 2005.
- [15] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Slam: A factored solution to the simultaneous localization and mapping problem," presented at the *Proc. AAAI National Conf.*, 2002.
- [16] P. Colaneri and G. de Nicolao, "Multirate LQG control of continuous-time stochastic systems," *Automatica*, vol. 31, pp. 591–596, 1995.
- [17] J. Tornero, R. Piza, P. Albertos, and J. Salt, "Multirate LQG controller applied to self-location and path tracking in mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2001, pp. 625–630.
- [18] D. Lee and M. Tomizuka, "Multirate optimal state estimation with sensor fusion," in *Proc. American Control Conf.*, 2003, pp. 2887–2892.
- [19] P. Khargonekar, K. Poolla, and A. Tannenbaum, "Robust control of linear time-invariant plants using periodic compensation," *IEEE Trans. Automat. Contr.*, vol. AC-30, pp. 1088–1085, 1985.
- [20] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that probably converges," presented at the *Int. Joint Conf. Artificial Intelligence 2003*.
- [21] G. Bourhis, O. Horn, O. Habert, and A. Pruski, "An autonomous vehicle for people with motor disabilities," *IEEE Robot. Automat. Mag.*, vol. 7, no. 1, pp. 20–28, 2001.
- [22] S. Fioretti, T. Leo, and S. Longhi, "A navigation system for increasing the autonomy and the security of powered wheelchairs," *IEEE Trans. Rehab. Eng.*, vol. 8, no. 4, pp. 490–498, 2000.
- [23] E. Prassler, J. Scholz, and P. Fiorini, "A robotic wheelchair for crowded public environments," *IEEE Robot. Automat. Mag.*, vol. 7, no. 1, pp. 38–45, 2001.
- [24] G. Araujo and M. Aldon, "Line extraction in 2D range images for mobile robotics," *J. Intell. Robot. Sys.*, vol. 40, no. 3, pp. 267–297, 2004.
- [25] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York: John Wiley, 2001.
- [26] G. Araujo and M. Aldon, "Optimal mobile robot pose estimation using geometrical maps," *IEEE Trans. Robot. Automat.*, vol. 18, no. 1, pp. 87–94, 2002.
- [27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.
- [28] A. Doucet, "On sequential simulation-based methods for bayesian filtering," CUED/F-INFENG/TR.310, Dept. of Eng. Univ. Cambridge, UK Tech. Rep., 1998.

- [29] A. Doucet, J. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods In Practice*. Berlin, Germany, Springer, 2001.
- [30] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter." CUED/F-INFENG/TR 380, Eng. Dept. Cambridge Univ. UK Tech. Rep., 2000.
- [31] L. Armesto and J. Tornero, "Slam based on kalman filter for multirate fusion of laser and encoder measurements," in *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 2004, pp. 1860–1865.
- [32] A. Huster and S. Rock, "Relative position sensing by fusing monocular vision and inertial rate sensors," in *Proc. Int. Conf. on Advanced Robotics*, 2003, pp. 1562–1567.
- [33] L. Armesto, S. Chroust, M. Vincze, and J. Tornero, "Multi-rate fusion with vision and inertial sensors," in *Proc. Int. Conf. on Robotics and Automation*, 2004, pp. 193–199.
- [34] J. Tornero, Y. Gu, and M. Tomizuka, "Analysis of multi-rate discrete equivalent of continuous controller," in *Proc. American Control Conf.*, 1999, pp. 2759–2763.
- [35] J. Tornero and M. Tomizuka, "Dual-rate high order hold equivalent controllers," in *Proc. American Control Conf.*, 2000, pp. 175–179.
- [36] ———, "Modeling, analysis and design tools for dual-rate systems," in *American Control Conference*, 2002, pp. 4116–4121.
- [37] L. Armesto and J. Tornero, "Dual-rate high order holds based on primitive functions," in *Proc. American Control Conf.*, 2003, pp. 1140–1145.

**Leopoldo Armesto** received the B.Sc. degree in electronic engineering, the M.Sc. degree in control systems engineering, and the Ph.D. degree in automation and industrial computer science from the Technical University of Valencia, Spain, in 1998, 2001, and 2005, respectively. He held a Ph.D. fellowship for three years at the Department of Systems Engineering and Control at the same University, where he is an assistant professor since 2004. He is currently a member of the Robotics and Automation Research Group and also of the Design and Manufacture Institute of the Technical University of Valencia. He was a visiting researcher at the Automation and Control Institute (ACIN) Vienna University of Technology in 2003 and Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione (DIGA) Università Politecnica delle Marche, in 2005. He was awarded second best M.Sc. Spanish National Prize and Best Technical University of Valencia Ph.D. prize.

**Gianluca Ippoliti** received the doctoral degree in electronic engineering in 1996 from the Università Politecnica delle Marche (formerly University of Ancona), Italy. From 1997 to 1998, he was with ISERM Unité 103, Montpellier, France, and then with the University of Montpellier I in the framework of the European research projects CAMARN and MOBINET. In 2002, he received a Ph.D. degree in intelligent artificial systems from the Università Politecnica delle Marche, and from 2002 to 2005, he was a postdoctoral fellow at the Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione, at the same University. Since March 2005, he is an assistant professor at the Università Politecnica delle Marche. His main research interests include switched systems and supervisory control, neural network-based system identification and control, marine vehicle control, and mobile robot localization and control.

**Sauro Longhi** received the doctoral degree in electronic engineering in 1979 from the University of Ancona, Italy,

and postgraduate diploma in automatic control in 1985 from the University of Rome La Sapienza, Italy. From 1980 to 1981, he held a fellowship at the University of Ancona. From 1981 to 1983, he was with the Telettra S.p.A., Chieti, Italy. Since 1983, he has been at the Dipartimento di Elettrotecnica e Automatica of the University of Ancona, now Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione, of the Università Politecnica delle Marche, Ancona. Currently, he holds the position of full professor in control systems technologies and of coordinator of the Ph.D course in intelligent artificial systems at the Università Politecnica delle Marche, Ancona. His research interests include modeling, identification and control of linear systems, control of mobile base robots and underwater vehicles, service robots for assistive applications, Web technology in process control and remote control laboratories, power management in hybrid cars, and cooperative control of autonomous agents. In these fields, he has published more than 180 articles in international journals and conferences. His research has been supported by the Italian Ministry of Education and University, the Italian National Research Council, the Italian Space Agency, and the European Union. He is a member of the Technical Committee on Marine Systems in IFAC. He was the NOC chair of the IFAC Conference on Control Applications in Marine Systems, Ancona, Italy, July 2004, and the IPC chair of the IFAC Conference on Control Applications in Marine Systems, CAMS 2007, Bol, Croatia, September 2007. Since April 2007, he is the general administrator of the academic spin-off IDEA.

**Josep Tornero** received the M.S. degree in systems and control from the University of Manchester, Institute of Science and Technology, in 1982, and the Ph.D. degree in electrical engineering from the Technical University of Valencia, Spain, in 1985. He is currently a professor at the Department of Systems Engineering and Control, Head of the Robotics and Automation Research Group, and also the director of the Design and Manufacture Institute of the Technical University of Valencia. He has been a visiting professor at the CIRSSE (NASA Center for Intelligent Robotics Systems for Space Exploration); the Rensselaer Polytechnic Institute at Troy (New York); and at the Department of Mechanical Engineering at the University of California (Berkeley). He is particularly interested in modeling, control, and simulation of auto-guided vehicles and robot arms; modeling, analysis, and control of multirate sampled data systems; and collision detection or avoidance and automatic trajectory generation. He has participated in many European research projects such as ESPRIT, BRITE, EUREKA, and STRIDE, and in educational projects such as ERAMUS, INTERCAMPUS, ALPHAS, and TEMPUS.

**Address for Correspondence:** Sauro Longhi, Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione, Università Politecnica delle Marche via Brecce Bianche, 60131 Ancona, Italy. E-mail: sauro.longhi@univpm.it.

## *A National Perspective on the Needs, Themes, and Major Groups*

BY NICK BARNES AND  
ALEXANDER ZELINSKY

# Robotics Research in Australia

Australia has a strong robotics research community for the size of the population, often only third in the number of papers at major robotics gatherings after the United States and Japan [1]. This may be partly due to the unique needs of the Australian economy, creating new domains for the applications of robotics. The Australian economy has a strong basis in agriculture and mining, but the manufacturing sector plays a less significant role than it did historically. As Australia is an island nation, it is highly dependent on transport. Because of the vast size of Australia and its comparatively sparse population, surveillance, surveying large areas, and long distance transport are key needs. Also, having rich off-shore natural resources in tourism, such as the Great Barrier Reef, and in mining, such as oil reserves, there is a need for underwater vehicles. Australia also has natural advantages in robotics research over more populous nations. Infrastructure costs are low, and the low population provides significant space for field tests with minimal danger to people.

This has led to a vibrant interest in robotics research in Australia, with a strong focus on field and service robotics. The sector has strong support from the government, thereby creating a unique robotics research environment. This article summarizes

current work in Australian robotics research. We outline the work in progress at the major research groups across Australia and draw themes emerging in Australian research. We focus on research groups rather than on industrial robotics. For a broad recent review of robotics research worldwide, see the National Science report, which contains some more information on the Australian Centre for Field Robotics (ACFR) and the Commonwealth Scientific and Industrial Research Organisation (CSIRO) Robotics groups. Further information about Australian robotics research can also be seen from the Proceedings of the Annual Conference and the Australasian Conference on Robotics and Automation hosted by the Australian Robotics and Automation Association ([www.araa.asn.au](http://www.araa.asn.au)). The homepage of the Australian Robotics Network (<http://www.araa.asn.au/groups.html>) is another resource.

### Research Groups

The locations of the robotics research groups, apart from the industry, in Australia are shown in Figure 1 (see Table 1).

#### ACFR

ACFR is based at the Faculty of Engineering at the University of Sydney and is Australia's largest robotics group and one of the largest groups in the world. ACFR collaborates widely,



©CARTESIA

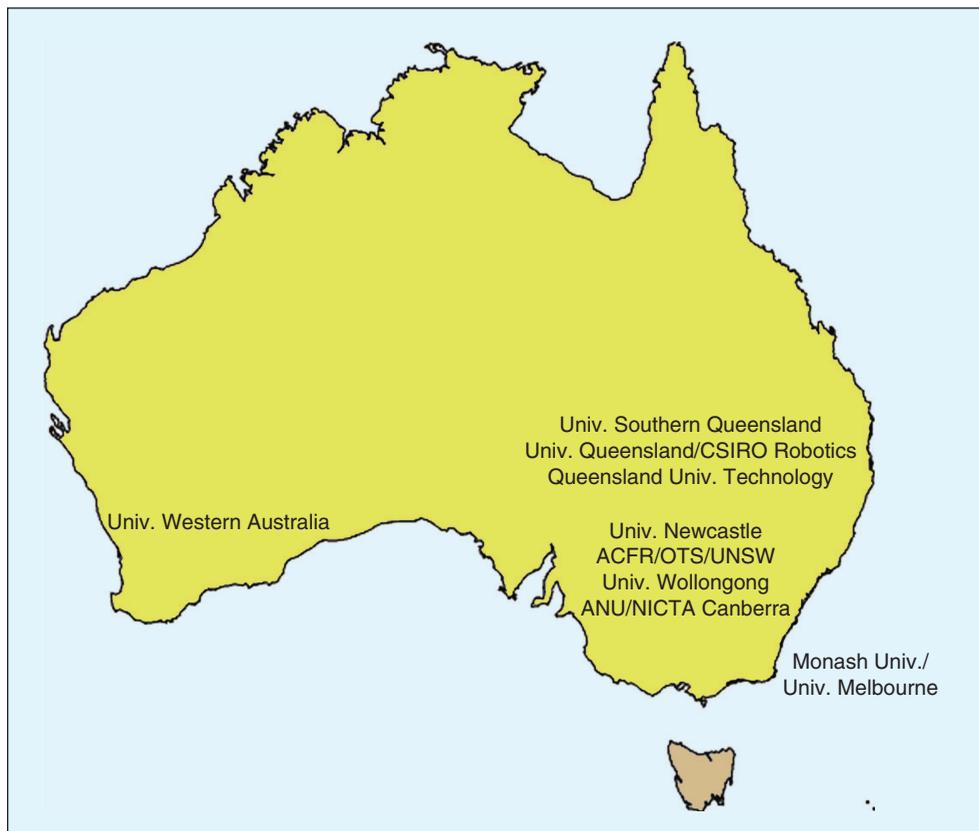
particularly with Australian roboticists from the University of Technology, Sydney (UTS), and the University of New South Wales (UNSW), through the Australian Research Council Centre of Excellence for Autonomous Systems (CAS) (<http://www.cas.edu.au/>) and with Defence Science and Technology

Organisation (DSTO) through the Centre of Expertise in Defence Autonomous Vehicle systems. They focus on field robotics, application of control, sensing, and systems engineering of autonomous machines operating in outdoor and hostile environments. The researchers have a long record of well-respected

research in mapping, particularly using the extended Kalman filter, and, more recently, with particle filters. Other strengths are decentralized data fusion; the design, development, and use of millimeter-wave radar for robotic applications, and the commercial application of field robotics. Notable field applications include autonomous port container vehicles (see Figure 2), outdoor mapping vehicles, and mining applications. Further, they are currently focusing on underwater mapping research, flying vehicles, and the development of control systems.

### CSIRO Robotics

Multiple research groups within CSIRO have shown interest in robotics, the largest being the CSIRO Information Communication Technology Centre Robotics group. The group specializes in reactive control



**Figure 1.** Robotics research groups across Australia [courtesy Geoscience Australia, copyright Commonwealth of Australia (Geoscience Australia)].

**Table 1. Robotics research groups in Australia.**

The Australian Centre for Field Robotics (ACFR)	<a href="http://www.acfr.usyd.edu.au/">www.acfr.usyd.edu.au/</a>
The Australian National University (ANU)	<a href="http://cecs.anu.edu.au/">cecs.anu.edu.au/</a> , <a href="http://www.rsbs.anu.edu.au/ResearchGroups/VIS/index.php">www.rsbs.anu.edu.au/ResearchGroups/VIS/index.php</a>
Australian Research Council Centre of Excellence for Autonomous Systems (CAS)	<a href="http://www.cas.edu.au/home.html">www.cas.edu.au/home.html</a>
Commonwealth Scientific and Industrial Research Organisation (CSIRO) Robotics	<a href="http://www.ict.csiro.au/robotics/">www.ict.csiro.au/robotics/</a>
Defence Science and Technology Organisation (DSTO)	<a href="http://www.dsto.defence.gov.au">www.dsto.defence.gov.au</a>
Monash University	<a href="http://www.ecse.monash.edu.au/centres/irrc/index.php">www.ecse.monash.edu.au/centres/irrc/index.php</a>
National Information Communication Technology Australia (NICTA), Canberra Laboratory	<a href="http://nicta.com.au/director/research/programs/asst.cfm">nicta.com.au/director/research/programs/asst.cfm</a>
Queensland University of Technology (QUT)	<a href="http://www.bee.qut.edu.au/research/projects/quav">www.bee.qut.edu.au/research/projects/quav</a>
University of Melbourne	<a href="http://www.mame.mu.oz.au/groups/index.html">www.mame.mu.oz.au/groups/index.html</a>
University of New South Wales (UNSW)	<a href="http://www.ai.cse.unsw.edu.au/robo/robo.html">www.ai.cse.unsw.edu.au/robo/robo.html</a>
University of Newcastle	<a href="http://robots.newcastle.edu.au">robots.newcastle.edu.au</a>
University of Queensland (UQ)	<a href="http://www.itee.uq.edu.au/~cis/">www.itee.uq.edu.au/~cis/</a>
University of Technology, Sydney (UTS)	<a href="http://www.eng.uts.edu.au/research/Capabilities/MechInt.html">www.eng.uts.edu.au/research/Capabilities/MechInt.html</a>
University of Southern Queensland	<a href="http://www.usq.edu.au/users/billings/">www.usq.edu.au/users/billings/</a>
University of Western Australia	<a href="http://www.mech.uwa.edu.au/jpt/brochure.html">www.mech.uwa.edu.au/jpt/brochure.html</a>
University of Wollongong	<a href="http://www.uow.edu.au/~phillip/rolab/robotics.html">www.uow.edu.au/~phillip/rolab/robotics.html</a>

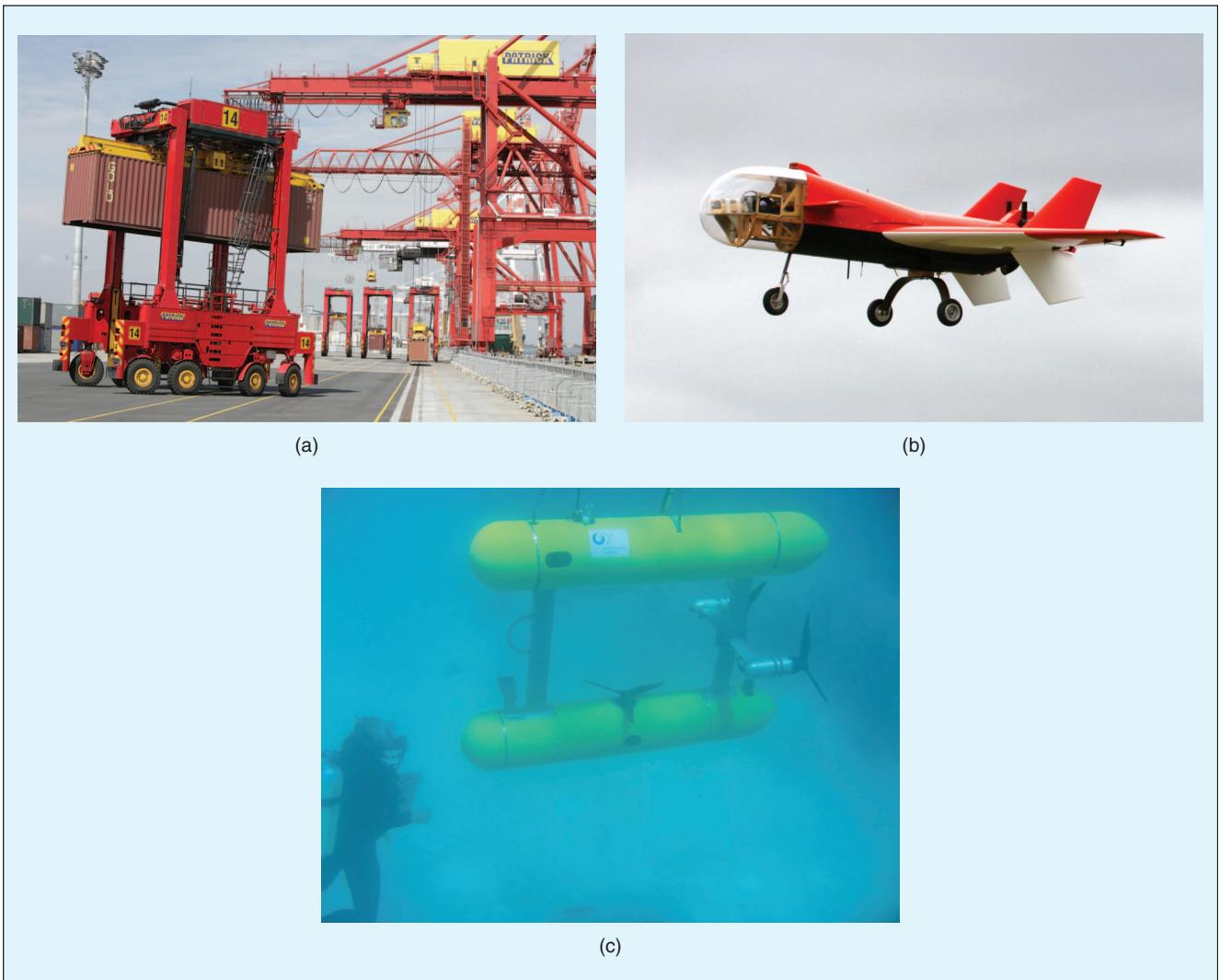
while using only very approximate world models. They are a leading group in visual servoing, particularly for real-world applications. The group is a world leader in robotics and automation in mining applications, having automated the swing control of a 5,000-t dragline [5], in addition to work on rope-shovel excavators, haulage equipment, and explosive loading. They have a strong focus on visual control of mobile robots, including flying robots, underwater vehicles, and outdoor vehicles, and recently on a new program of research in wireless sensor and actuator networks for agriculture, marine, mining, and environmental applications. CSIRO Robotics collaborates with the Australian Research Centre for Aerospace Automation along with researchers from Queensland University of Technology (QUT), as well as with groups at University of Queensland (UQ) and Australian National University (ANU).

A group at Food Science Australia (FSA), a joint venture of CSIRO and the Victorian Government, has developed robotic meat processing equipment. A remotely controlled vehicle for underground coal mine reconnaissance was also developed by CSIRO Exploration and Mining.

### **ANU Campus Canberra, ANU, and NICTA (Canberra Research Laboratory)**

At the ANU campus in Canberra, several separate groups include robotics in their research themes, with collaborations across the groups.

The Visual Sciences group at the Research School for Biological Sciences (RSBS) studies the behaviour of insects and has developed robots as a testbed for understanding insect behavior, as well as robotic systems based on insect behavior as an outcome in itself. They are best known for their work on honeybee navigation. The Vision Science, Technologies and Applications (VISTA) program at NICTA is developing driver assistance systems for smart cars and novel algorithms for vision systems using biologically inspired algorithms and sensors from human and insect vision. The NICTA group collaborates with RSBS, for research on biologically inspired sensors, as well as with the Research School for Information Systems Engineering (RSISE) on real-time driver assistance systems. The driver assistance project is a collaboration with CSIRO Robotics on the use of



**Figure 2.** ACFR research: (a) the autonomous straddle carrier, an automated stevedoring vehicle, which is in operation since 2002 in Brisbane, developed by ACFR, Patrick Corporation, and Kalmar Industries; (b) ACFR's Brumby MKIII UAV; (c) ACFR's underwater vehicle. (Images courtesy of Hugh Durrant Whyte.)

infrared cameras for driver assistance. Within RSISE, researchers are developing a team of submersibles and are studying manipulator dynamics. Researchers of the Faculty of Engineering and Information Technology study flying robots and visual servoing on behaviors using insect-inspired control in collaboration with the groups at RSBS, CSIRO Robotics, and NICTA.

### **Monash University**

The Intelligent Robotics Research Centre is part of the Australian Research Council (ARC) Centre for Perceptive and Intelligent Machines in Complex Environments, and some of their key areas of recent research are swarm robotics, tactile sensing, chemical trail sensing robots, and the use of highly accurate sonar and laser range finders for problems such as SLAM and humanoid robots.

### **UQ**

The Mechatronics group of the Division of Complex and Intelligent Systems Research, Department of Electrical Engineering, UQ, boasts Australia's most developed walking humanoid robot. They have succeeded in the development of the F180 league of Robocup, and active research in biologically inspired navigation is underway. The group has strong ties with CSIRO Robotics.

### **UTS**

The key areas of robotics research undertaken by the Mechatronics and Intelligent Systems laboratory are biomedical devices, smart autonomous robots, and highly efficient motors. Field robotics research has included ports and underground mines through CAS in collaboration with ACFR.

### **Other Groups**

One of the first projects in Australia was aimed at automated robotic sheep shearing, at the University of Western Australia. More recent work is within the fields of surgical robots and demining.

The Artificial Intelligence group at UNSW has a strong history of research in artificial intelligence. They are partners with CAS and focus on multiagent task planning, learning, and vision-based collision avoidance. They have successfully participated in the Four-Legged League in the Robocup competition. The Newcastle Robotics Lab, University of Newcastle, were winners in the same competition in 2006. Their main focus in robotics is currently machine learning.

At QUT, the aerospace avionics group has developed unmanned aerial vehicles (UAVs), global navigation satellite systems, avionics payloads, and satellite-based sensors. Other work within QUT includes vehicle guidance in mining and underwater vehicles. At the Department of Mechanical and Manufacturing Engineering, the University of Melbourne, mechatronics researchers are involved in the development of various types of dynamic systems and control research, in collaboration with the Automotive industry for developing bivariate control systems.

DSTO conducts and funds various research projects in autonomous vehicles including ground, aerial, and underwater vehicles. This includes the Centre of Expertise at ACFR.

Research at the Intelligent Robotics Laboratory at the University of Wollongong includes work on outdoor robotics and sensing for horticulture and flying robots. Agricultural applications have also been the focus of research at the University of Southern Queensland. The group has links with the National Centre for Engineering in Agriculture.

## **Research Themes**

Australia's needs have had a strong impact on what has always been a research community with a strong pragmatic approach. In the early days, the needs of the wool industry led to a sheep shearing robot. More recently, the needs of the vast size of Australia have led to research in remote observation and transport. Because of these needs, along with those of a significant mining industry, the emphasis of Australian robotics research is field robotics. Despite the size, a significant proportion of the population lives in remote communities, and so for the provision of medical facilities, research in medical robotics is underway. Facilitating such research requires theoretical underpinning, and the strongest of these areas has been SLAM and multirobot coordination. Biologically inspired robotics research has risen partly due to service areas such as transport and military applications. However, this work emerged from the strength of Australian research in insects. Insect research is important to an island nation with a strong agricultural sector.

### **Remote Observation**

As a large sparsely populated island nation, Australia has vast tracts of land and sea that present interesting problems in monitoring. This includes environmental monitoring, as well as remote surveying and surveillance. Australian robotics research currently focuses on underwater robots and UAVs.

The Great Barrier Reef is a major environmental site and a backbone of Australia's tourist industry, but it is at the same time a delicate ecosystem that comes under threat from both human and natural forces. CSIRO Robotics has developed a new submersible robot for environmental monitoring of the reef [7]. For a broader range of underwater monitoring, ANU is developing a school of small submarines (see Figure 3), and recent work includes scheduling for optimal communication through a multihop wireless network across the school [22]. Bridging the gap between land and water, researches at ACFR have modeled and are addressing control issues of an amphibious vehicle for autonomous control [10].

Off the ground, ACFR is using a Brumby MkIII UAV (see Figure 2) for feature tracking and mapping using vision and global positioning system (GPS)/inertial navigation system (INS) [13]. CSIRO's helicopter research program demonstrated the first GPS-free vision-based hover and, more recently, has developed research in urban canyon navigation [11] and a new cable-array robot for air vehicle simulation, for easing the cost of development of air vehicle guidance systems [26]. At ANU, Mahony is researching visual servoing for aerial vehicles [15] and quad rotor flying vehicles.

### **Transport and Mining**

Australia is large geographically but is sparsely populated. So efficient and safe transport has always been a key concern. With

Australians driving large distances between capital cities, driver support is an important theme in Australian robotics research. NICTA/ANU have a long research history in driver assistance. Sign detection is a more recent theme, with work on speed signs and giveaway/stop signs [3]. In their work Fletcher et al. [9] have correlated driver gaze with sign detection to assess whether a driver has looked at a sign on the road, and they have improved sign classification performance through super-resolution.

Australian robotics research has a proud history in industrial transport with Durrant-Whyte's well-known early work on container port transport [8] now being adapted into a new working port in Australia [18] (see Figure 2). Mining is a major contributor to the Australian economy and so efficient and safe extraction and transport in mining have been major areas of research for CSIRO and ACFR. In a recent work, Kloos et al. [14] have used radio frequency identification device (RFID) tags to ensure the safety of personnel in mining environments, particularly around large mining trucks.

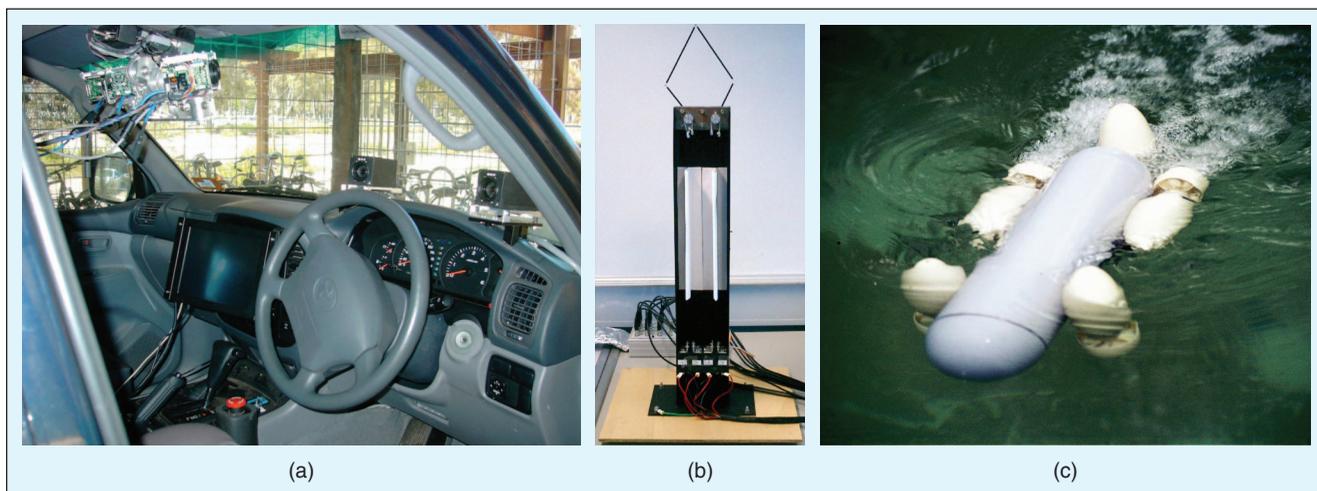
### Medical Robotics and Manipulation

Given that a significant number of Australians live in isolated towns, a key problem has been the provision of medical facilities to remote communities. Robotics-assisted surgery has much to offer in this regard. The University of Western Australia is studying robotics surgery, including modeling brain tissue deformation due to the force of surgical tools [17].

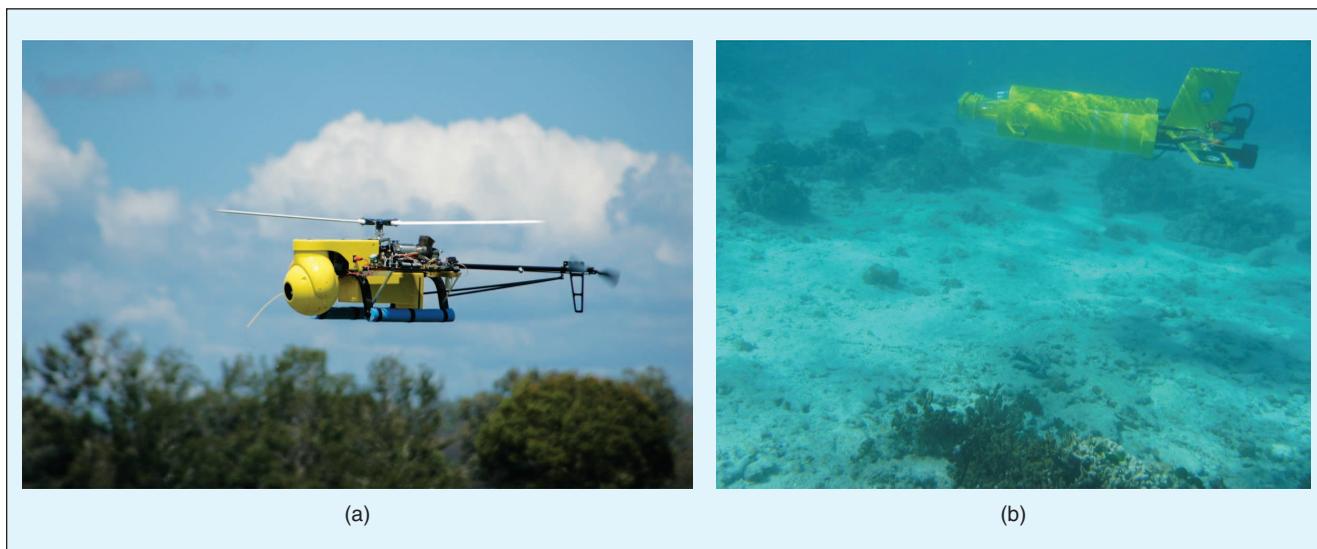
At the same time, fundamental research continues on manipulators for applications such as surgery. For example at ANU, Teh and Featherstone are developing fast motion control for shape memory alloy actuators [24].

### Simultaneous Localization and Mapping (SLAM)

ACFR contributed pioneering work on SLAM (e.g., [6]) and continues to build on this research. Recent work includes the performance of particle filter-based FastSLAM and EKF-based methods [2]. ACFR has applied SLAM in very unstructured, outdoor, and underwater environments.



**Figure 3.** ANU/NICTA research: (a) the Smart Vehicle project, (b) an SMA actuated pantograph robot, and (c) a first generation seraphina from the school of autonomous underwater vehicles. (Images courtesy of Roy Featherstone, Uwe R. Zimmer, and Felix Schill.)



**Figure 4.** CSIRO robotics: (a) CSIRO helicopter platform from UAV Vision, Sydney, with CSIRO avionics, and (b) CSIRO starbug underwater vehicle. (Images courtesy of Peter Corke, Johathan Roberts, Matthew Dunbabin, and Stefan Hrabar.)

## Multirobot Research

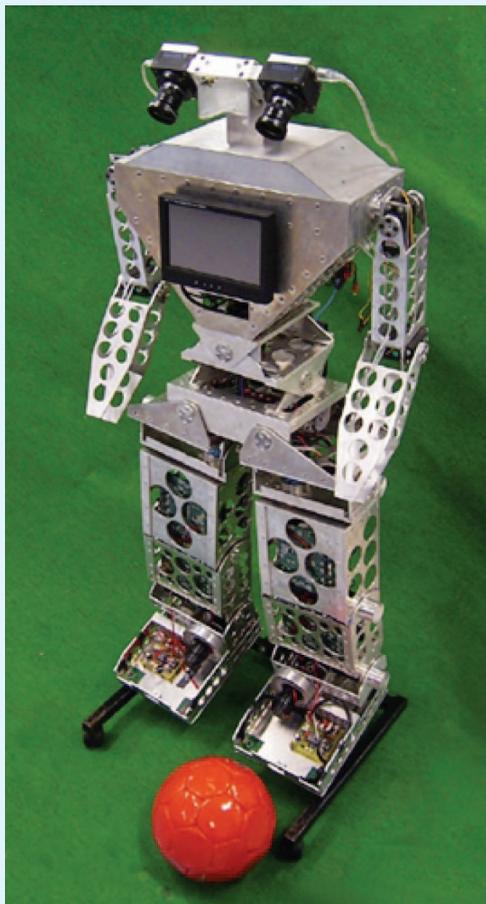
Based on the needs of the continent, Australian research has considered teams of robots. A major branch of research at ACFR is considering decentralized data fusion; for instance, using Gaussian mixture models [25]. CSIRO is incorporating

its underwater robots into an underwater sensor network [27], including docking of AUVs and cooperative behaviors. ANU's underwater robot school has been mentioned previously (Figure 4).

Most of the multirobot research in Australia is targeted toward the Robocup competition. Australia has performed well with UNSW/NICTA winning the Four-Legged League three times, with the University of Newcastle winning in 2006, and UQ securing second place in the F180 league. Chalup et al. from the University of Newcastle have demonstrated the use of machine learning in the Four-Legged League [19], while Chang and Wyeth from the UQ have developed probabilistic world modeling for distributed team planning [4].



(a)



(b)

**Figure 5.** Robotics research at UQ: (a) Robocup in the F180 League, 2003, and (b) the "guroo" humanoid robot. (Images courtesy of Gordon Wyeth.)

## Biological Inspiration

Pioneering research in insect biological research at ANU Biological Sciences (e.g., [23]) continues to be an inspiration for robot algorithms. Desert ant navigation has been the inspiration for docking of position, based on the bearing of a set of landmarks. Wei et al. [28] extended this work by docking for full robot pose, including orientation using only bearing information from landmarks. Work has also focused on robotic guidance using a human-like visual sensor, the log-polar camera [29]. At UQ, a continuing project is investigating the navigation of rats, looking specifically at analyses of neurons in the rat hippocampus that appear to represent topological spatial position (Rat-SLAM). Recent work has addressed goal-directed navigation [16]. The same group has also developed a humanoid robot [12] (see Figure 5). At Monash University, research has addressed the issues of multisensor synergies on humanoids [21] and continued work on robotic odor sensing, applying this to the guidance of underground chemical sources [20].

## Conclusion

Robotics research in Australia is well-grounded in the pragmatic attention to the needs of the nation. Some of the applications, particularly in mining and port automation, are among the leading examples of industrial applications of robotics worldwide. At the same time, Australian researchers are among the leaders in several fields of fundamental research within robotics, in areas including SLAM, distributed data fusion, visual control, and vision in vehicles.

## Acknowledgments

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

## Keywords

Robotics research, Australian robotics, robotic applications.

## References

- [1] ARC research network in robotics (2004). [Online]. Available: <http://www.araa.asn.au/groups.html>

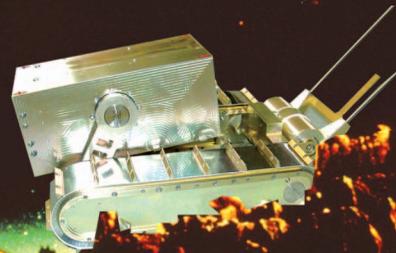
- [2] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the fastslam algorithm," in *Proc. ICRA2006, IEEE Int. Conf. Robotics and Automation*, 2006, pp. 424–429.
- [3] N. Barnes and G. Loy, "Real-time regular polygonal sign detection," in *Proc. 5th Int. Conf. Field and Service Robotics (FSR05)*, 2005, pp. 55–66.
- [4] M. Chang and G. F. Wyeth, "Probabilistic world modeling for distributed team planning," in *Proc. Int. Conf. on Intelligent Robots and Systems (IROS 2004)*, 2004, vol. 2, pp. 1426–1431.
- [5] P. Corke, J. Roberts, and G. Winstanley, "Vision-based control for mining automation," *Robot. Automat. Mag.*, vol. 5, no. 4, pp. 44–49, Dec. 1998.
- [6] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localisation and mapping building (SLAM) problem," *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, June 2001.
- [7] M. Dunbabin, J. Roberts, K. Usher, G. Winstanley, and P. Corke, "A hybrid AUV design for shallow water reef navigation," in *Proc. IEEE Int. Conf. Robotics and Automation*, Barcelona, Apr. 2005, pp. 2117–2122.
- [8] H. F. Durrant-Whyte, "An autonomous guided vehicle for cargo handling operations," *J. Robot. Res.*, vol. 15, no. 5, pp. 407–440, Oct. 1996.
- [9] L. Fletcher, G. Loy, N. Barnes, and A. Zelinsky, "Correlating driver gaze with the road scene for driver assistance," *Robot. Auton. Sys.*, vol. 52, no. 1, pp. 71–84, 2005.
- [10] Q. P. Ha, T. H. Tran, S. Scheding, G. Dissanayake, and H. F. Durrant-Whyte, "Control issues of an autonomous vehicle," in *Proc. 22nd IAARC/CIB/IEEE/IFAC/IFR Int. Symp. Automation and Robotics in Construction*, Ferrara, Italy, 2005, pp. 1–7.
- [11] S. Hrabar, G. Sukhatme, P. Corke, K. Usher, and J. Roberts, "Combined optic-flow and stereo-based navigation of urban canyons for a UAV," in *Proc. Int. Conf. on Intelligent Robots and Systems (IROS)*, Alberta, Aug. 2005, pp. 3309–3316.
- [12] D. Kee and G. Wyeth, "Biologically inspired joint control for a humanoid robot," in *Proc. IEEE Conf. Humanoid Robotics (Humanoids 2004)*, Los Angeles, CA, Nov. 2004, pp. 385–401.
- [13] J. Kim, M. Ridley, E. Nettleton, and S. Sukkarieh, "Real-time experiment of feature tracking/mapping using a low-cost vision and GPS/INS system on a UAV platform," *J. Global Positioning Sys.*, vol. 3, no. 1–2, pp. 167–172, 2005.
- [14] G. Kloos, J. Guivant, and E. Nebot, "Range based localisation for mining safety—Radio frequency identification device (RFID) based system to detect the presence of personnel in close proximity to large machines," presented at the 2005 Australian Mining Technology Conf., Sept. 2005.
- [15] R. Mahony and T. Hamel, "Image based visual servo control of aerial robotic systems using linear image features," *IEEE Trans. Robotics*, vol. 21, no. 2, pp. 227–239, Apr. 2005.
- [16] M. Milford, G. Wyeth, and D. Prasser, "Efficient goal directed navigation using ratslam," in *Proc. ICRA2005, IEEE Int. Conf. Robotics and Automation*, Barcelona, Spain, 2005, pp. 1097–1102.
- [17] K. Miller, Z. Taylor, and W. L. Nowinski, "Towards computing brain deformations for diagnosis, prognosis and neurosurgical simulation," *J. Mechanics Med. Biol.*, vol. 5, no. 1, pp. 105–121, 2005.
- [18] G. Nelmes, "Container port automation," in *Proc. 5th Int. Conf. Field and Service Robotics (FSR05)*, 2005, pp. 3–8.
- [19] M. J. Quinlan, S. K. Chalup, and R. H. Middleton, "Application of SVMs for colour classification and collision detection with AIBO robots," in *Advances in Neural Information Processing Systems (NIPS) 16*, 2004, pp. 635–642.
- [20] R. A. Russell, "Robotic location of underground chemical sources," *Robotica*, vol. 22, no. 1, pp. 109–115, 2004.
- [21] R. A. Russell, G. Taylor, L. Kleeman, and A. H. Purnamadajaja, "Multi-sensory synergies in humanoid robotics," *Int. J. Humanoid Robot.*, vol. 1, no. 2, pp. 289–314, 2004.
- [22] F. Schill, U. R. Zimmer, and J. Trumpf, "Towards optimal tdma scheduling for robotic swarm communication," in *Proc. Towards Autonomous Robotic Systems (TAROS)*, London, UK, Sept. 2005, pp. 197–203.
- [23] M. V. Srinivasan, S. Zhang, S. M. Altwein, and J. Tautz, "Honeybee navigation: Nature and calibration of the 'odometer,'" *Science*, vol. 287, no. 5454, pp. 851–85, Feb. 2000.
- [24] Y. H. Teh and R. Featherstone, "A new control system for fast motion control of SMA actuator wires," presented at the Shape Memory And Related Technologies Conf. (SMART 2004), Nov. 2004.
- [25] B. Upcroft, L. L. Ong, S. Kumar, M. Ridley, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte, "Rich probabilistic representations for bearing only decentralized data fusion," in *Proc. IEEE Conf. Information Fusion*, 2005, pp. 1054–1061.
- [26] K. Usher, G. Winstanley, P. Corke, and D. Stauffacher, "Air vehicle simulator: An application for a cable-array robot," in *Proc. 2004 Int. Conf. Robotics and Automation*, Barcelona, Spain, Apr. 2005, pp. 2253–2258.
- [27] I. Vasilescu, K. Kotay, D. Rus, P. Corke, M. Dunbabin, and P. Schmidt, "Data collection, storage and retrieval with an underwater sensor network," in *Proc. IEEE SenSys*, 2005, pp. 154–165.
- [28] R. Wei, D. Austin, and R. Mahony, "Biomimetic application of desert ant visual navigation for mobile robot docking with weighted landmarks," *Int. J. Intell. Sys. Technol. Appl.*, vol. 1, no. 1/2, pp. 174–190, 2005.
- [29] A. Yeung and N. Barnes, "Efficient active monocular fixation using the log-polar sensor," *Int. J. Intell. Sys. Technol. Appl.*, vol. 1, no. 1/2, pp. 157–173, 2005.

**Nick Barnes** received his B.Sc. degree with honors in 1992 and a Ph.D. degree in computer vision for robot guidance in 1999 from the University of Melbourne. From 1992 to 1994, he worked as a consultant in the information technology industry. In 1999, he was a visiting research fellow at the LIRA Lab at the University of Genoa, Italy. From 2000 to 2003, he was a lecturer at the Department of Computer Science and Software Engineering, University of Melbourne. Since 2003, he has been with NICTA's Canberra Research Laboratory. He is currently the principal researcher and research group manager in computer vision. His research interests include visual dynamic scene analysis, computational models of biological vision, feature detection, vision for vehicle guidance, and medical image analysis.

**Alexander Zelinsky** is a research leader in mobile robotics, computer vision, and human-machine interaction. He is the group executive of information and communication sciences and technology at CSIRO. Prior to joining CSIRO, he was the CEO and cofounder of Seeing Machines. He was professor and head of the Department of Systems Engineering from 1996 to 2000. He served on the IEEE Robotics and Automation Society (RAS) AdCom (2005–2007), and he is the vice-president of the RAS Industrial Activities Board (2008–2009). His contributions have been recognized by awards in Australia and internationally, which include the Australian Engineering Excellence Awards (1999 and 2002), US R&D magazine Top 100 Award (2001), Clunies-Ross Award (2005), and Technology Pioneer at the World Economic Forum (2002–2004). He is an elected fellow of the Australian Academy of Technological Sciences and Engineering (2002) and an elected Fellow of the IEEE (2008).

**Address for Correspondence:** Nick Barnes, NICTA, Building A, 7 London Circuit, Canberra, ACT 2601, Australia. E-mail: nick.barnes@nicta.com.au.

# Nanokhod Exploration Rover



©PHOTODISC (NANOKHOD IMAGE, COURTESY OF ESA)

## *A Rugged Rover Suited for Small, Low-Cost, Planetary Lander Missions*

**T**he European Space Agency (ESA) is currently concluding a dedicated technology development to mature and finalize the Nanokhod microrover [1], [2] into a complete engineering model with integrated geochemical payload package. The goal of this development is to build and test the Nanokhod such that it can serve a wide range of mission applications in the future. The rover will be able to deliver scientific payloads to not only atmospheric planets such as Mars but also environmentally more extreme nonatmospheric celestial bodies such as Mercury or the Moon.

With respect to current well-known exploration rovers, such as the Pathfinder rover Sojourner (11 kg) [3] or the more recent Mars exploration rovers (MERs) Spirit and Opportunity (174 kg each) [4], the Nanokhod exploration rover belongs to a totally different class, with a total system mass less than 3 kg. This includes the science payload instruments.

Another rover in the same class was developed by the National Aeronautics and Space Administration (NASA) for the Hayabusa mission [5]. This contribution was canceled while the hardware was already in an advanced development stage.

The Nanokhod rover is a payload-efficient system that can be used to send a maximum amount of scientific payload to a planetary surface with the minimum rover mass. The high payload to total mass ratio of about 0.27 can enable low-cost planetary surface exploration missions where total available mass and power is extremely restricted.

To arrive at the relatively high payload mass efficiency, various tasks such as navigation and power are shared between the rover and the lander via a tether connection. Furthermore, instead of a wheeled locomotion system, tracks provide high mobility and robust locomotion between rocks while keeping the chassis design simple. The Nanokhod can carry out scientific investigations in the vicinity of a static scientific lander within a radius of about 50 m (Figure 1).

Digital Object Identifier 10.1109/MRA.2008.917888

**BY ANDRÉ SCHIELE, JENS ROMSTEDT, CHRISTOPHER LEE, HARTMUT HENKEL,  
SABINE KLINKNER, REINHOLD BERTRAND, RUDOLF RIEDER, RALF GELLERT,  
GÖSTAR KLINGELHÖFER, BODO BERNHARDT, AND HARALD MICHAELIS**

In situ analysis of the surface chemistry and mineralogy is the primary scientific objective of the Nanokhod. The implemented geochemical instrument package supports the analysis of soil and rock elemental composition and content of iron-bearing minerals. In particular, the 800-g Nanokhod payload consists of the alpha-particle X-ray spectrometer (APXS) to measure chemical composition, the Mössbauer (MB) spectrometer (MIMOS2) to measure mineralogical composition, and a microscopic and far-field camera (MIROCAM) to obtain images of surface structure and morphology. In addition, the MIROCAM can provide surface reflectance spectroscopic data.

With respect to prior versions of the Nanokhod, the current redesign aims at building a full engineering model, ready to be extensively tested on system level.

For the first time, strict miniaturization aims at fully integrating the rover with the scientific payloads. In particular, the flight-qualified European instruments developed and used for the MER missions are redesigned to fit into the Nanokhod. Front ends of the APXS, MIMOS2, and MIROCAM are mechanically integrated into the rover payload compartment. The instrument back ends are integrated electronically. At the same time, the instruments are undergoing a major redesign to offer purely digital interfaces that allow full digital signal processing. Processing can then be done exclusively in the rover controller within the payload compartment.

Originally, the redesigned Nanokhod rover should have been a payload element of the BepiColombo Mission lander module. Despite the cancellation of the landing element for cost reasons in 2004, and thus the loss of an immediate flight opportunity for Nanokhod, the key objectives for the design were kept. This allowed us to gain important experience in the design, development, and testing of miniaturized integrated robotic vehicles for extreme environments, which can be of benefit to potential future missions with similar requirements.

The goal of this article is to present the redesign that was required to prepare the Nanokhod rover for use in harsh planetary environments determined by temperatures approaching  $-183^{\circ}\text{C}$ , dusty regolith surfaces, and the absence of atmosphere. The presented design will, however, not only be applicable to this type of mission but also naturally cope with applications in environmentally more benign places.

## The Nanokhod Rover

### Design Overview

The resulting rover design is presented in Figure 2. Like its predecessors, the rover is composed of two tracked locomotion units, which are rigidly connected via a tether bridge.

The payload cabin (PLC) is attached to the two locomotion units via two lever arms. Two motor or gear assemblies, a lower and an upper one, actuate the left lever arm and provide two degrees of freedom for PLC positioning. The right lever arm contains the cable harness to the PLC. The locomotion units, PLC, tether unit, and the right lever arm are interconnected with Micro D-Sub Miniature (MDM) connectors to provide modularity and to greatly facilitate assembly and maintenance.

The entire stowed volume of the Nanokhod rover is  $240 \times 165 \times 65 \text{ mm}^3$ . The total system mass is 2.95 kg without margin, whereas the payload mass alone is 800 g without margin. Power consumed by the rover, as seen from the lander, has an estimated peak at 5.7 W during locomotion.

### Rover Mechanical Design

With respect to prior developments that were aimed at improving the compatibility of some subsystems with low temperature [6], in this work, the entire Nanokhod design was iterated to eliminate all remaining shortcomings in the design. Fitting, gear, bearing, and motor design, as well as pairing of materials, were tailored appropriately to the extreme low-temperature



Figure 1. The Nanokhod engineering qualification model (courtesy ESA).

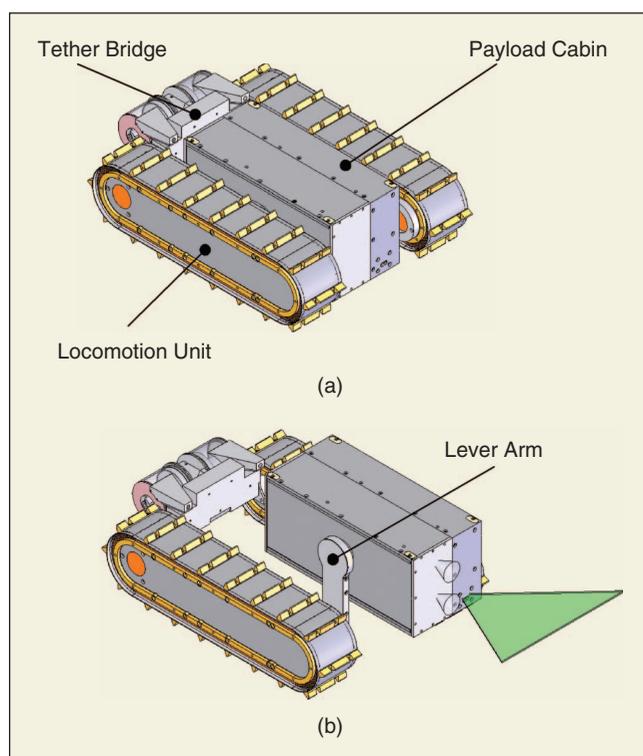


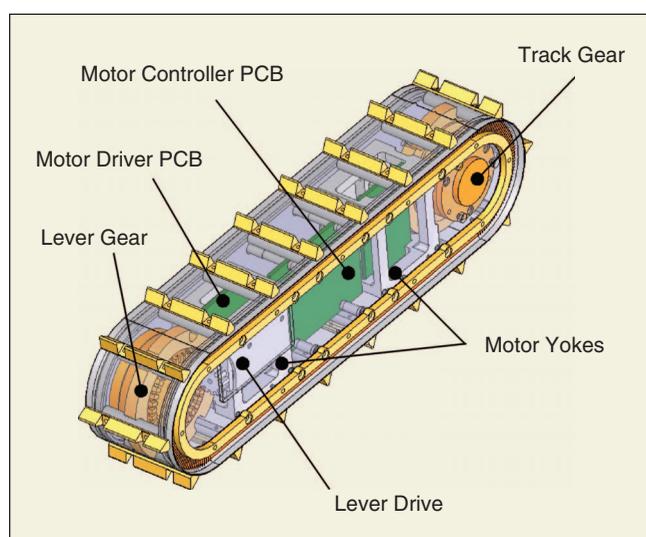
Figure 2. Computer-aided design (CAD) model of the redesigned Nanokhod rover. (a) Stowed configuration. (b) Unfolded payload cabin (PLC) with outline of instrument field of views.

application. Prior to incorporation into the design, low-temperature functional tests have been performed to verify subsystem-level performance. The structural design was improved to withstand the expected vibrations and shocks occurring during launch and landing.

### Locomotion Units

Both locomotion units consist of two walls each, supporting the track foil around their circumference. The motor yokes (Figure 3) connect both walls structurally. The track runs over two sprocket gears, of which only one is actuated per unit.

The right locomotion unit contains the motors and gears for forward and backward actuation of the right track. Furthermore, it contains printed circuit boards (PCBs) for the drive and control electronics of the motors and the PCB containing the tether interface electronics. The left locomotion unit (Figure 3) contains the motor and gear unit for the actuation of the left track and additionally a motor and gear for actuation of the lower lever arm joint, which positions the payload cab. All



**Figure 3.** CAD model overview of left locomotion unit containing motor gear assemblies for track and lever actuation (the track and outside wall are set transparent).



**Figure 4.** Cryogenic test setup of a new track seal prototype.

drive and control electronic PCBs are again contained inside the locomotion unit housing.

The locomotion unit sealing is achieved at low temperature through a 0.25-mm-thick poly-tetra-fluor-ethylene (PTFE) seal, which is preloaded toward the track foil with a CuBe spring. Seal performance was analyzed by visual checks after cryogenic tests (Figure 4). The seal and the spring are arranged such that temperature-related shrinkage of the PTFE, the track, or the spring does not negatively affect seal performance.

Seal contact pressure amounts to about 0.2 N/mm. This is sufficiently high to provide good sealing performance and sufficiently low to produce minimal friction. The total friction, per locomotion unit, sums to about 470 mNm at the gear output in room temperature. The friction decreases at low temperatures.

### Actuation

The locomotion design is similar for all drives. The drivetrain design is strong enough to enable terrain gradeability of at least 30° in regolith. Furthermore, the design allows the lever drives to move the rover chassis over the PLC, which increases obstacle-climbing capability. Locomotion velocity of the rover is about 3 m/h. The chosen actuator is an AM1020 stepper motor, without grease lubricant, that runs constantly at 5,000 r/min. All gear units of the rover incorporate HFUC-Series harmonic drives (HDs) to reduce backlash to less than 5 arcmin and provide sufficient robustness against vibration and shock loads. For the track actuation, the motor output is coupled via a planetary gear head and a crown gear to the wave generator (WG) of an HFUC HD unit. The flex spline (FS) is fixed to the locomotion unit walls whereas the circular spline (CS) is attached to the drive sprocket. The sprocket assembly is supported with bearings on the gear housing.

The concept of the lower-lever drive is similar, with the difference that the FS is connected directly to the lever whereas the CS is fixed on the locomotion unit walls. The passive sprocket that supports and guides the track is mounted around the CS with bearings allowing it to run free. Sealing between the gear and the walls is provided by PTFE O-rings. The upper-lever drive is arranged similarly, with the CS attached to the PLC structure. Furthermore, the pinion gear is inverted to be more compact. A shaft for a rotational encoder is provided from the WG to the outside of the assembly (Figure 5). The output torque of all motor-gear systems is about 3.2 Nm at the HD exit.

The expected low number of gear revolutions over the rover lifetime allowed relaxation of the safety margins. This enables designing the gears against ratcheting torque, which leads to a further reduction of volume. An HFUC-type gear was selected as it provides a reduced length and generally higher ratcheting values as opposed to the HDUC-Series types. The HFUC 05-08-2A-R-SP was chosen for all gear units. All contact surfaces and the bearings are dry lubricated with Microseal 200-1, which is based on a MoS<sub>2</sub> lubricant. The lubricant meets the Aerospace Material Specifications (AMS) standard STD-2526 and has passed all relevant ESA off-gassing tests. Thermal expansion characteristics were considered in the material selection of the gears. The selected parts consist of 15-5 PH and 17-4 PH, space-qualified stainless steel and 440C for the bearings.

## Payload Cabin

The rover PLC optimally integrates the geochemistry instrument package facility (GIPF), the central rover controller PCBs, and the motor driver or controller PCBs for the upper-lever drive. Figure 6 shows the arrangements of the science payload front ends inside the PLC. All instrument axes are positioned symmetrically such that they can be positioned on the same spot by rotating the upper-lever drive only.

On each corner of the PLC, light-emitting diodes (LEDs) are situated to support the lander-based visual localization system [7]. A ring containing microswitches to detect contact with objects is mounted in front of the MB spectrometer. It allows the detection of the minimum contact force of 1 N required for the MIMOS2.

## Tether Unit

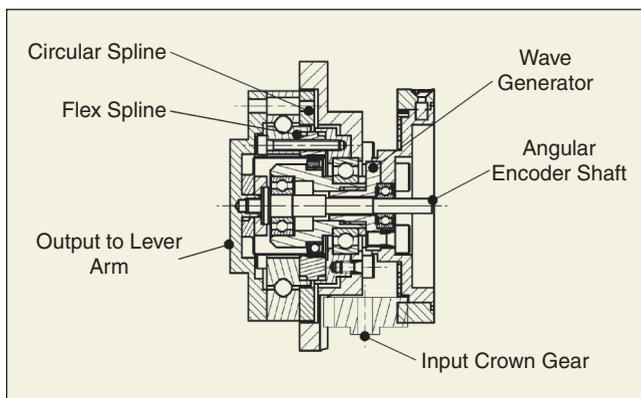
The tether unit has been slightly relocated to increase ground clearance of the rover. Spring deployable tether guides ensure guidance of the unwinding tether and allow the rover to drive backward for at least one body length. The tether spools that contain each 50 m of RF litz can easily be removed for inspection and testing. They are attached via ball bearings to a central axis to provide minimal unwinding resistance. Electrical contact between the unwinding tether and the rover is provided by gold-plated spring-loaded slip rings that are sealed from the environment.

## Lever Arms

The driven left arm consists of two purely structural parts. Either end is connected to the FS of the lever actuation units. The nonactuated right arm passes the rover bus between the PLC and the right locomotion unit. A cable spiral at either end enables a 370° rotation of the lever. On the PLC side, the spiral is included in the arm itself whereas the lower spiral is contained inside the locomotion unit to gain ground clearance through a smaller arm base diameter (Figure 7).

## Rover Electrical Design

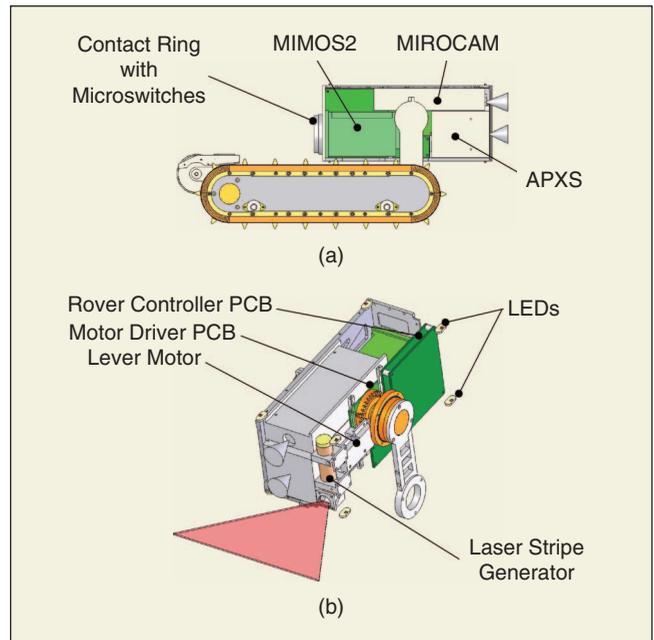
No off-the-shelf electronic parts exist for the required temperature range. This is why a series of component tests has been performed to aid the rover system design. Such tests were performed on all critical components and focused on ascertaining



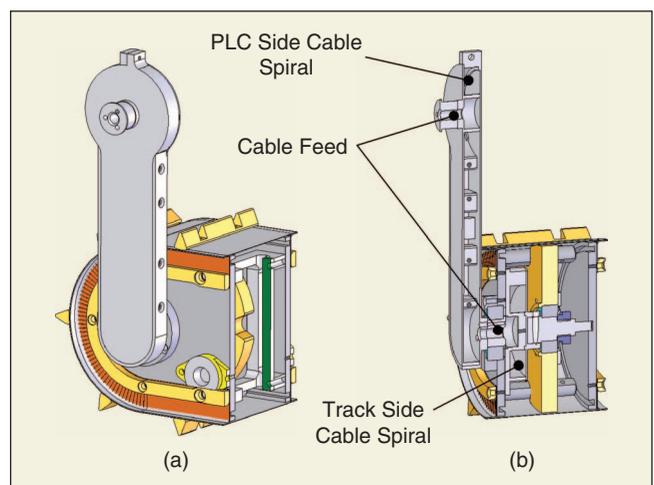
**Figure 5.** Assembly view of drive unit for upper-lever drive inside the PLC. CS, FS, and WG of HD unit are shown.

functionality and assessing parametric variation over the temperature range. Parts selected for the rover must not necessarily show ideal behavior but rather demonstrate a well-defined predictable functionality. This allows establishing design guidelines for compensating or reducing the effects of extreme low temperature on the components. The following general design guidelines are applied:

- ◆ critical signals or signal processing should not be analogue because of the unpredictable gains and offsets over large temperature ranges
- ◆ if an analogue measurement is required, values have to be measured differentially
- ◆ the selected active components should be MOS type rather than bipolar type (e.g., bipolar transistor gain reduces toward zero at temperatures around  $-180\text{ }^{\circ}\text{C}$ )



**Figure 6.** CAD overview of GIPF payload arrangement in PLC. (a) Overview. (b) PLC walls removed to illustrate inside arrangement.



**Figure 7.** CAD assembly view on right lever arm (PLC not shown and locomotion unit partially shown). (a) Overview. (b) Vertical section cut.

- ◆ the dependence on high-accuracy resistance-capacitance timing circuits should be reduced wherever possible.

Furthermore, radiation compatibility to levels of up to a total dose of 30 krd is important. Even though radiation tolerance is not required for the engineering model, all critical components were chosen such that flight-equivalent radiation-hard components exist. The electronic architecture was designed such that the transition to a radiation-hard version would not require any major redesign.

### Tether Communications

Communication between the lander and the rover is performed via tether wires, which carry both power (dc) and communication signals. The tether consists of two wires formed from 30 strands of 0.054-mm Cu-LS RF litz. With respect to prior versions, the increased wire diameter reduces the overall resistance by more than 25%. The 50-m-long wires are enamel insulated and are covered by a fine silk. One wire carries the power supply whereas the other one carries the return. Each wire is again split into two electrically separate bundles to be able to carry differential communication signals. One wire is used for telecommand uplink whereas the other is used for telemetry return. The differential signals are coupled to each wire by transformers. The transceiver circuits convert the character coding generated by the logical units in the rover or lander into electrical signals that can be transmitted over the tether. A differential driver followed by a pulse shaper is used in front of the transmit (Tx) splitter.

The receive (Rx) circuits are responsible to split the communication signals from the power supply. All splitting transformers are Arnold 0.4-in moly permalloy powder core, which have better permeability stability over the temperature range than usual ferrites. Even though the overall permeability

is lower for ferrites, it is better for establishing a constant working point for low power consumption over the entire temperature range. After electromagnetic interference (EMI) filtering, the Rx communication signals are fed to level detectors for conversion into logic-level signals.

### Power and Grounding

Power is directly supplied from the lander power bus. For a 50-m tether, it is more efficient to pass a 28-V signal directly over the tether than an up-converted one. The full galvanic isolation of earlier designs has been abandoned because the overall power consumption of the rover is sufficiently low. However, isolation can only be discarded if the lander bus is appropriately protected against excessive current by active elements.

Therefore, a solid-state current monitor protects both lines of the lander bus (Figure 8). After common mode and differential mode filtering, the power signal is passed to a pair of signal toroids to create two separate channels per wire, which is required for differential communications.

On the rover side (Figure 9), after separating power from communication signals and filtering with a differential mode filter, the signal is passed to the power converter unit to produce regulated  $\pm 5$  V for further use. Furthermore, an LM3488 controller switches the 28-V line for motor supply. The LM3488 provides pulsewidth modulation (PWM) output and current limit and is rated down to  $-40$  °C. Tests have shown that it is operational below  $-150$  °C, with a variation of operating frequency of less than 10% at 1 MHz. The variation decreases at lower frequencies.

The presented solution has a power conversion efficiency between 70 and 75% depending on the overall power level consumed. With full galvanic isolation to the lander bus, efficiency would drop to 51–59% depending on the chosen transmission scheme over the tether.

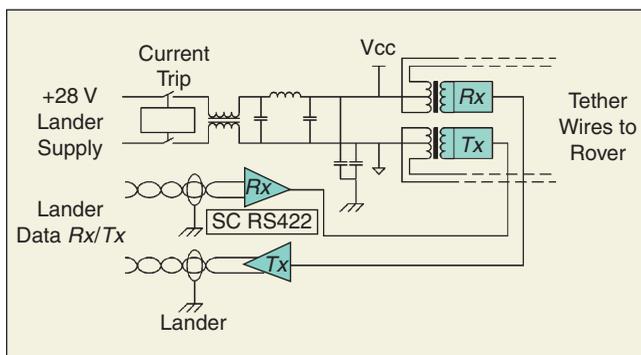


Figure 8. Lander side power and grounding diagram.

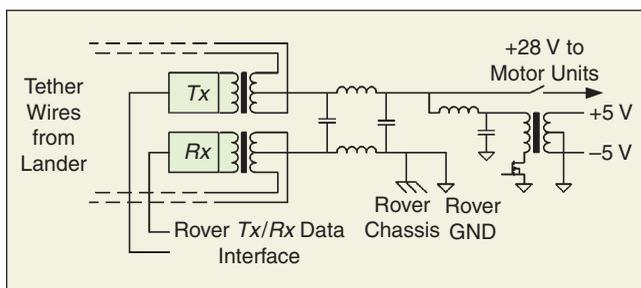


Figure 9. Rover side power and grounding diagram.

### Internal Bus

A new nine-wire serial bus system, using the I<sup>2</sup>C bus protocol for data exchange, has been incorporated into the Nanokhod. This drastically reduces the internal wiring with respect to prior versions. Several nodes, each containing an I<sup>2</sup>C interface, act as functional blocks of the rover.

### Rover Nodes

The rover central controller node and the tether unit node are core systems and always remain powered. The aim of the on-rover controller is to be a central control unit for rover and integrated payloads. However, the integration of payload back-end electronics is still ongoing, which is why the central controller structure could still change at a later stage. In a flight version, an field programmable gate array (FPGA) with implemented IP core, e.g., an 80C51 type processor, would be selected. For the engineering model, an 8-b microcontroller, which has similar capabilities, is chosen, but only functionality that can also be implemented in an FPGA is used. The selected device is a Philips P87C552. Even though the device has its own RAM and EPROM, external RAM/EEPROM/PROM units are used, which are representative for the flight units. Furthermore, the

internal analog to digital converter (ADC) was not used but replaced by an external one. This was done to establish a flight-equivalent design.

Three additional motor controller nodes exist. Their design is again a generic one. The nodes are implemented in the right and left track and in the payload cab. When not in use, these nodes can be powered down. Their state is controlled by the 28-V line, which also supplies the power to the motors. If the 28 V is not present, the node switches off. Power cycling the 28-V line can reset a disabled node. Each of these nodes can interface to up to two motor driver boards that contain the current controllers for the motors. Furthermore, the generic motor controller nodes are responsible for collecting housekeeping data of the sensors distributed around the rover. Their design is illustrated in Figure 10. Because all these nodes have independent oscillators, time synchronization of their oscillator frequencies is performed. The synchronization is achieved by the general broadcast message facility of the I<sup>2</sup>C.

In a flight design, common functionality of all nodes would be implemented in a mixed signal or digital application-specific integrated circuit (ASIC). This would result in a radiation-hard design, with good immunity to single event upsets. Considered processes are the MG2RT (0.5 μm) from Amtel or the Aeroflex CRH (0.25 μm) processes. For this project, the ASIC functionality is mimicked by a Motorola 8-b microprocessor. Again, only representative functionality of the microprocessor is used. The ADC is an external one and is simulated by an AD7888. This 12-b ADC is functional at -180 °C. It shows a deviation of about 5 mV at -80 °C and about 3 mV at -150 °C, which corresponds to a maximum of 8 and 5 LSB at low temperatures. For radiation compatibility, an Intersil HS0548RH together with an AD574 could be used to replace the engineering model design.

### Motor Drivers

The motor driver boards are located next to the motors to minimize EMI to the node boards. The motors always operate at a constant speed in a current control mode. The control circuitry is implemented using an L6207 motor controller. The IC provides current control and motor switching in a single CMOS package. With a chopping frequency of 43 kHz, a step speed of 1.6 kHz (~5,000 rev/min) is achieved in the motor.

### Sensors

Sensors of the rover are acquired via the eight-channel ADCs on each rover node and provided to the central controller.

To measure the angles of the payload cab, endless rotational

potentiometers PIHER N-15 are used. They are preferred over optical encoders because of their lower-power consumption and better suitability in dusty environments. The angle is acquired via differential measurements to compensate for temperature drift in carbon film and voltage supply.

The rover can measure the gravity vector with three accelerometers that are placed in an orthogonal set inside the locomotion units. The ADXL203 has shown repeatable output characteristics during cold temperature tests, however, because of its nonlinearity, it requires local heating or recalibration with calibration tables.

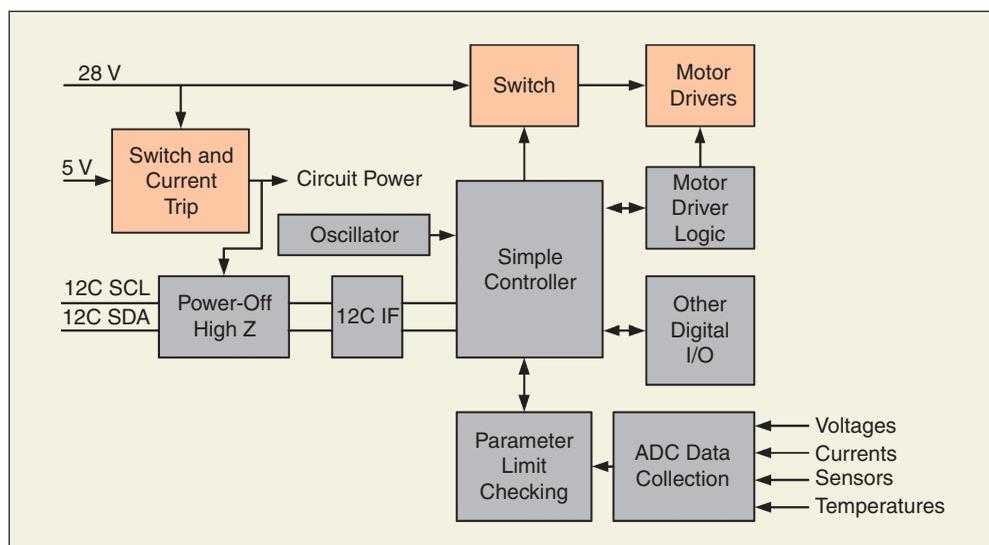
Contact sensors for detecting whether the PLC is docked to a target are mechanical switches (ITT KSR Series), which have proven to function properly in cryogenic temperatures. Their membrane is exchanged with a thin PTFE membrane. They are more lightweight and compact than any other possible contact sensors.

In addition, voltages, currents, and temperatures of the rover electronics are monitored and converted by the eight-channel ADCs on each node.

### Rover Thermal Design

The general thermal concept of the rover is a passive one. The goal of the design is to retain as much heat as possible within the rover while avoiding local hot spots. These can easily occur for active components in vacuum. Therefore, the following approach is followed in the design:

- ◆ heat conduction to the regolith via the tracks is minimized by design
- ◆ heat radiation to the outside is minimized using low-emissive finish, such as Alodine 1200 ( $\epsilon = 0.12$ )
- ◆ all PCBs are mounted to the rover structure via aluminum standoffs to allow conduction and avoid local heating
- ◆ thermistors measure the temperature of the main motor driver chips and other hot power supply components
- ◆ some sensors contain thermistors to allow for calibration against temperature.



**Figure 10.** Node design of motor drive node (gray boxes show functionality to be implemented into an ASIC).

## Thermal Model

A simple thermal model of the rover was created to predict general chassis temperatures and transient times. The model consisted of five nodes, for which temperatures were calculated per step, and a ground node, which describes the planet surface.

- ◆ The PLC node contains the motor driver electronics, rover controller, motor, and payloads as heat sources. Radiation to open space and to the planetary surface and conduction via the levers to the locomotion unit nodes is considered in the heat path.
- ◆ For each locomotion unit, two nodes, an inside one and an outside one, are implemented. All motors and electronics are considered as heat sources. Conduction and radiation paths are implemented as well.
- ◆ The tether unit is modeled as the fifth node, not containing heat sources but only providing radiation and conduction paths to the environment.

The thermal analyses performed have not shown critical hot-spot areas in the rover. However, further analysis during thermal-vacuum testing will be performed on the engineering model to consolidate these results and to verify the thermal model.

## Rover Payload

For the first time, the Nanokhod integrates the front ends of three scientific instruments and a laser stripe sensor into the PLC. This payload suite is called GIPF.

Two of the instruments have flight heritage whereas one, the camera, is a new development. However, all three front ends are built in engineering model quality and will undergo extensive environmental and functional testing in the future. The instrument back ends are currently being integrated and miniaturized into a common subsystem. In the current stage, the rover controller reserves allocations for the common subsystem.

## MIMOS2

The purpose of the MIMOS2 instrument is to identify iron-bearing minerals. The MB spectrometer is developed by the Institute of Inorganic and Analytic Chemistry at the University of Mainz. The instrument has flight heritage, was part of the Beagle2 lander [8], and functioned as a scientific payload on the Spirit and Opportunity rovers of the MER mission [9]. The

instrument has undergone some changes in mechanical design to optimally integrate into the Nanokhod PLC (Figure 11).

Furthermore, to improve radiation shielding with respect to prior designs, the outer MB drive tube was manufactured entirely in tantalum.

For better handling, each detector of the new sensor head can be changed without dismantling them all. The overall dimensions of the sensor head are rectangular, with a cross section of  $41 \times 41$  mm and a length of 81 mm. The overall mass of the sensor head for GIPF is about 300 g. First measurements of the new front end, without optimized setting of the discriminator thresholds, show the principal readiness of this sensor head (Figure 12).

## APXS

The purpose of the APXS is the determination of the elemental chemical composition of geological samples.

This instrument has a long heritage as a flight instrument for space missions. The APXS was, among others, a part of Pathfinder, Rosetta, and the MER missions. On MER, the spectrometer is attached as scientific payload to the Spirit and Opportunity rovers.

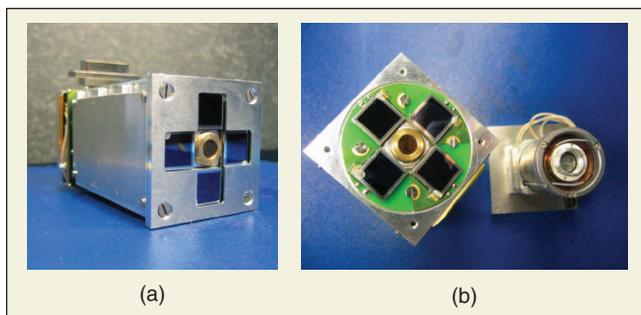
With respect to the MER design of the instrument [10], some changes have been performed to the sensor head to allow the integration into the Nanokhod rover. First of all, removing the cylindrical external casing, the protective door mechanism, and the alpha detector system (including the PCBs with the alpha detector preamplifiers) reduced the size of the instrument. Because of a greatly improved X-ray detector system, the alpha channels provide only little extra value and will, in fact, be omitted in all future instrument versions. The protective door mechanism has proven not to be required on Mars. Changes of the sensor head are illustrated in Figures 13 and 14.

Added benefit of the mechanical rearrangement was the increase of sensitivity of about two and one half times (Figure 15), which resulted from a reduced working distance of 15 mm instead of the earlier 26 mm. The resulting instrument front end has a rectangular external envelope. The cross section is  $39 \times 39$  mm whereas the length of the sensor head is 47 mm. The mass of the sensor head is about 115 g.

Regarding the instrument back-end electronics, the traditional analogue electronics and microcontroller have been replaced with a commercial digital signal processor, the  $\mu$ DXP, provided by X-ray Instrumentation Associates, Newark, California. The porting into the digital domain is a first important step to enable miniaturization of the complete instrument electronics into the Nanokhod rover.  $\mu$ DXP is chosen because all filter algorithms are implemented in hard-wired logic on a Xilinx FPGA. This will allow to transfer the key technology into a space-qualified radiation-hard FPGA in a next step.

## Interference between MIMOS and APXS

Measurements have shown that because of the close geometry of the PLC, the  $^{57}\text{Co}$  sources, in particular the calibration source, of the MB spectrometer significantly increases the background of the X-ray APXS spectra. Measured background from an 80 mCi  $^{57}\text{Co}$  source is shown with different



**Figure 11.** Front end of the MB spectrometer as adapted for the PLC of the Nanokhod: (a) Assembled view. (b) MB drive disassembled from detectors. (Developed by the University of Mainz.)

shielding setups in Figure 16. The placement of the source and the strength of the source was representative for the instruments and their accommodation in the PLC.

To achieve meaningful measurements with the APXS near the MIMOS, a 10-mm Densimet (porous W filled with alloy of Ni, Fe, or Cu, density  $\sim 17 \text{ g/cm}^3$ ) shield has to be additionally implemented between the instruments. The shield had to be placed near the source, close to the MB drive. The influence of APXS radioactive sources to MB spectra is proven to be negligible.

### MIROCAM

The goal of the MIROCAM is to provide close-up images of the same field of view as covered by the MIMOS2 and APXS. The Institute of Planetary Research, German Aerospace Agency, Berlin, has newly developed the MIROCAM under ESA contract.

The complete camera system is integrated into a very small package. The cross section is  $20 \times 40 \text{ mm}$ , with a length of 100 mm. A picture of an early prototype is shown in Figure 17.

Considering the uncertainty in the distance estimate from the lens to the sample surface, as well as the surface roughness of the samples, an autofocus system was developed for the MIROCAM. Moreover, in addition to close-up imaging, the camera can change the focus to infinity to support navigation of the rover. The opto-mechanical system is based on an ARSAPE AM1020 stepper motor, in combination with an MHD10-160-H HD gear, developed by Micromotion GmbH. An excenter converts the rotary output onto a linear bearing that carries the lens system.

Four pairs of red, green, blue, and infrared LEDs are arranged around the camera aperture for illumination of

the target surface. The LEDs are switched by Vishay Si3865 power switches.

The image sensor of the first prototype is the LM9637,  $648 \text{ (h)} \times 488 \text{ (v)}$ , monochrome active pixel sensor (APS). It is placed behind a modified lens from Thales Optic and has a focal length of 8.3 mm.

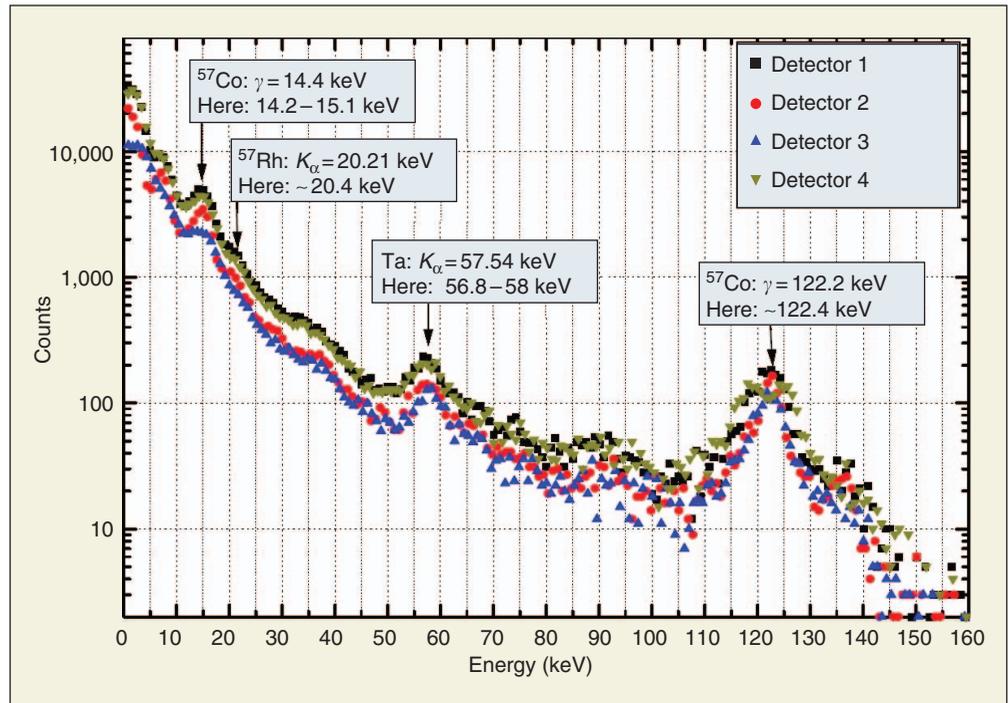


Figure 12. First energy spectrum measured with new MIMOS2 sensor head. For comparison, the literature values of some materials of the source ( $^{57}\text{Co}$  in rhodium matrix) are shown.

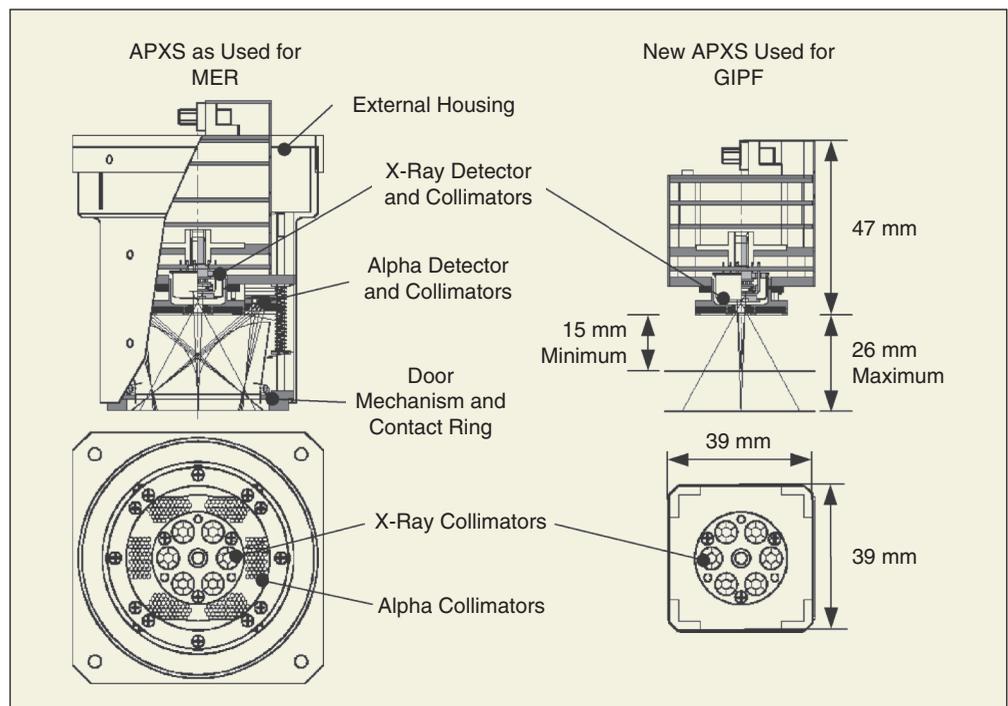
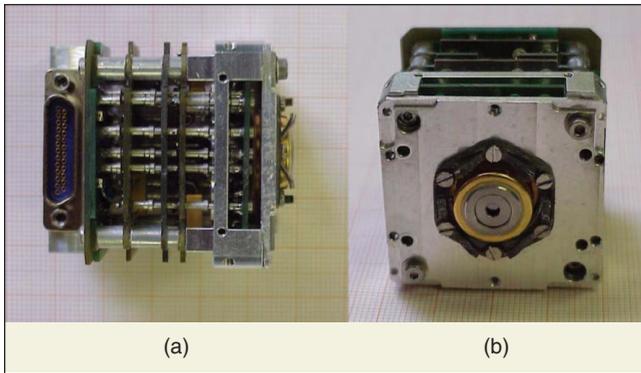


Figure 13. APXS sensor head comparison between MER and GIPF design.

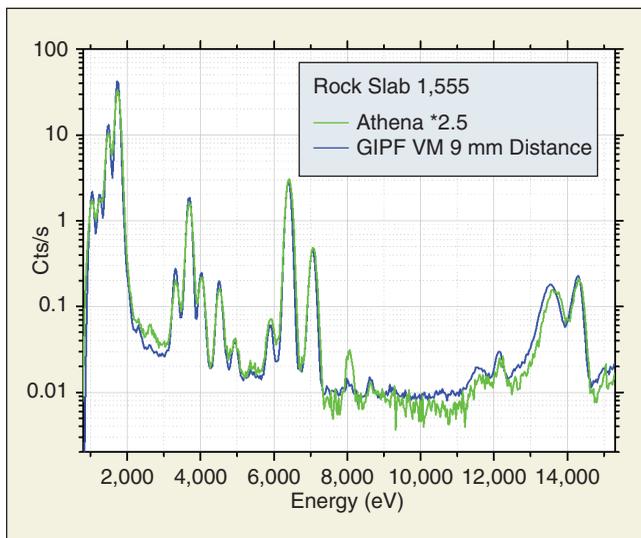
The electronics of the camera are based on a Xilinx Spartan 2 FPGA (XC2S100) that includes all control electronics for the APS, the LED illumination, and the focus control. An additional driver component for the stepper motor is the L293DNE integrated circuit. The camera can be fully controlled by an I<sup>2</sup>C serial interface from the common subsystem. Demonstration pictures are shown for infinity and close-up modes in Figure 18.

## Rover Navigation

The navigation concept of the Nanokhod is similar to previous versions (lander based), but with some extensions. The lander stereo camera system that is mounted onto a pan-and-tilt unit extracts the light emitted from the LEDs attached to the rover PLC (Figure 19) to determine their position and orientation in the terrain [7], [11]. Custom-developed lightweight stereo camera systems have been built by the ESA, together with industry and research institutes, targeting a mass of below 1.5 kg.



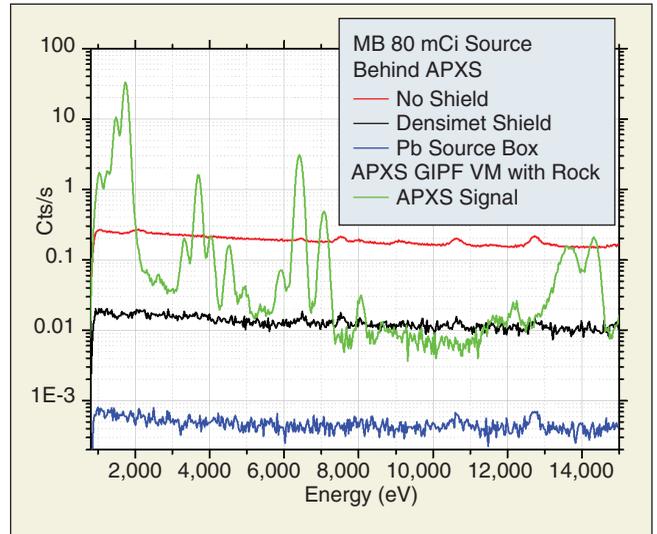
**Figure 14.** The new APXS front end developed for the Nanokhod PLC (without sources) by MPICH in Mainz: (a) Side view. (b) Front view.



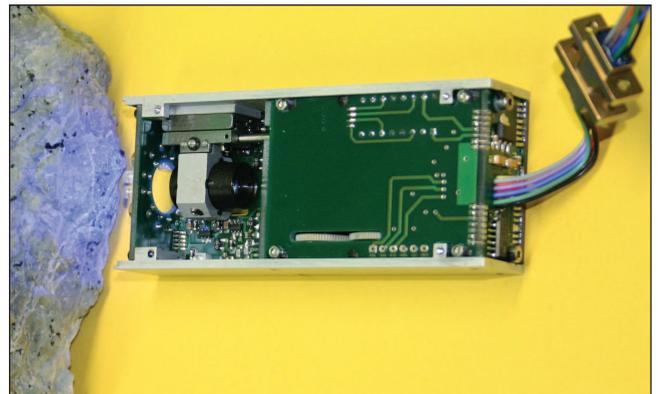
**Figure 15.** Comparison of response for different working distances (sample 1555, andesitic basalt): APXS for MER –26 mm × 2.5 and APXS for GIPF –17 mm.

In the new Nanokhod, tilt sensors have also been integrated, which allow to continuously check for safety hazards, such as tipping over because of very steep slopes. The baseline navigation concept relies on direct visibility of the rover from the lander stereo camera. In the case of nonnominal loss of rover visibility, the Nanokhod can use its hazard avoidance systems for local stand-alone navigation.

In that case, the tilt sensors, the laser stripe sensor, and the MIROCAM camera are used together with position tracking



**Figure 16.** Background in APXS X-ray spectra due to MIMOS2 source with different shielding: No shield (red), 10-mm Densimet shield (black), and enclosure of MIMOS2 source in 10-mm led container (blue).



**Figure 17.** Assembly view of MIROCAM camera developed by DLR Berlin.



**Figure 18.** Infinity and close-up pictures taken by MIROCAM.

based on odometry to find a suitable path in the unknown terrain. The MIROCAM can collect overview images of the rover surroundings by making use of its infinite focus mode. The whole data set can be used on ground for rover position identification.

### Laser Stripe System

The integrated laser stripe system allows implementation of simple terrain reconstruction and navigation functions directly inside the Nanokhod central processor. A simple laser line extraction algorithm is used on the MIROCAM images for orientation and distance detection of obstacles such as rocks or trenches. This information can be used for approaching and docking of the PLC to scientifically interesting targets, which is important if the rover itself occludes pictures from the lander stereovision system.

A laser plane is projected horizontally out of the PLC with a fan angle of 14°. A laser diode (Stocker Yale Mini Laser) with a non-Gaussian line projection optic produces a uniform intensity line on the terrain. The accommodation inside the PLC is optimized to increase the maximum and minimum distance at which obstacles can be recognized. With the given position inside the PLC, objects can be detected in distances of 105–630 mm in front of the rover.

The laser diode functions properly at cryogenic temperature, and analysis has shown that a laser source power of 1 mW is sufficient for applying the navigation computations.

The structured light generation is used in the following way for navigation. The laser projects a known pattern of light into the scene. Terrain slope or object or obstacle detection is performed on a multiple line picture acquired in several steps by the MIROCAM and assembled by the on-rover processor. Each line results from an image that is acquired with a different but known PLC angle with respect to the terrain. Feature extraction algorithms can be applied onto the picture, knowing the rotational information provided by the PLC angular sensors. Figure 20 illustrates the concept and how three different line reflections assembled in one camera picture indicate the presence of a potential obstacle.

The rover is able to send this information back to the lander computer to request an appropriate reaction, or it can execute some locally implemented routines. This way, constant monitoring by lander stereo imagery can be reduced in nominal situations, which saves energy for the mission.

### Rover Energy Budget

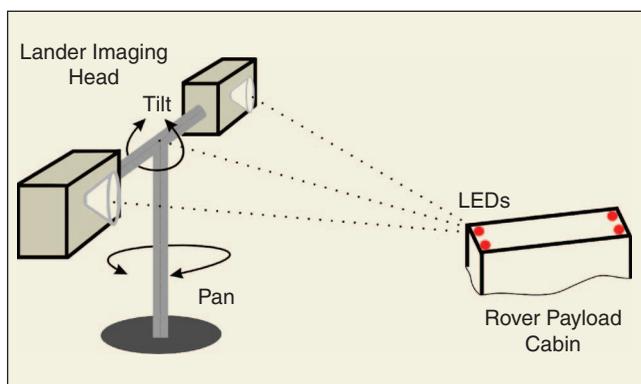
Energy required to perform essential tasks is summarized in Figure 21. The values shown result from a performed case study.

The rover breaks the shown segments down into subsegments. Each subsegment consists of the following accumulated actions, for which the power consumption is shown:

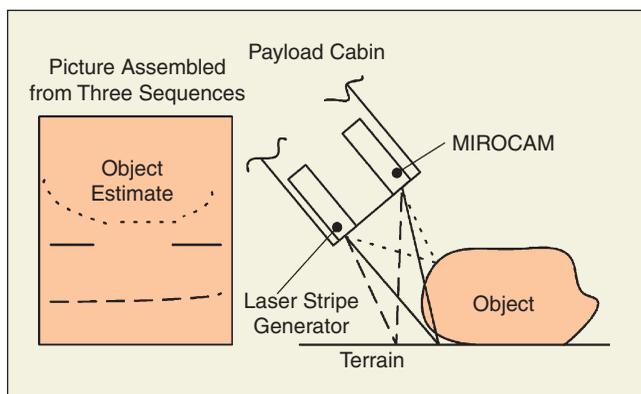
- ◆ rover rotation (45° in graph)
- ◆ slope and obstacle identification with laser and camera
- ◆ rover movement in straight line (40 cm in graph).

## The rover payload cabin optimally integrates the geochemistry instrument package.

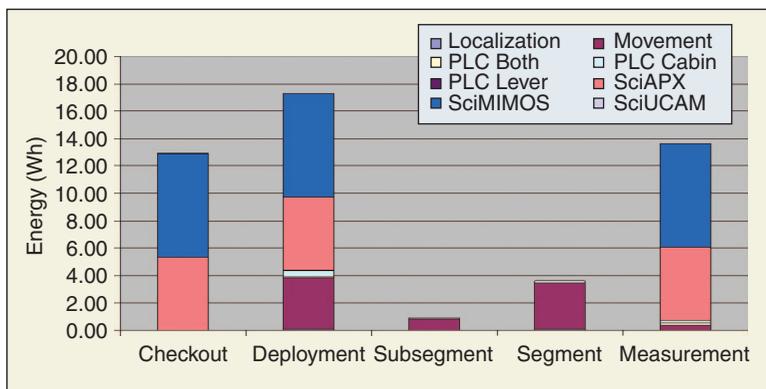
One segment comprises three subsegments and the power consumed for the lander-based localization procedure between each subsegment motion. This includes power consumed for the lander cameras, the rover LEDs, and all required calculations.



**Figure 19.** Schematic of Nanokhod navigation concept. Imaging head attached to lander base will recognize active markers attached to rover PLC.



**Figure 20.** Use of structured light to extract environment information from a sequence of three images.



**Figure 21.** Energy required by the rover during several mission steps.

The measurement sequence shown includes docking and undocking of the PLC and one measurement cycle of each of the instruments. The checkout and deployment sequence shown is used for instrument calibration at mission start-up.

## Conclusions

With respect to prior Nanokhod versions, this article introduces a novel engineering model design. The presented design has been optimized to withstand cryogenic temperatures approaching those of open space, vacuum, and dusty regolith soil. Although the Nanokhod engineering model is still under construction and integration at the time of writing, extensive testing has already been performed on key subsystems under cryogenic conditions. This strengthens the confidence in a successful implementation of the design.

In particular, a new sealing design has been developed and successfully tested. Furthermore, a new and more efficient power transmission scheme has been designed. It is optimized for the given tether length and the overall low power consumption of the rover. A serial bus within the rover serves several nodes that fulfill all functions related to the rover control and operations. The electronic architecture is chosen such that radiation-hard space-qualified components can be used in the next instance without a major redesign. The drivetrain of the rover has been redesigned to be able to overcome obstacles of 0.1 m height and provide terrain gradeability of at least 30°. Furthermore, stiffness of the PLC has been improved by replacing the worm gears of earlier designs with HDs. This has also led to much higher positioning accuracy. In general, all requirements posed by a mission to the night-side surface of an atmosphere-less planet like Mercury are fully met by the design. This will make the rover suitable for application on Mars or the Moon.

Furthermore, for the first time, front ends of three scientific instruments, the APXS, MIMOS2, and MIROCAM, are fully integrated into the PLC at engineering model quality. The common subsystem to fully integrate the instrument back ends into the rover is currently under development. Experiments have successfully demonstrated the feasibility of using the APXS and MIMOS2 together in tight integration. This might open the way to a dedicated geochemical instrument package facility that can be used even independently from the rover, eventually delivered as a plug-and-play instrument housed inside the rover PLC, with purely digital interfaces and power supply lines. It is envisaged to perform extensive testing in cryogenic temperatures with the engineering model. Tests will be performed in vacuum and dust environment at the same time.

## Keywords

Exploration rover, harsh environment, tracks, payload, rugged, cryogenic temperature, vacuum compatibility, navigation, mobility.

## References

[1] R. Rieder, H. Waenke, V. Gromov, A. Kemurdjian, V. Kucherenkov, P. Sologub, B. Andreichikov, A. Kalyushniy, B. Korchuganov, V. Linkin, E. Kankeleit, G. Klingelhöfer, J. Foh, P. Held, R. Teucher, O. Prilutzki, E. Evlanov, V. Khromov, B. Zubkov, T. Economu, G. Neukum, H. Hoffman,

H. Michaelis, E. Ress, M. Langfeld, and P. Jakes, "Nanokhod: A miniature instrument deployment device with instrumentation for chemical, mineralogical and geological analysis of planetary surfaces, for use in connection with fixed planetary surface stations," in *Proc. Abstracts of the Lunar and Planetary Science Conf.*, 1995, vol. 26, pp. 1161–1162.

[2] R. Bertrand, R. Rieder, and M. V. Winnendaal, "European tracked micro-rover for planetary surface exploration," in *Proc. ASTRA Workshop ESA*, Noordwijk, The Netherlands, 1998, pp. 1–8.

[3] M. P. Golombek, R. C. Anderson, J. R. Barnes, J. F. Bell, N. T. Bridges, D. T. Britt, J. Brueckner, R. A. Cook, D. Crisp, and J. A. Crisp, "Overview of the Mars Pathfinder Mission: Launch through landing, surface operations, data sets, and science results," *J. Geophys. Res.*, vol. 104, no. 4, pp. 8523–8553.

[4] S. W. Squyres, R. E. Arvidson, E. T. Baumgartner, J. F. Bell, III, P. R. Christensen, S. Gorevan, K. E. Herkenhoff, G. Klingelhöfer, M. B. Madsen, R. V. Morris, R. Rieder, and R. A. Romero, "Athena Mars rover science investigation," *J. Geophys. Res.*, vol. 108, no. E12, p. 8062, 2003, doi: 10.1029/2003JE002150.

[5] B. H. Wilcox and R. M. Jones, "The MUSES-CN nanorover mission and related technology," in *Proc. Aerospace Conf.*, 2000, vol. 7, pp. 287–295.

[6] R. Bertrand, J. Dacolmo, and S. Klinkner, "RTPE: Robotic technology for planetary exploration," European Space Agency, Noordwijk, Netherlands, Final Rep. RTPE-54-10, 2003.

[7] M. Vergauwen, M. Pollefeys, and L. van den Gol, "A stereo vision system for support of planetary surface exploration," in *Proc. Int. Conf. Computer Vision Systems (ICVS)*, Canada, 2001, pp. 298–312.

[8] C. T. Pillinger, *The Guide to Beagle 2*. London: Faber and Faber, 2003.

[9] G. Klingelhöfer, R. V. Morris, B. Bernhardt, D. Rodionov, P. A. de Souza, Jr., S. W. Squires, J. Foh, E. Kankeleit, U. Bonnes, R. Gellert, C. Schröder, S. Linkin, E. Evlanov, B. Zubkov, and O. Prilutzki, "Athena MIMOS II Mössbauer spectrometer investigation," *J. Geophys. Res.*, vol. 108, no. E12, pp. 8–1–8–18, 2003.

[10] R. Rieder, R. Gellert, J. Brückner, G. Klingelhöfer, G. Dreibus, and A. Yen, "The new Athena alpha particle X-ray spectrometer for the Mars exploration rovers," *J. Geophys. Res.*, vol. 108, no. E12, pp. 7–1–7–13, 2003.

[11] B.-M. Steinmetz, K. Arbter, B. Brunner, and K. Landzettel, "Autonomous vision based navigation of the Nanokhod rover," in *Proc. iSAI-RAS*, Montreal, Canada, 2001, pp. 1–8.

**André Schiele** received Dipl. Ing. (FH) in microsystems (MEMS) engineering from the University of Applied Sciences in Kaiserslautern, Germany, in 2001. He designed mechatronics for payloads in space as a consultant from 1998 to 2002 and was involved in the mechanical redesign of the APXS for the Rosetta mission and the Nanokhod rover for Max-Planck Institute for Chemistry in Mainz, Germany. He worked on the mechanical design of Nanokhod components for von Hoerner & Sulgar GmbH, Germany. He is a staff robot system engineer with ESA/ESTEC, Noordwijk, the Netherlands, since 2002 and is responsible for the Nanokhod development and qualification. He joined the Man-Machine Systems Laboratory of Delft University of Technology as a Ph.D. candidate in robotics in 2002.

**Jens Romstedt** received the M.Sc. degree in mineralogy from the University of Hamburg, Germany, in 1990. He received the Ph.D. degree in planetology from the Wilhelms University of Münster, Germany, in 1995. After two years of research fellowship at ESA/ESTEC, Netherlands, as a team member of a group to develop an atomic force microscope for the Rosetta space mission, he became an appointed staff member to follow a variety of technology developments and give support to a variety of future space missions.

**Christopher Lee** is a chartered engineer with the Institute of Engineering Technology (formerly the IEE). He obtained his M.Eng. degree in 1991, followed by a Ph.D. degree in 1996, based on work he undertook in the design, development, and deployment of a seaborne laser wave slope meter funded by the U.K. Defence Research Agency. After two years at Roke Manor Research, he moved to Imperial College, London, where he played a pivotal role in the Rosetta Plasma Consortium. In 2003, he moved to von Hoerner & Sulger GmbH, where he leads the Nanokhod Project and other projects including a full-size ExoMars Rover breadboard for Astrium UK.

**Hartmut Henkel** joined von Hoerner & Sulger GmbH in 1994 as head of the space and rocket instrumentation group. He was technical project leader for several scientific instruments in space, among others, including CIDA/STARDUST, COSIMA/ROSETTA (mass spectrometers), and LAD/TWINS (Lyman-Alpha detectors). He is responsible for the miniaturized payload instrumentation for the Nanokhod rover. He was also responsible for the high-resolution XUV CCD camera development for sun research. He received his Ph.D. degree in electrical engineering in 1991 at the University of Darmstadt.

**Sabine Klinkner** received her degree as a diploma engineer for aerospace technology at the University of Stuttgart in 2002 in the field of electrical propulsion and thermal analysis. Since then, she has been working at the SME Company von Hoerner & Sulger GmbH in the fields of robotic systems and mechanical design including involvement in the design of the ROKVISS camera currently flying on the International Space Station (ISS) and a full-size Exomars Breadboard model built for Astrium UK. Since 2004, she has also been studying for her Ph.D. degree at the University of Stuttgart in the field of system design and thermal analysis.

**Reinhold Bertrand** holds an engineering degree in aerospace engineering and received his Ph.D. (Dr.-Ing.) from Stuttgart University on space systems design and simulation. From 1997 to 2004, he worked in industry (von Hoerner & Sulger GmbH, Schwetzingen, Germany) as head of space systems design and robotics and was responsible for a multitude of industrial research and development activities as industrial project manager. Subject areas covered robotic systems and in situ measurement techniques for planetary exploration. He joined the ESA as senior technical and management assistant in the Ground Systems Engineering Department in 2004. He is a senior lecturer at Stuttgart University.

**Rudolf Rieder** received his Ph.D. degree in physics from the University of Vienna, Austria, in 1966. From 1986 until his retirement in 2005, he worked at the Max-Planck Institute for Chemistry in Mainz, Germany, developing several generations of instruments for the in situ measurement of the elemental chemical composition of planetary surfaces, three of which were deployed on Mars on board of the NASA rovers Sojourner, Spirit, and Opportunity (another instrument is on

board of the Rosetta lander). He was also engaged in the development of deployment devices for in situ instruments that ultimately led to the design currently known as Nanokhod. He took up his earlier activity as consultant (Zivilingenieur fuer Technische Physik) after his retirement from work at the Max-Planck Institute for Chemistry.

**Ralf Gellert** is with the department of physics at the University of Guelph, Ontario, Canada. From 1999 to 2004, he was lead engineer and calibration lead for the APXS on the MERs and Rosetta missions with Max-Planck Institute for Cosmochemistry and University of Mainz, Germany. Since 2005, he is a principal investigator for the APXS on the Mars Science Laboratory mission and lead scientist for the APXS instruments on MER.

**Göstar Klingelhöfer** is with the Institute of Inorganic and Analytical Chemistry at Johannes Gutenberg-University in Mainz, Germany. He is head of the research group Planetary Exploration. He has more than 20 years of experience in design, system engineering, and project management of scientific instruments and space instrumentation (e.g., Mars-96 MIMOS I, APEX-2001 MIMOS II, MER-2003, ExoMars, Phobos-Grund). He is the principal investigator for the APXS on Rosetta, the Moessbauer MIMOS-II on MER-Spirit and Opportunity, the ESA Beagle-2 lander, and ESA technology studies.

**Bodo Bernhardt** received his degree as a diploma engineer of physics at the Technical University of Darmstadt, Germany, in 1997. As research assistant at the Institute for Nuclear Physics, Darmstadt, he was working on the detector system and miniaturization of the MB spectrometer MIMOS II. From 1999 to 2004, he was research assistant at Johannes Gutenberg-University Mainz and lead instrument engineer for the MB spectrometer on the NASA MER 2003 mission and the Mars-Express-2003 Beagle 2 lander mission. He was assisting MER operations during cruise phase and in the first surface operations phases of the mission. Since then, he has been working at the SME Company von Hoerner & Sulger GmbH in the fields of analog and digital electronics design for space instrumentation.

**Harald Michaelis** is with the Institute of Planetary Research of the German Aerospace Agency (DLR e.V.) in Berlin, Germany. He is head of the Planetary Sensor Systems section, with more than 20 years experience in design, system engineering, and project management of scientific space instruments (e.g., PHOBOS-FGMM, CASSINI-DISR sensor electronics, Imager for Mars-Pathfinder, Rosetta-VIRTIS/ROLIS, MEX-HRSC, Venus Monitoring camera, Bepi-Colombo laser altimeter, many ESA technology studies).

**Address for Correspondence:** André Schiele, European Space Agency, Automation and Robotics Section, Keplerlaan 1, 2201 AZ Noordwijk ZH, The Netherlands. Phone: +31 71565 3760. Fax: +31 71565 5419. E-mail: Andre.Schiele@esa.int.

# Embedded System Design for Robots

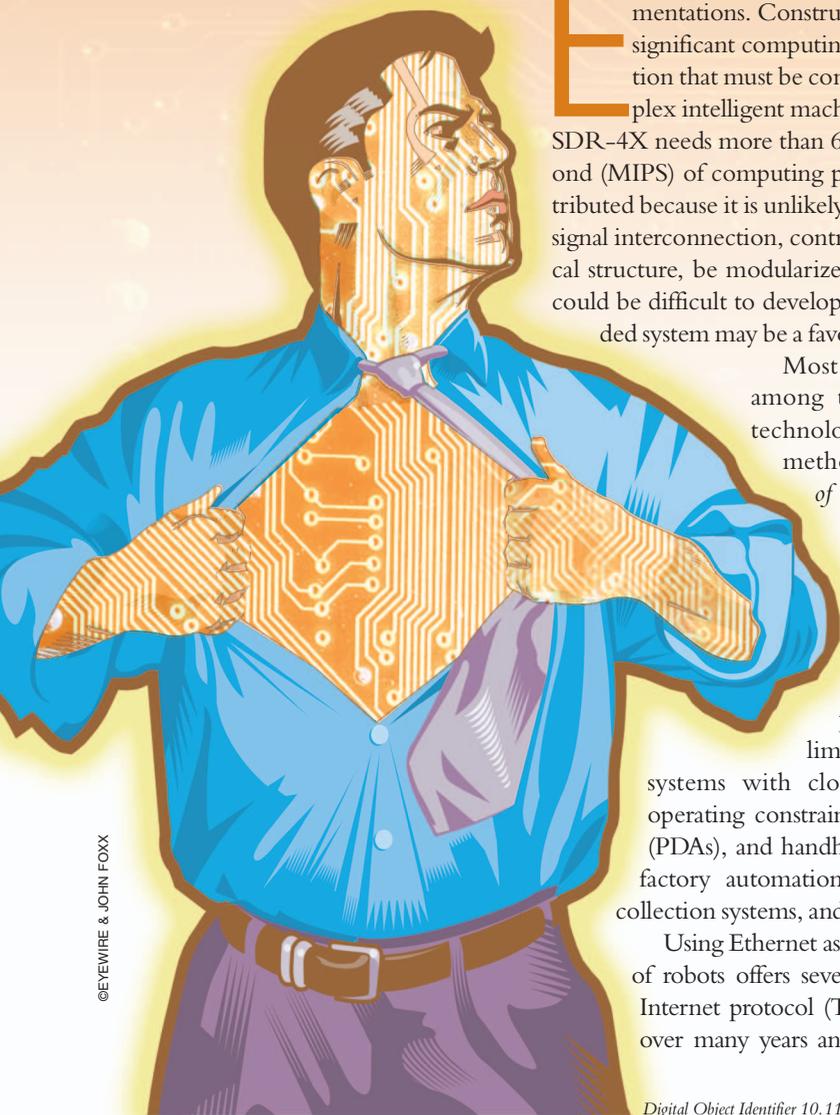
*Design Concept,  
System Architecture,  
and Implementation*

BY LI-WEI WU AND  
JWU-SHENG HU

Embedded system design is essential for successful intelligent robotic implementations. Constructing a robot that can perform complicated tasks requires significant computing power and system integration effort [1]–[3]. The question that must be considered is, “What embedded system is needed for a complex intelligent machine such as a humanoid robot?” for example, the SONY SDR-4X needs more than 60 processors and over 2,260 million instructions per second (MIPS) of computing power [3]. The computing architecture is inherently distributed because it is unlikely to dump all the raw information into a single CPU. The signal interconnection, control, and information processing should, like the mechanical structure, be modularized. Without careful design, the entire embedded system could be difficult to develop, maintain, and extend. Therefore, a distributed embedded system may be a favorable choice.

Most distributed systems have a certain network topology among their processing units. Based on currently available technology, what would be the appropriate networking method for an intelligent machine? In May 1998, the *Journal of Internet Computing* ran a special issue on embedded Ethernet technologies and highlighted recent developments and industrial applications of embedded Internet technology [4]. This technology supports devices and operating environments outside the traditional desktop PC envelope, where onboard memory, CPU power and speed, display capability, persistent storage, and costs are usually severely limited. At the soft end of the spectrum are embedded systems with close-to-desktop-PC resources and no real-time (RT) operating constraints, including cellular phones, personal digital assistants (PDAs), and handheld terminals. At the hard or deeply embedded end are factory automation and machine controllers, instrumentation and data collection systems, and telecommunication equipment [5], [6].

Using Ethernet as the communications backbone for the embedded systems of robots offers several advantages. First, the transmission control protocol/Internet protocol (TCP/IP) de facto standard has been proven to be robust over many years and is open to the technical community. Second, when a



©EYEWIRE & JOHN FOX

Digital Object Identifier 10.1109/MRA.2008.917885

robot is interconnected with the access network (the public network) and the home network, the designer must heavily emphasize the addressing system and related security issues. Historically, the IP community has carefully considered these problems, and therefore, IP technology is competent for addressing these issues. The abundance of applications and tools make Ethernet highly effective for unifying the system interface and reducing system complexity. For example, it can be used as the communication backbone in parallel processing [7]–[9]. The latest technology has pushed Ethernet bandwidth to the Gb/s/Tb/s range [10], leading to proposals to incorporate it into the system-on-a-chip (SoC) design in both the academic and industrial sectors [11]–[13]. In terms of bandwidth and cost, Ethernet offers a very competitive solution for embedded applications.

However, Ethernet devices require synchronization for purposes such as motor control and vision. Conventional protocols, e.g., user datagram protocol (UDP) and TCP/IP, are inadequate for synchronization because the 1-persistent carrier sense multiple access with collision detection (CSMA/CD) protocol has unpredictable delay characteristics. CSMA/CD employs an exponential back-off mechanism to circumvent collisions, making the network nondeterministic and thus non-RT. When both RT and non-RT packets are transported over an ordinary Ethernet, RT packets may experience a large delay because of 1) the contention with non-RT packets in the originating node and 2) the collision with RT and non-RT packets from other nodes.

Numerous studies have been undertaken to support RT bandwidth guarantees over Ethernet hardware such as 100VG AnyLAN [14], RETHER [15], EtheReal [17], [18], RTnet [19], RtP [20], and TTP/C [21]. However, most of them have a concentrated management structure, requiring an effective and programmable switch router or hub to run the RT schedule algorithm. Such a router is usually very large and superfluous for small robot systems.

In 2003, the International Electrotechnical Commission in Geneva developed the standardization process for RT Ethernet (RTE) to use Ethernet as a fieldbus alternative, encouraging the further establishment of the standards while maintaining the highest possible levels of flexibility [22]. Currently, industrial Ethernet is also vigorously becoming established in automation technology, and several promising approaches for RTE are forcing their way into industry, such as PROFINET, Ethernet for control automation technology (EtherCAT), and Powerlink. The various RTE approaches differ significantly and are not compatible with one another. Therefore, they are associated with some defects in embedded systems of small mobile robots. Each issue is explained in the following sections.

## PROFINET

PROFINET is the industrial Ethernet solution supported by PROFIBUS International User Group. PROFINET's background is distributed automation: objects can be easily described, reused, and connected to one another. It supports two protocols: standard distributed component object model (DCOM) over TCP/IP for non-RT communication and RT class I protocol for medium-performance real time. The well-known DCOM [23] can be seamlessly integrated with process

automation based on object linking and embedding for process control. Notably, however, DCOM is a high-overhead protocol and thus unsuitable for embedded, low-cost systems or small robot systems [24]. For further information of PROFINET, please see [22], [24]–[28].

## EtherCAT

EtherCAT is an open RT solution that was developed by Beckhoff and is supported by the EtherCAT Technology Group. It is tailored for centralized automation only. It does not support distributed control systems. EtherCAT can be regarded as a new fieldbus with Ethernet cables. Notably, EtherCAT is limited to the use of Beckhoff proprietary application-specific integrated circuits (ASICs) and so cannot be widely applied in any other embedded Ethernet development solution (standard Ethernet chips) [29]. This issue will limit the development of robot systems. Please see [22], [30], and [31] for more detail information.

## Ethernet Powerlink

Ethernet Powerlink [32]–[35] was defined by Bernecker & Rainer, and it is now supported by the Ethernet Powerlink Standardization Group ([www.ethernetpowerlink.org](http://www.ethernetpowerlink.org)). Ethernet Powerlink is mainly based on a principle called slot communication network management (SCNM). It uses individual slots for isochronous data and shared slots for asynchronous data. The SCNM method guarantees collision-free data transfer between master and slave. All nodes within a segment are synchronized by the reception of the start-of-cycle Ethernet packet, which the master sends at the beginning of each cycle. The master polls each slave using a poll-request frame. A slave is only allowed to send an Ethernet packet after it has received such a frame. However, SCNM method does not make the best of current network switch performance. Since a current switch can functionally be considered to be a multiport bridge (see “Network Switch”), in practice, a switch is much more powerful than a traditional bridge primarily because of its ASIC-based hardware architecture and its ultrarapid simultaneous multiple access memory. Additionally, a switch can have an IP address and as many media access control (MAC) addresses as ports, facilitating its configuration. Restated, an Ethernet switch can provide one transmitting collision domain per port (dedicated bandwidth segment), allowing the collisions to be completely eliminated if the port is in full duplex [36] and only one station is connected to it [37], [38]. In fact, in such a configuration, the CSMA/CD protocol is kept only to ensure compatibility with the classic shared Ethernet since no transmitting collision is possible. However, the SCNM only allows an Ethernet packet to be sent after the reception of the poll-request frame, so Powerlink has not fully utilized the characteristics of the current network switch. By using the current switch technique, the improvement in the RTE protocol can allow different collision domains to send Ethernet packets in each individual slot.

To effectively address the issues raised earlier, this study implements a hardware RT protocol (HRTTP) for an embedded Ethernet robot system. An HRTTP-task time wave structure based on the packet traffic control approach is proposed to enable management to reduce each receive collision domain

and form microsegments that are separated by network switch. Furthermore, the H RTP is a distributed RT protocol with a small footprint and is very suitable for the proposed application (especially for low-end standard embedded Ethernet solutions). Embedded networking was achieved with a distributed embedded system using Ethernet to validate H RTP. The system comprises microcontrollers for motoring and sensing and a host platform that uses SoC. The microcontrollers interface with the network interface controller (NIC) chips where both H RTP and a lean TCP/IP protocol stack are implemented. An RT UDP-to-serial packet converter was designed and implemented to handle H RTP Ethernet packets. The host platform runs embedded Linux and has two network interfaces, a 10/100 Mb/s LAN and a wireless LAN. The network protocol stack of the embedded system was modified to accommodate H RTP. A multiaxis robot platform was also constructed to demonstrate the capability of the embedded system. The platform has 16 degrees of freedom (DoF) (16 motors) and the number of sensors can be expanded. The robot takes the form of a quadruped walking machine. Each foot has four motors and is capable of a complex motion profile. H RTP's transparency enables a developer to manipulate the robot easily using socket programming on a desktop computer or mobile device (such as PDA). The proposed architecture provides a system design tutorial for highly intelligent and extremely complex robot systems.

The rest of this article is organized as follows. The "H RTP" section briefly describes the H RTP model and its simple remote clock synchronization methodology. This section also outlines

H RTP's control packet structure. The "Implementation" section explains the application of distributed architecture in a robot electronic platform, and this platform is applied to a complex quadruped to demonstrate its effectiveness in the conclusion. Network switch, OSI model, and TDMA-E are discussed later.

## H RTP

This section elucidates the basic concepts and fundamental elements of H RTP. First, the control network topology for robots is proposed and related issues considered. The new open system interconnection (OSI) model, incorporating packet traffic concept in the H RTP architecture, is described, and several state machines are defined to explain how each mechanism controls the state in H RTP. Furthermore, the clock synchronization issue is addressed. Eventually, the simple H RTP control packet is presented. All conceptual elements are considered later.

### Proposed Control Network Topology

Figure 1 schematically depicts the proposed model. The robot system consists of three layers: the Internet layer, the gateway layer, and the control and sensing layer. Clearly, this structure mimics the topology of today's network infrastructure, in which the control and sensing layer is the so-called Intranet. However, the difference is that RT messaging must be enforced for RT control (as in motor synchronization). The proposed H RTP is implemented to provide RT messaging. Significantly, by maintaining TCP/IP compatibility, the robot system can easily expand within this layer (such as by adding another processor or sensor). Additionally, robot internal communications between the Internet and the control and sensing layers become transparent (such as a simple bridge function in the gateway layer). Many communications technologies can also be implemented over TCP/IP.

However, the CSMA/CD of Ethernet employs an exponential back-off mechanism to prevent collisions that make the network nondeterministic and thus non-RT. Specifically, the Ethernet packets are gathered into a gate, and collisions occur in the receiver node (Figure 2). If an embedded Ethernet receiver cannot handle these packets, then the receiver node also breaks and affect the stability of the robot system. Furthermore, the proposed H RTP is a distributed RT protocol, which is

## Network Switch

Switches occupy the same place in the network as hubs. Unlike hubs, switches examine each packet and process it accordingly rather than simply repeating the signal to all ports. Switches map the Ethernet addresses of the nodes residing on each network segment and then allow only the necessary traffic to pass through the switch. When a packet is received by the switch, the switch examines the destination and source hardware addresses and compares them to a table of network segments and addresses. If the segments are the same, then the packet is dropped (filtered); if the segments are different, then the packet is forwarded to the proper segment. Additionally, switches prevent bad or misaligned packets from spreading by not forwarding them.

The filtering of packets and the regeneration of forwarded packets enable-switching technology to split a network into separate collision domains. The regeneration of packets allows for greater distances and more nodes to be used in the overall network design and dramatically lowers the overall collision rates. In switched networks, each segment is an independent collision domain. In shared networks, all nodes reside in one, large shared collision domain.

Most switches are easy to install and self-learning. They determine the Ethernet addresses in use on each segment, building a table as packets are passed through the switch. This plug and play element makes switches an attractive alternative to hubs [37], [38].

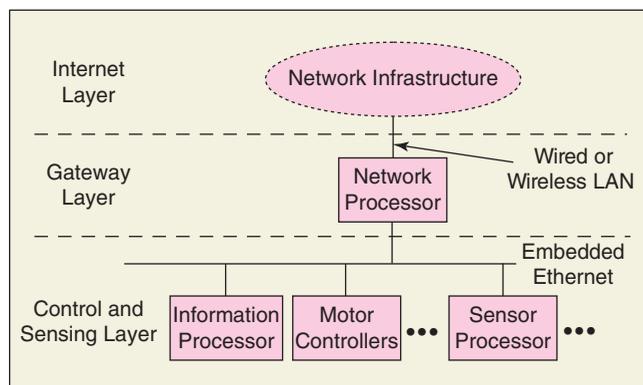


Figure 1. Robot architecture platform model.

targeted for embedded RTE environments. The HRTP model concept is explained later.

### New OSI Model for HRTP

Figure 3 presents a model of the HRTP system in a robot internal Ethernet network. The new OSI model (see “OSI Model”) and the conventional model differ mainly in that the proposed model inserts toggle traffic switch, called the HRTP-transport switch, between the transport and data link layers. This alteration does not affect original TCP/IP because HRTP integrates into the network’s interface driver. The HRTP network driver has the following advantages:

- ◆ high-speed switching
- ◆ adaptation to original TCP/IP protocol stacks
- ◆ efficient utilization of operating system resources
- ◆ increased driver por-tability.

The traffic switch (HRTP-transport switch) intercepts all sending out packets and divides time slices according to the HRTP schedule and the common base clock. Figure 4 describes HRTP’s time mechanism. The following list presents the states of the HRTP state machine.

- ◆ *Initialization*: Prepares the intranet system (local embedded Ethernet nodes in robot system).
- ◆ *Remote clock synchronization*: Synchronizes the clock to ensure accurate time events, as explained later.

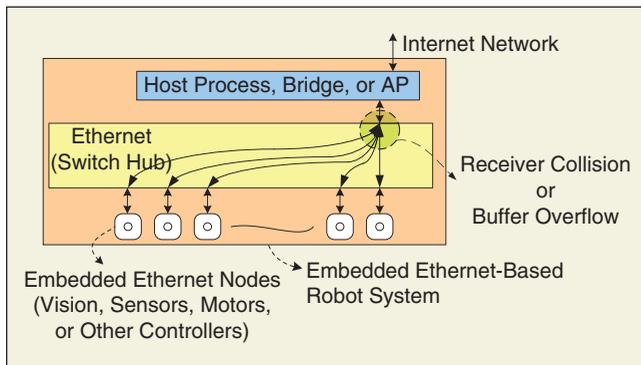


Figure 2. Receiver collision model.

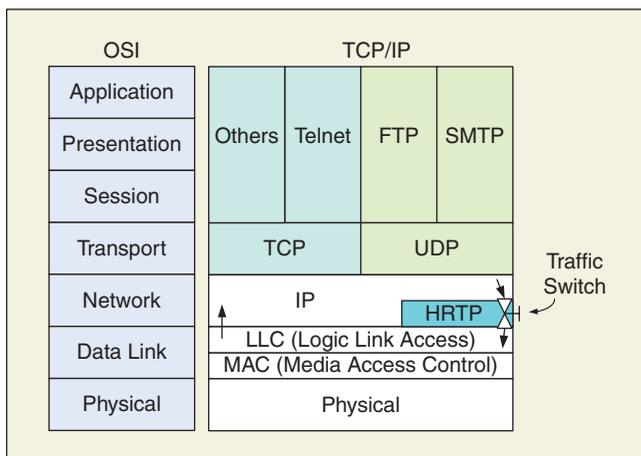


Figure 3. A new OSI model describing the service of HRTP system.

- ◆ *Explore intranet*: Exports the robot’s locally embedded Ethernet network state to determine the client’s IP address and the devices.
- ◆ *Post HRTP schedule*: Sends out HRTP schedule to all devices.
- ◆ *Start clock*: Starts running common base clock.
- ◆ *Traffic control*: Begins HRTP traffic control.
- ◆ *Idle*: System enters sleep mode.

The state machine is controlled by HRTP command packets, as presented in the “Content of HRTP Control Packet” section. Collision occurs in received nodes according to the switch property described in the introductory paragraph. Notably, HRTP applies the distributed conception to lower the workload of the conventional centralized RT management structure and simplify the RT

### OSI Model

The OSI model [37] is a reference model developed by the International Organization for Standardization in 1984 as a conceptual framework of standards for communication in a network that links various equipment and applications from different vendors. It is now considered to be the primary architectural model for intercomputing and inter-networking communications. The structures of most of the network communication protocols used today are based on the OSI model. The OSI model defines the communication process as involving seven layers and divides the tasks involved with moving information between networked computers into seven smaller, more manageable task groups. A task or group of tasks is then assigned to each of the seven OSI layers. Each layer is reasonably self-contained so that the tasks assigned to each layer can be implemented independently. This enables the solutions offered by one layer to be updated without adversely affecting the other layers.

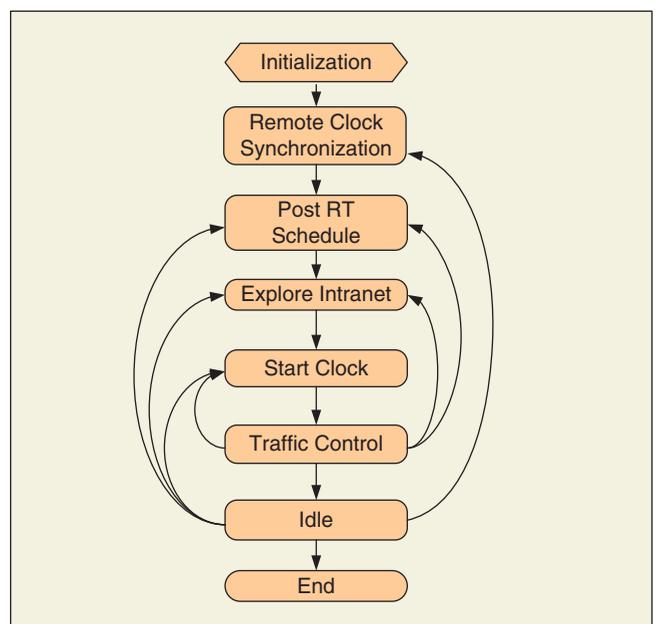
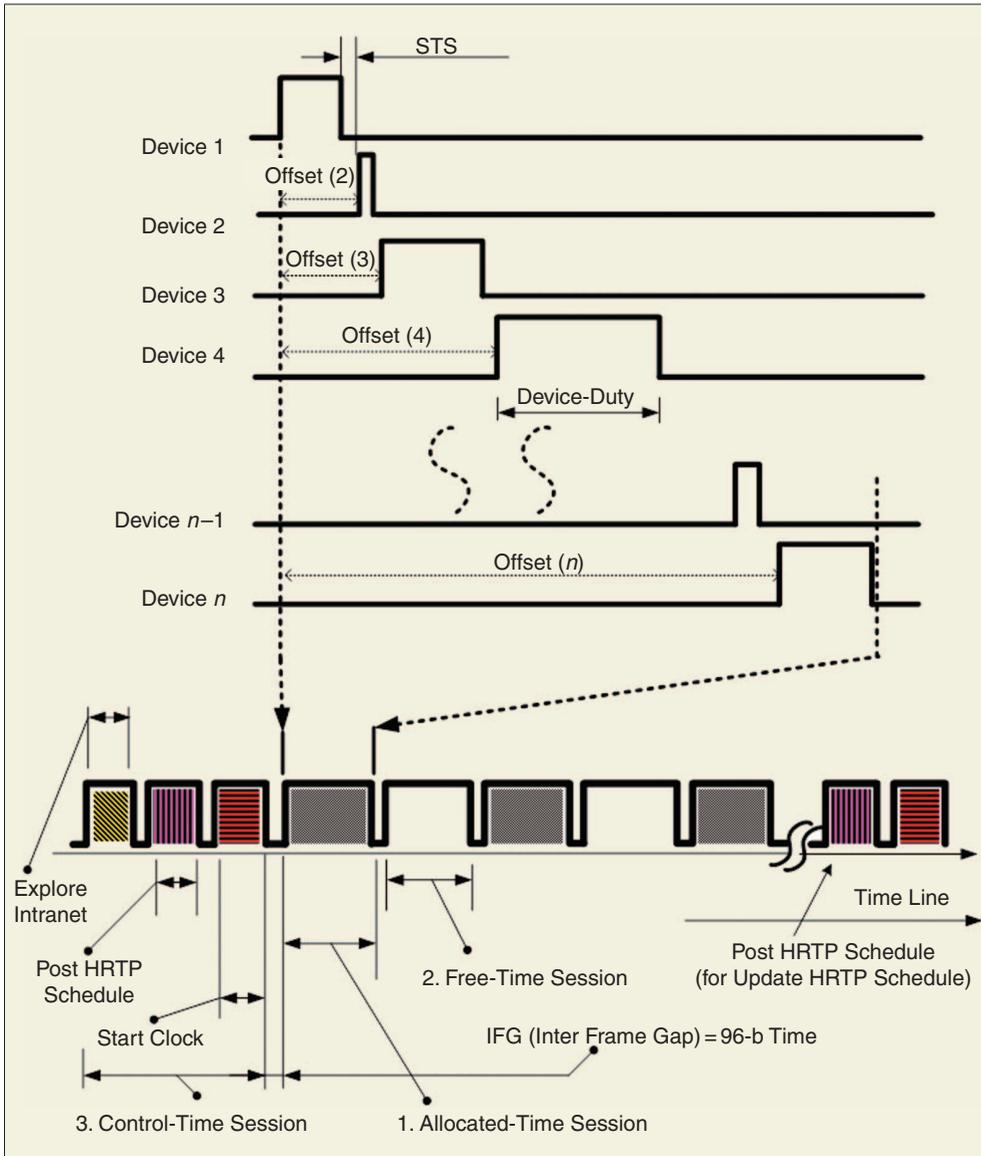


Figure 4. State machine of HRTP.



**Figure 5.** H RTP time wave structure based on packet traffic control technique approach.

schedule. Hence, the right of management is dispatched to each received node. Further, the rights of management contain governing the events trigger of overall state machine, generating the packet content of RT schedule and posting the packet of RT schedule.

### TDMA-E

TDMA-E is Ethernet transmission technology that allows a number of RTE sending nodes to access the respective Ethernet receive nodes without interference by allocating unique time slots to each RT node within each collision domain. Since the current Ethernet switch can provide one transmitting collision domain per port, allowing complete elimination of the collisions if the port is in full duplex (IEEE 802.3x) [36], and is connected to only one station, the TDMA-E transmission scheme multiplexes packets over different collision domains in a network switch (see “Network Switch”). Figure 23 shows an example of TDMA-E in one network switch.

### H RTP Time Wave Structure

The H RTP-task time wave structure is presented to describe further the time control mechanism of H RTP. The H RTP time wave structure is based on packet traffic control technique. A time-line is divided into three sessions using time-triggered architecture (Figure 5). These are 1) allocated (fixed)-time session, 2) free-time session, and 3) control-time session. Each session is explained later.

#### Allocated (Fixed)-Time Session

The allocated-time session mainly allocates to each sending node a time slot in which to transmit RT data (device 1 ~ device  $n$ , as illustrated in Figure 5). Consecutive sessions are separated by safety-time sections (STS), defined as

$$STS = T_{IFG} + T_{BTMT}, \quad (1)$$

where  $T_{IFG}$  represents the interframe gap (IFG) and  $T_{BTMT}$  denotes the NIC’s buffer transfer maxima time (NIC\_BTMT). The IFG is 96 b in length [36]. The NIC\_BTMT is defined by the MAC ring buffer. NIC\_BTMT is related to the

MAC device and the supporting driver and reserves a short span of time for transmitting packets under the MAC ring buffer. Therefore, the STS can circumvent the ring buffer overrun problem. The wake-up of the device duty is managed by the variable offset ( $n$ ), as indicated in Figure 5.

Furthermore, the allocated-time session is mainly based on the time division multiple access Ethernet (TDMA-E) model concept (see “TDMA-E”) to simplify and govern the distributed embedded Ethernet packet traffic.

#### Free-Time Session

The free-time session operates as in the standard IEEE 802.3 [36] and allows every device to send packets unrestricted by the CSMA/CD protocol. The principal service transmits non-RT packets. In the free-time session, all devices compete for time slots to transmit using the conventional CSMA/CD protocol. Its objective is to preserve the original CSMA/CD flexibility on H RTP.

### Control-Time Session

The control-time session is defined to detect and control the device nodes in intranet, the command of post H RTP schedule and start clock (Figures 4 and 5), and occurs at the beginning of each Ethernet traffic period. This controlling and detecting of behavior is regarded as explorer intranet (Figure 4). The transmission is governed by the H RTP time schedule and driven by time events. The traffic control model is elucidated next.

### H RTP Traffic Control Model

The previous sections described the concepts that underlie H RTP. This section introduces the operation of H RTP in real hardware. The hardware implementation of H RTP is associated with a traffic control model and is called the hardware RT traffic control model (H RTP-TCM). Figure 6 illustrates the H RTP-TCM, which consists of six main modules.

- ◆ *H RTP timer*: It is the RT timer in each embedded Ethernet device.
- ◆ *H RTP schedule*: It stores and governs the schedule of H RTP.
- ◆ *Traffic switch*: It controls the traffic flow. According to H RTP timer and H RTP schedule, traffic switch controls the passing through and blocking of Ethernet packets.
- ◆ *FIFO buffer*: FIFO stands for first in/first out and refers to the way in which the NIC processes data. A FIFO buffer is a memory device that allows flow from the CPU to the network controller and vice versa to be controlled and ensures packet integrity.
- ◆ *General-packet driver*: It includes TCP/UDP/IP packet driver.
- ◆ *H RTP-communications interface*: It receives H RTP command packets during control-time session.

However, the H RTP protocol can only be optimized by synchronizing all network devices. When the clocks are not synchronized, H RTP merely provides a new, reasonable, and

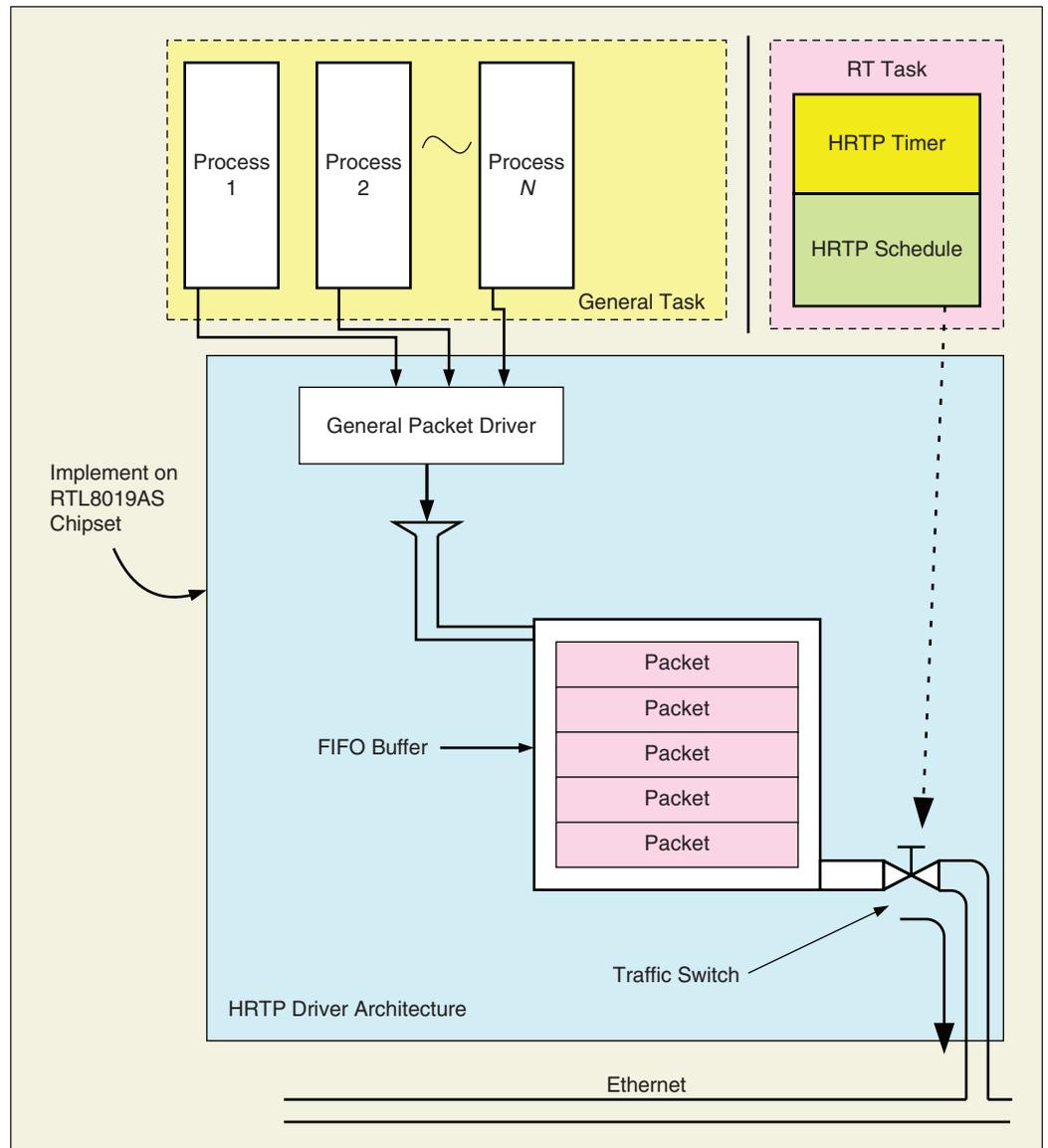


Figure 6. RT traffic control model of H RTP.

effective scheme to plan the use of intranet transmission resources. Figure 7 describes packet transmission with reference to the RT traffic control model when the H RTP schedule is changed.

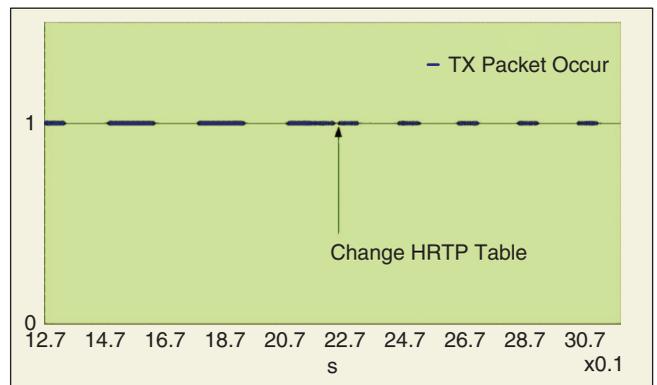
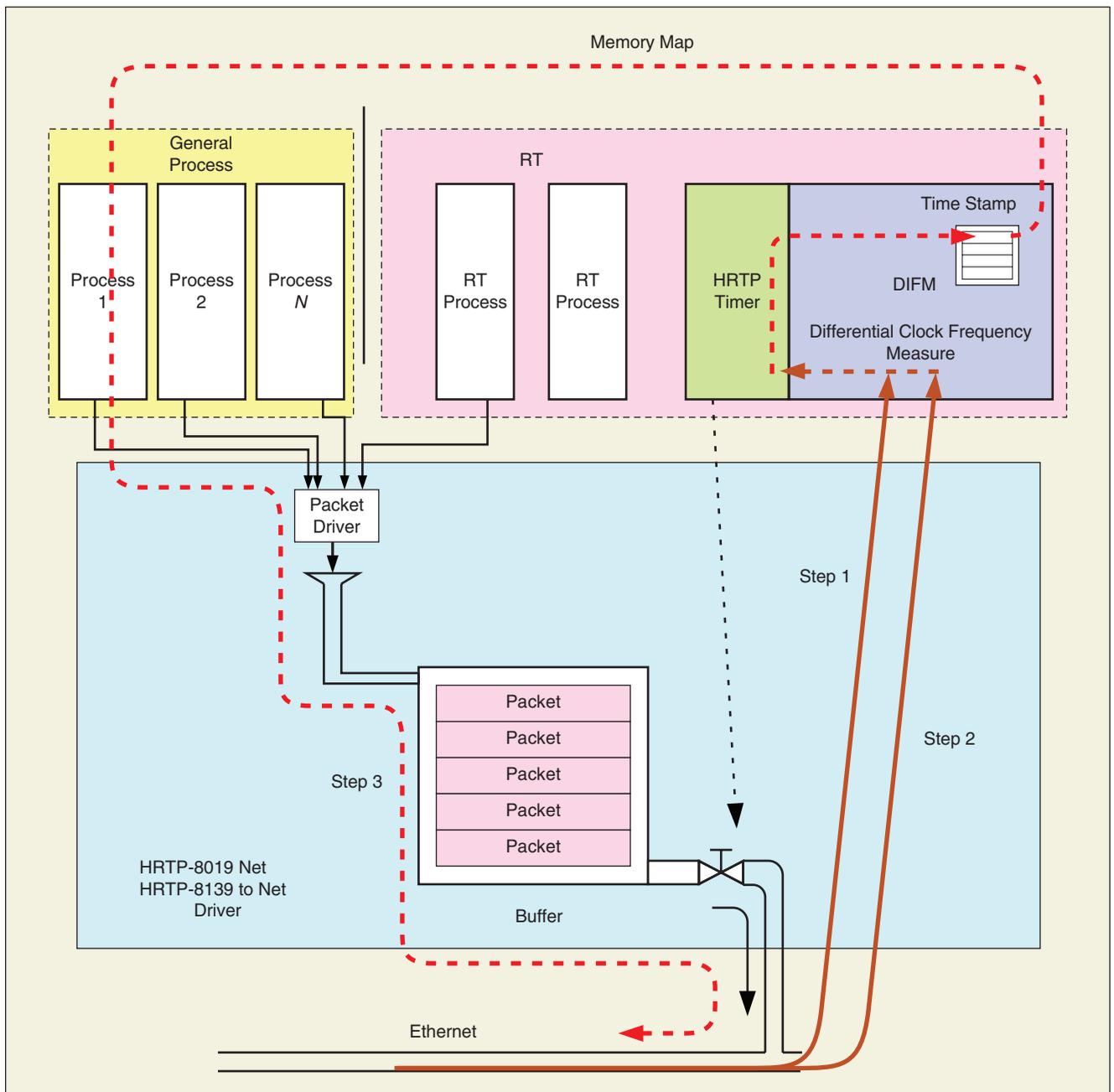


Figure 7. Packets transmitted using H RTP-TCM and instantaneous changes to the H RTP schedule.

## Clock Synchronization

TCP/IP and CSMA/CD are the two factors that most strongly influence timing control. Timing precision depends on the application. In an H RTP system, the RT traffic controller modes are designed to be distributed. The remote clocks are not always synchronized. Clock synchronization is extremely important in ensuring that a traffic event is performed correctly. Minimizing the communications delay is generally a critical design consideration. However, in control applications, the deterministic response time requirement is frequently more important than the minimization of delay. The network time protocol (NTP) is used to synchronize the time of a computer client or server with another server or reference

time source, such as a radio or satellite receiver or modem. Typical NTP configurations utilize multiple redundant servers and diverse network paths to obtain high accuracy and reliability [39]–[41]. However, NTP alone is not intended for high-reliability RT applications because it does not ensure the precision of the global time base [20]. For the proposed application, a differential clock methodology, similar to the IEEE 1588 Standard [42], was adopted. The internal clock is synchronized by adjusting the local clock of every participating node to the local clock of a specific master node. This method is called the master-worker scheme [20]. This protocol is quite simple and easy to implement. Figure 8 presents this synchronization mechanism within H RTP and synchronization steps. On the



**Figure 8.** Clock synchronization model in H RTP. (The differential time is calculated in Step 1 and Step 2, and the result corrects the H RTP timer. Step 3 reports the clock ticks to the network.)

platform designed in this work, the resulting clock synchronization error is approximately  $\pm 0.5$  ms (see the “Implementation” section).

### Content of H RTP Control Packet

Generally, H RTP defines every receive node as a host node to manage its client nodes. Thus, H RTP allows the embedded Ethernet robot system to consist of several receive nodes in a network switch. Accordingly, H RTP also provides three packets to manage each subsystem and the timing control.

### Request Report IP Data Command

Figure 9 proposes the request report IP data command packet, which is the first command packet that initializes the H RTP-embedded Ethernet systems. The hosts (defined as the receive nodes in the network switch) initiate the transmission of request report IP data command packets. This packet requests the client’s IP address and numbers of the devices from a management host LAN device and sends this information to every client. Every client device that receives such a packet replies with the same packet to the management host LAN device. The host management LAN device records these IP addresses to access the LAN status.

### H RTP Posting Schedule Command

The main function of the H RTP schedule command packet is to describe event-driven traffic switches. This packet fully describes the sliced segment of the H RTP-task time wave structure and has a variable packet length (Figure 10). Free-time and allocated-time sessions are both 32 b long. The duty cycle of the device is defined by

$$\text{Device duty} = \frac{T_{RT}}{T_{ALT} + T_{FT}}, \quad (2)$$

where  $T_{RT}$  denotes the length of the allocated-time session for each node and  $T_{ALT}$  and  $T_{FT}$  are the lengths of the entire allocated-time and free-time sessions, respectively. In the H RTP posting schedule command content, Device ( $n$ ) Duty is defined as (2). Additionally, Figure 5 defines offset ( $n$ ) as the offset time value of device  $n$ , whose traffic switch is turned on during an allocated-time session.

As stated in the introductory paragraphs, a current network switch can provide a separate transmitting collision domain for each port. Thus, the H RTP exploits this characteristic of network switch and TDMA-E to govern the RT schedule (see “TDMA-E”). In the process, each received node generates the schedule and manages the Ethernet traffic

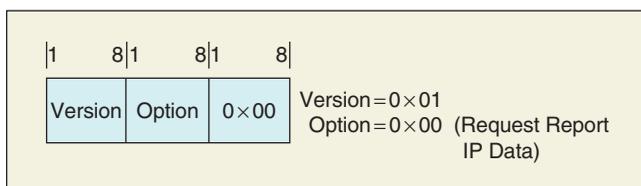


Figure 9. Content of request report IP data.

of all sending nodes. However, a network switch generally has some receive nodes and some sending nodes. And the low-end embedded Ethernet solutions usually have limits in handling the Ethernet packets and information processing. Hence, the Ethernet packet schedule should satisfy the following two constraints:

$$DR \leq MGS(i) \cdot TSL, \quad (3)$$

where  $i = 1, 2, \dots, m$ , and

$$MHR \geq \sum_{i=1}^n MGS(i), \quad (4)$$

where the parameters of (3) and (4) are defined as follows.

- 1)  $DR$ : Overall information processing Demand Rates of each receive node in its respective collision domain (in B/s)
- 2)  $TSL$ : H RTP Time Slot Length (traffic on) of each node in one second
- 3)  $MGS(i)$ : Maximum Generating rates of information of the Sending node ( $i$ ) (in B/s)
- 4)  $MHR$ : Maximum information Handling Rates of receive node.

Equation (3) claims that the sending node’s ( $i$ ) generation of information satisfies process demands. Equation (4) claims that the packet flow of receive node ( $n$ ) may be smooth and capable of being processed. Each receive nodes in a switch will generate the respective H RTP schedule command packet and send it to its client nodes. Accordingly, the H RTP mechanism can schedule and guarantee that the entire packet transmission process of embedded Ethernet devices in each switch collision domain is in real time.

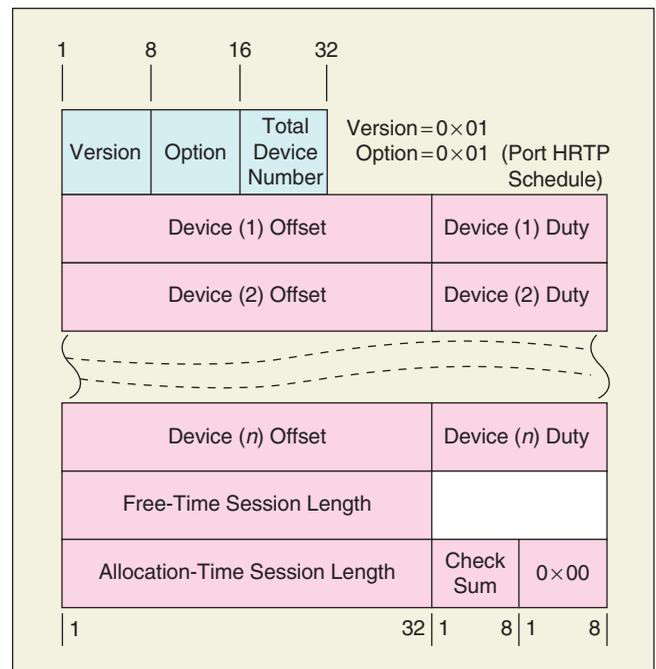


Figure 10. Content of H RTP posting schedule command.

### Start Clock Command

The primary function of this packet is to fire up all of the H RTP client timers. Figure 11 shows the content of start clock command.

In summary, the H RTP utilizes the self-splitting separate collision domains property of network switch technology and elaborately controls the traffic of packets of every embedded Ethernet device to ensure RT operation. The distributed

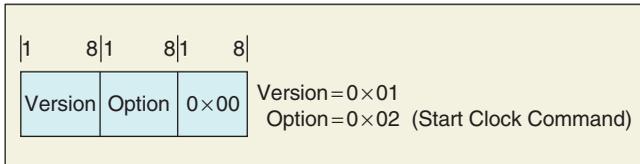


Figure 11. Content of counter start command.

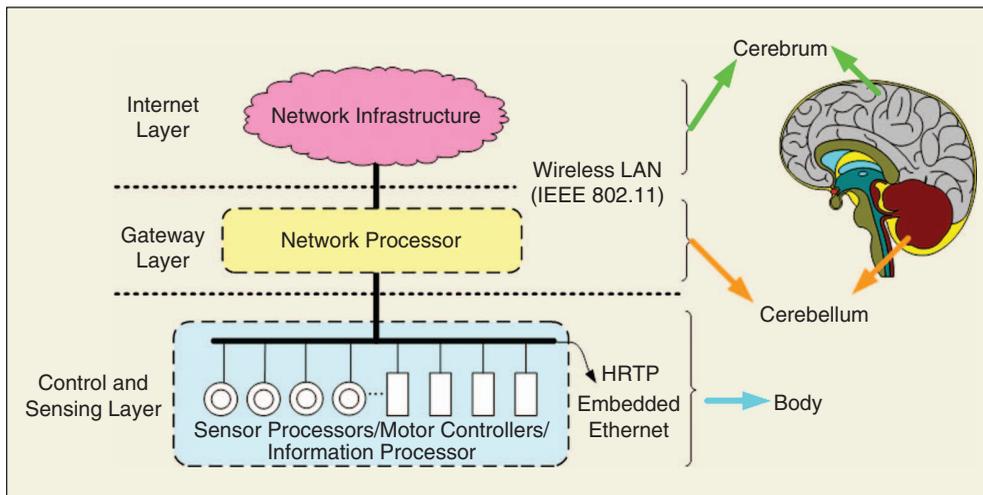


Figure 12. Novel distributed embedded platform using Ethernet as the communications backbone and based on a three-layer concept: 1) control and sensing layer, 2) gateway layer, and 3) Internet layer. This proposal regards the cerebrum, cerebellum, and nerve as abstract concepts. An RT protocol named H RTP is proposed in the control and sensing layer.

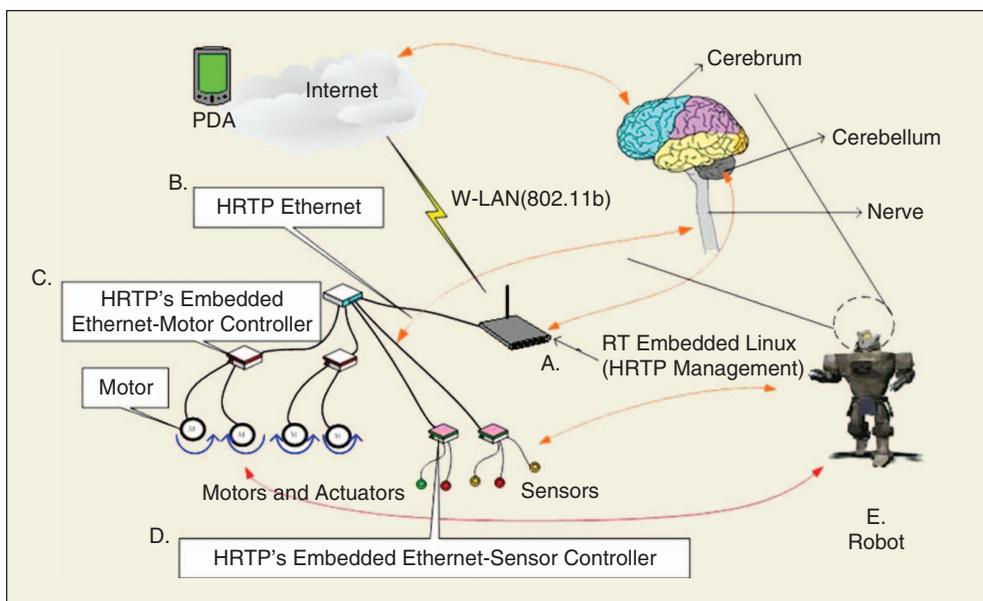


Figure 13. Conception of the platform and materialization.

tiny Ethernet RT protocol architecture for low-end embedded Ethernet devices and small robot systems is presented earlier. The following section will present the implementation in small robot systems.

## Implementation

### Overview

The proposed RT protocol is implemented on a small quadruped machine to enable transparent machine interconnections with the embedded RTE network environmental in frastructure (H RTP).

Nature provides very good design samples. The cerebrum, cerebellum, and nerves are metaphors for the computing power and database of computers on the Internet (Figure 12). Since the proposed novel distributed platform provides a wireless LAN interface, this platform concept is easy to integrate with other Internet devices.

Figure 13 displays a simple schematic diagram of the robot platform. In Figure 13, A is the host processor, running embedded RT Linux and serving as the central processing unit and a gateway or bridge to the outer world. B represents a hub for network expansion. Both C and D are microcontrollers, which handle motor control and sensing. E represents a robot in which the systems (A–D) are installed.

### Electronic

Figure 14 presents the design diagram of an embedded H RTP Ethernet controller and multimotor controller module microcontrollers (parts C and D in Figure 13). The embedded H RTP Ethernet controller, which consists of an 8-b micro control unit (MCU) (PIC-18F452), is based mainly on the sensor network protocol (I<sub>2</sub>C bus) and a serial motion control network. Each multi-microcontroller comprises an 8-b MCU (PIC-18F452), which controls eight motors and is integrated with numerous analog and digital interfaces, such as the pulse width modulation interface and the I<sub>2</sub>C EEPROM (Microchip

24LC256) file system. Figure 15 shows this module. Each embedded HRTTP Ethernet controller is interfaced with a 10-Mb/s Ethernet NIC (RTL-8019AS). A Media-GX (Geode CS5530A) Pentium-class SoC [43] is applied as the gateway-level network processor (Figure 1, Internet layer). Major peripherals that are connected to the SoC include wireless LAN, 10/100-Mb/s Ethernet NIC (RTL-8139), and the peripheral component interconnect bus.

### Robot Mechanical Design and System Integration

Figure 16 presents the computer-aided design (CAD) diagram of the four-footed machine, which we called O-Di robot (O-Di); each foot has 4 DoF. Each joint motor has one potential meter and one proportional-integral differential (PID) controller to control the position. Embedded HRTTP Ethernet is applied to update and reload sensor feedback to control O-Di's posture. Figure 17 schematically depicts the integration of the mechanical system with the electronic system. Figure 18 displays a photograph of the fully assembled machine.

### Software

The following software or firmware modules were installed to demonstrate the proposed system:

- 1) firmware on the microcontroller, including the motor control, the NIC driver, and a lean TCP/IP protocol stack [44] with HRTTP modification
- 2) an embedded RT Linux system, running on the Pentium-class SOC
- 3) a modified Linux TCP/UDP/IP protocol stack to provide HRTTP capability
- 4) a program to integrate robot network bridging and motion control functions
- 5) a software package for motion control and running on a PC.

The design architecture allows motion control software to be implemented easily on a PC and contains 16 slider bars to manage each individual motor directly. Additionally, it can also generate a database to record the state of motion and playback afterwards in a synchronized fashion. Figure 19 displays the overall system, including the remote control software. Although the present features are rather primitive, the platform clearly has significant potential for future expansion and software developments.

As mentioned earlier, the embedded Ethernet platform and HRTTP provide the system communication, integration, power saving, and low-cost advantage. Hence, the proposed

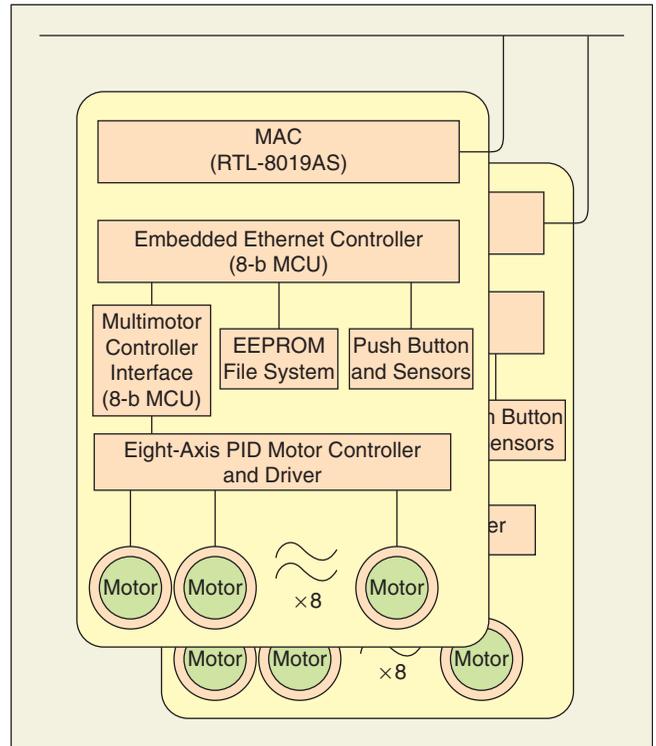


Figure 14. Schematic diagram of electronics.

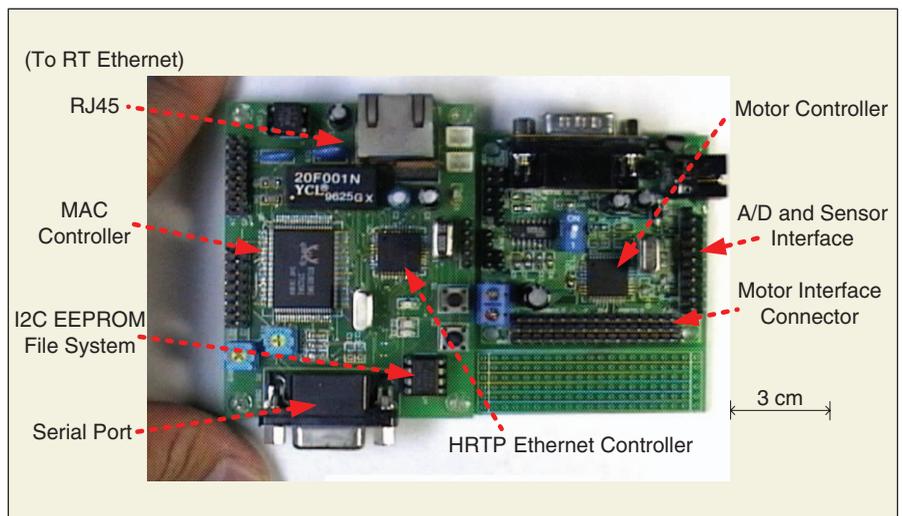


Figure 15. A prototype board with embedded HRTTP Ethernet controller and motor controller module.

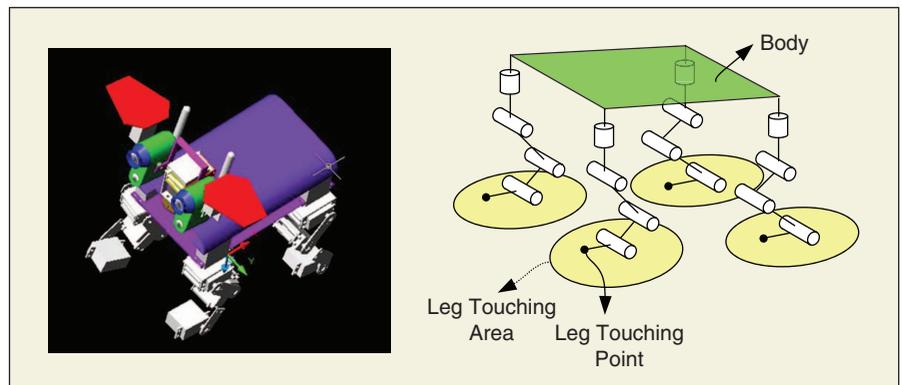


Figure 16. O-Di's CAD design and DoF.

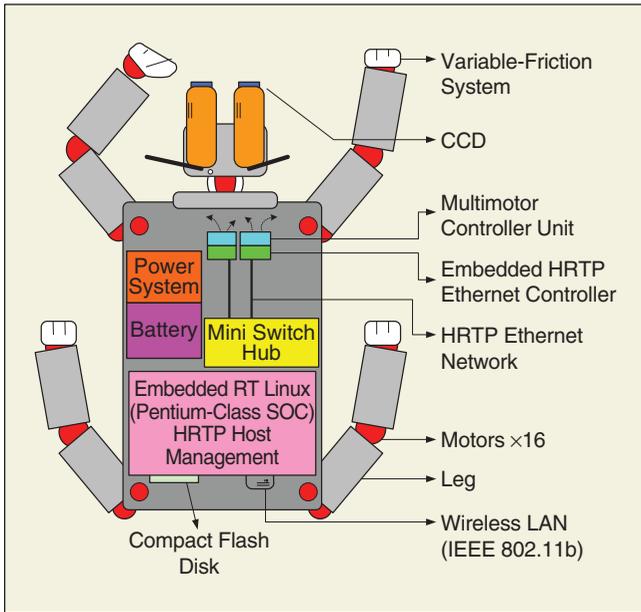


Figure 17. Schematic diagram of the O-Di.

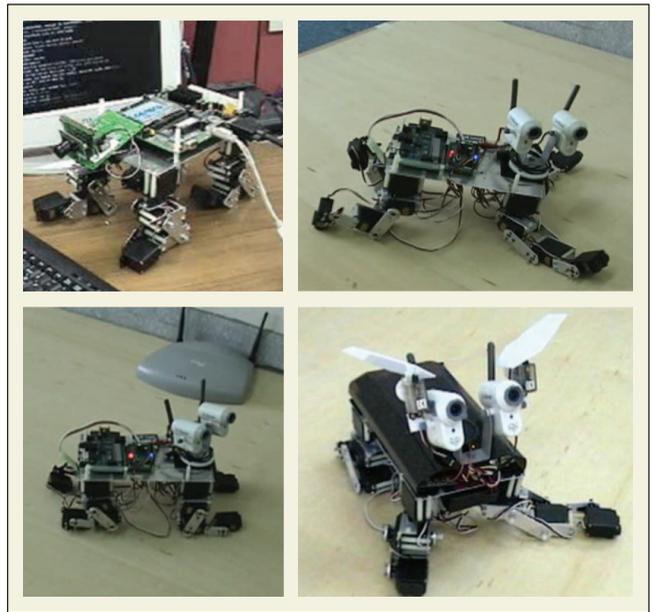


Figure 18. Fully assembled quadruped machine (O-Di).

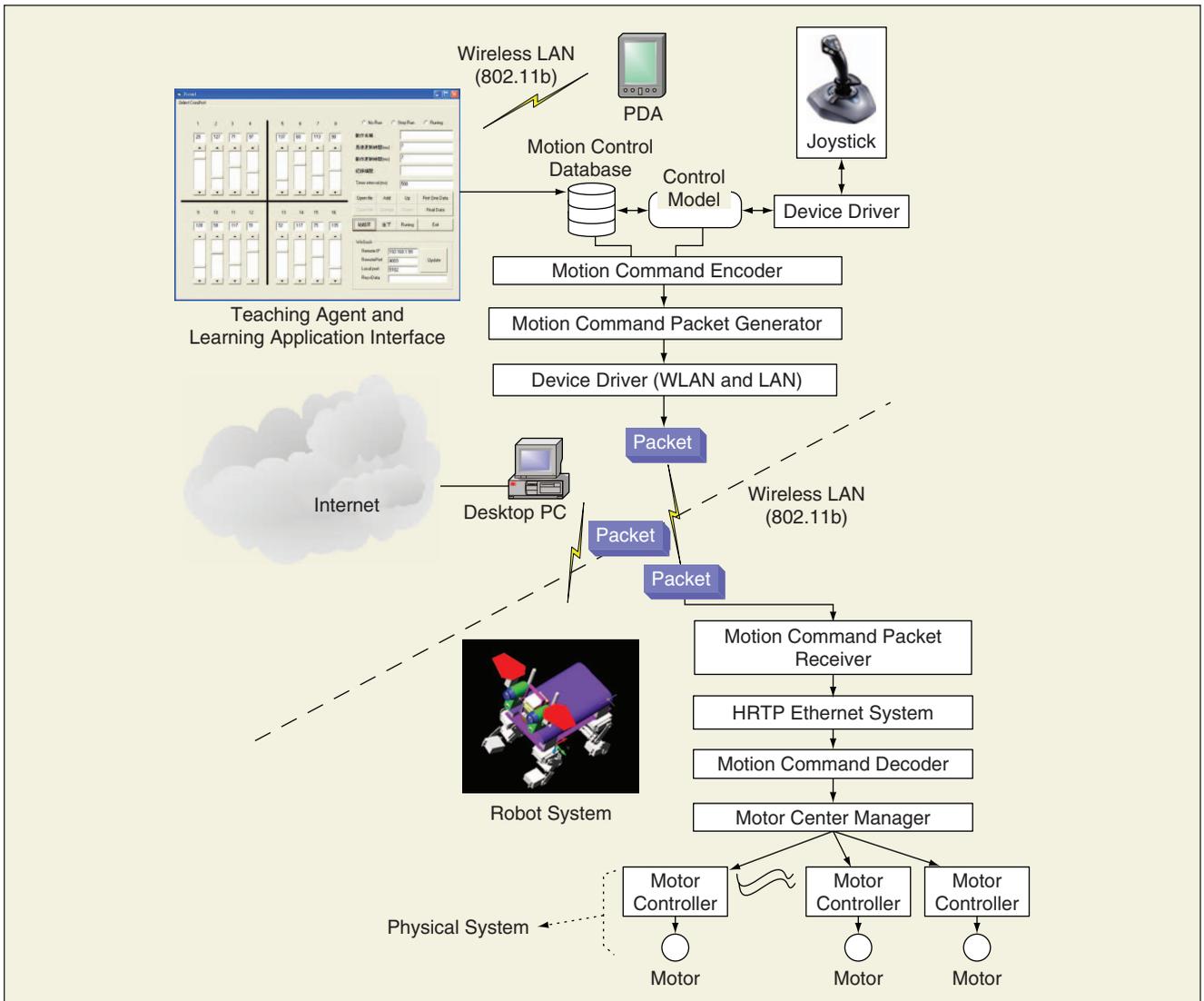
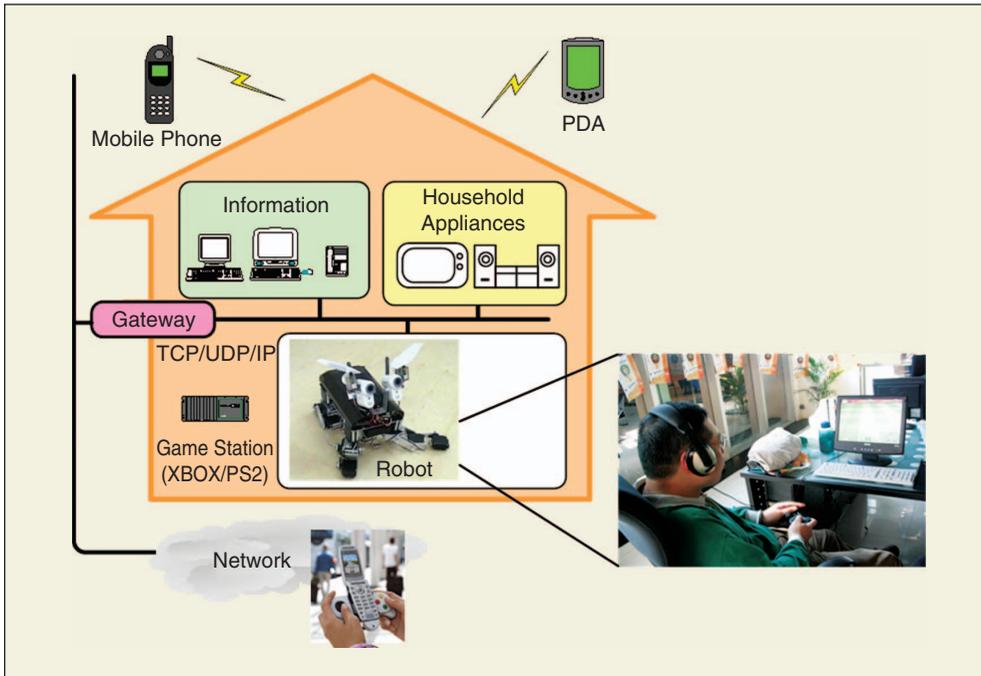


Figure 19. Overall system including PC-side remote control.

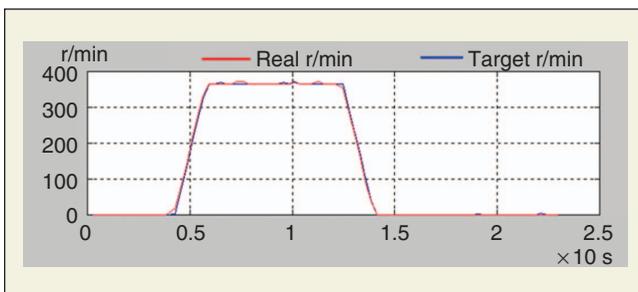


**Figure 20.** Applying propose robot network platform integrated with today's home network. Various network technologies can be interconnected in home network environments using TCP/UDP/IP.

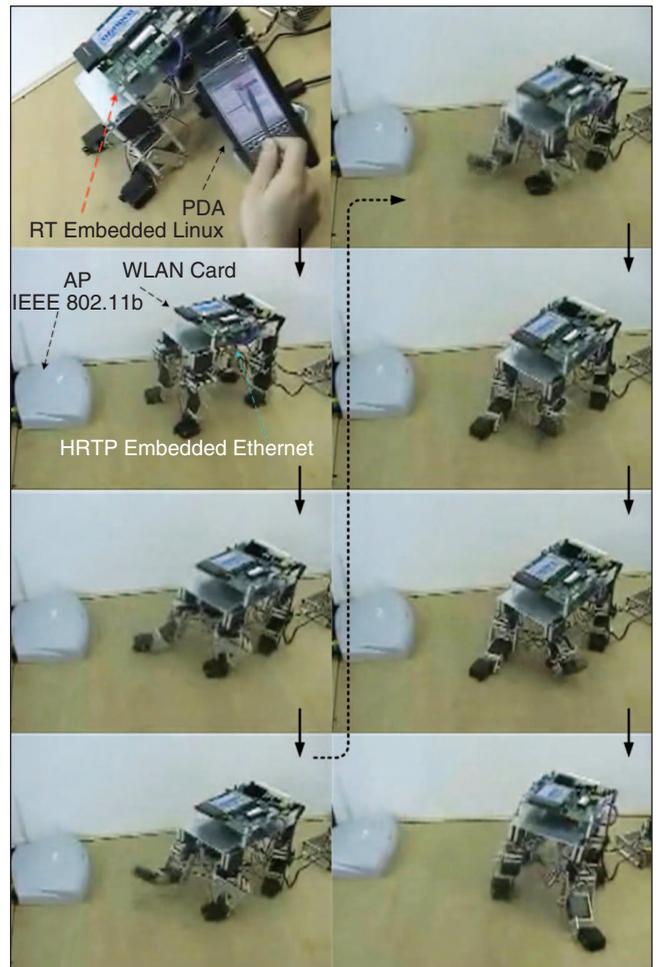
architecture has potential applications in the design architecture of home network-based robots. Furthermore, this robot platform can be easily integrated with today's home network. Various network technologies can be interconnected in home network environments using TCP/IP (Figure 20).

### Example of Motion and System Integration

The clock synchronization on the proposed platform has an accuracy of  $\pm 0.5$  ms for every 100 ms time-triggered protocol (TTP) cycle. The clock is resynchronized every 1 s, and the maximum error is 5 ms. The motion commands are issued every 50 ms. Restated, a 10% error in motion synchronization results from the timing mismatch. Because the system is low-end CPU, the result of performance is reasonable. For motion profiles, which do not need a high precision, this error is acceptable. The better performance for future can improve by improving CPU performance and raising clock frequency. Figure 21 displays a sample of the feedback control (using the PID algorithm to control a one-axial dc servo motor) in the proposed network system. The HRTP can serve as a low-collision RT



**Figure 21.** A sample result of motor feedback control (using PID algorithm to control one-axial dc servo motor).

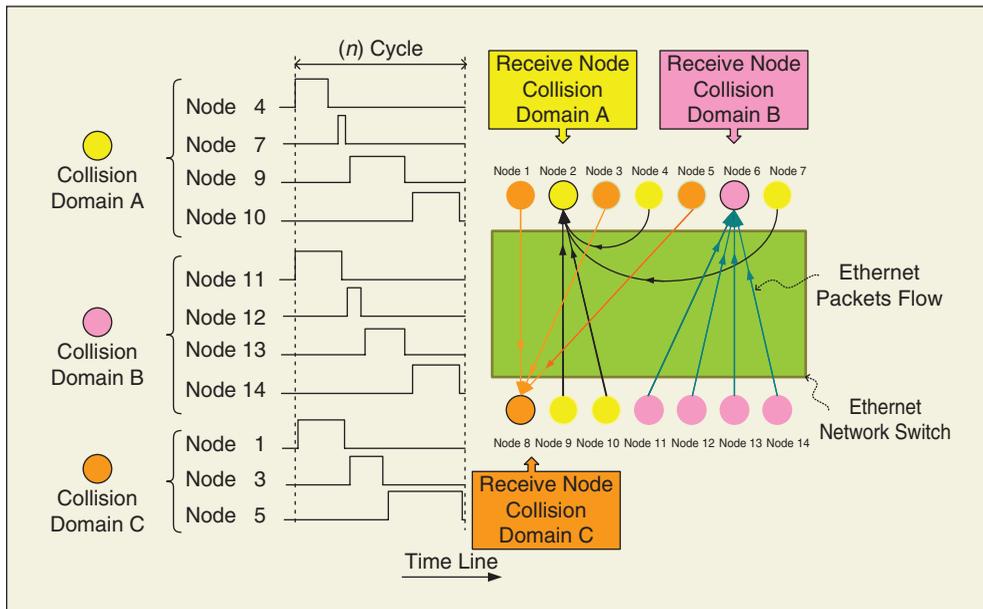


**Figure 22.** Robot walk around and PDA control robot via WLAN.

network environment. A two-wheel mode controller algorithm [45], [46] was implemented to demonstrate the proposed system's effectiveness. Figure 22 shows a sequence of pictures of the robot in motion. Full robot demo videos can be seen in the Web site <http://xlab.cn.nctu.edu.tw/Liwei/eRobot.htm> [47].

### Conclusions

This study presents a novel distributed embedded platform using Ethernet as the communications backbone, with three layers: 1) control and sensing layer, 2) gateway layer, and 3) Internet layer. This proposal regards the cerebrum, cerebellum, and



**Figure 23.** The three collision domains A, B, and C. The packet flows in an Ethernet switch. The current Ethernet switch technique can provide one transmitting collision domain per port (dedicated bandwidth segment), allowing the complete elimination of collisions if the port is in full duplex (IEEE 802.3x) and only one station is connected to it. The TDMA-E is transmission technology that allows a number of RT nodes to access the Ethernet node without interference, by allocating unique time slots to each RT node within each collision domain. However, it allows only one RT node transmission to access the respective Ethernet receive node in each collision domain. TDMA-E is most responsible for regularizing packet traffic through the group nodes in independent collision domains (first part).

nerve as abstract concepts. An RT protocol named HRTP is proposed in the control and sensing layer. The HRTP is a distributed RT protocol with a small footprint, which is especially suitable for the embedded and distributed network-based robot control application. HRTP is the first distributed RTE protocol stack specifically for an embedded Ethernet network. Notably, traditional RTE technologies have centralized structures that are large and superfluous for small robot systems. The proposed HRTP, in contrast to a distributed RT protocol, is for embedded RTE environments and robot network development.

A quadruped walking machine (O-Di) and an electronic system were designed to implement the proposed platform. Software modules were installed to demonstrate the overall system. On the basis of the three-layer concept, this model can be imitated to design more complex robot systems. Future research will involve designs of more complex machines to demonstrate the capability of the distributed embedded platform.

## Acknowledgments

Part of this work was presented at the 2006 IEEE International Conference on Robotics and Automation (15–19 May 2006) and the IEEE International Conference on Mechatronics (10–12 July 2005). This work led to Taiwan patent no. I238622, Taiwan patent no. I257214, U.S. patent application no. 11/296,217, and U.S. patent application no. 11/370,929. The authors thank the National Science Council of the Republic of China, Taiwan (contract no. NSC92-2213-E-009-004 and NSC94-2218-E-009-064), the

Ministry of Education of the Republic of China, Taiwan (contract no. 91-E-FA06-4-4), and Aiming for the Top University and Elite Research Center Development Plan program under the account no. 95W803E for financially supporting this research. This article has author-provided supplementary downloadable material available at <http://xlab.cn.nctu.edu.tw/Liwei/eRobot.htm>.

## Keywords

Network robot, embedded Ethernet, real-time network, distributed control, HRTP, O-Di robot, quadruped robot, real-time Ethernet, home robot.

## References

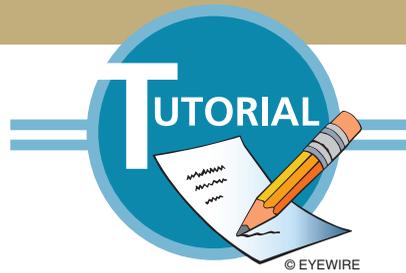
- [1] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent ASIMO: System overview and integration," *Intell. Robots Syst.*, vol. 3, pp. 2478–2483, 2002.
- [2] T. Ishida, Y. Kuroki, J. Yamaguchi, M. Fujita, and T. T. Doi, "Motion entertainment by a small humanoid robot based on OPEN-R," in *Proc. 2001 Conf. Intelligent Robots and Systems*, vol. 2, pp. 1079–1086.
- [3] T. Makimoto and T. T. Doi, "Chip technologies for entertainment robots—Present and future," in *Tech. Dig. Electron Devices Meeting, IEDM'02*, 2002, pp. 9–16.
- [4] B. H. Lee, "Embedded Internet systems: Poised for takeoff," *IEEE Internet Comput.*, vol. 2, pp. 24–29, 1998.
- [5] R. E. Filman, "Embedded Internet systems come home," *IEEE Internet Comput.*, vol. 5, no. 1, pp. 52–53, 2001.
- [6] J. Axelson, *Embedded Ethernet and Internet Complete: Designing and Programming Small Devices for Networking*. Madison, WI: Lakeview Research, 2003.
- [7] D. L. Sancho-Pradel, S. R. Jones, and R. M. Goodall, "System on programmable chip for real-time control implementations," in *Proc. 2002 IEEE Int. Conf. Field-Programmable Technology*, pp. 276–283.
- [8] S. Perez and J. Vila, "Building distributed embedded systems with RTLinux-GPL," in *Proc. 2003 IEEE Conf. Emerging Technologies and Factory Automation*, 2003, vol. 1, pp. 161–168.
- [9] J. S. Jwo, S. S. Chang, Y. C. Chen, and D. F. Hsu, "A distributed environment for hypercube computing," in *Proc. 2nd Aizu Int. Symp. Parallel Algorithms/Architecture Synthesis*, 1997, pp. 256–263.
- [10] J. Wang, (2002). Soft real-time switched ethernet: best-effort packet scheduling algorithm, implementation, and feasibility analysis, M.S. thesis. Dept. Electr. Comput. Eng., Faculty of the Virginia Polytechnic Institute and State Univ. [Online]. Available: <http://scholar.lib.vt.edu/theses/available/etd-10032002-152839/>
- [11] K. Schrodinger, J. Stimma, and M. Mauthe, "A fully integrated CMOS receiver front-end for optic gigabit Ethernet," *IEEE J. Solid State Circ.*, vol. 37, no. 7, pp. 874–880u, 2002.
- [12] B. Blaner, D. Czenkusch, R. Devins, and S. Stever, "An embedded PowerPCTM SOC for test and measurement applications," in *Proc. 13th Annu. IEEE Int. ASIC/SOC Conf.*, 2000, pp. 204–208.

- [13] J. Bilek and I. P. Ruzicka, "Evolutionary trends of embedded systems," in *Proc. 2003 IEEE Int. Conf. Industrial Technology*, vol. 2, pp. 901–905.
- [14] M. Molle, "100Base-T/IEEE802.12/packet switching," *IEEE Commun. Mag.*, vol. 15, no. 10, pp. 64–73, 1996.
- [15] C. Venkatramani, "Implementation and evaluation of RETHER: A real-time Ethernet protocol," Ph.D. dissertation, State Univ. New York at Stony Brook, 1996.
- [16] C. Venkatramani and T. S. Chiueh, "Supporting real-time traffic on Ethernet," in *Proc. 15th IEEE Real-Time Systems Symp.*, 1994, pp. 282–286.
- [17] S. Varadarajan and T. Chiueh, "Ethereal: A host-transparent real-time fast Ethernet switch," in *Proc. IEEE Int. Conf. Network Protocols*, 1998, pp. 12–21.
- [18] T. C. Chiueh and C. Venkatramani, "Supporting real-time traffic on Ethernet," in *Proc. IEEE Real-Time Symp.*, 1994, pp. 282–286.
- [19] H. Scholten, P. G. Jansen, F. Hanssen, and T. Hattink, "RTnet: A new approach to in-home real-time multimedia communication," in *Proc. 12th Workshop on Local and Metropolitan Area Networks*, 2002, pp. 15–19.
- [20] K. H. K. Kim, C. Im, and P. Athreya, "Realization of a distributed OS component for internal clock synchronization in a LAN environment," in *Proc. IEEE Object-Oriented Real-Time Distributed Computing*, 2002, pp. 263–270.
- [21] M. Schwarz, "Implementation of a TTP/C cluster based on commercial gigabit Ethernet components," M.S. thesis, Institut für Technische Informatik, Vienna Univ. Technology, Austria, 2002.
- [22] M. Felser, "Real-time Ethernet—Industry perspective," *Proc. IEEE*, vol. 93, no. 6, pp. 1118–1129, 2005.
- [23] N. Brown and C. Kindel (1996). Distributed component object model protocol—DCOM/1.0, Internet Draft. [Online]. Available: <http://www.microsoft.com/oledev/olecom/draft-brown-dcom-v1-spec-01.txt>
- [24] P. Ferrari, A. Flammini, D. Marioli, and A. Taroni, "Experimental evaluation of PROFINET performance," in *Proc. 2004 IEEE Int. Workshop on Factory Communication Systems*, pp. 331–334.
- [25] J. Jasperneite and J. Feld, "PROFINET: An integration platform for heterogeneous industrial communication systems," in *Proc. 10th IEEE Conf. Emerging Technologies and Factory Automation (ETFA 2005)*, 2005, vol. 1, pp. 815–822.
- [26] J. Feld, "PROFINET-scalable factory communication for all applications," in *5th IEEE Int. Workshop on Factory Communication Systems (WFCS'2004)*, Vienna, Sept. 2004, pp. 33–38.
- [27] K. Schneider, "Intelligent field devices in factory automation—Modular structures into manufacturing cells," in *Proc. Emerging Technologies and Factory Automation (ETFA '03)*, 2003, vol. 1, pp. 101–103.
- [28] M. Popp and P. Wenzel, "PROFINet-linking worlds," in *Proc. 2001 8th IEEE Int. Conf. Emerging Technologies and Factory Automation*, vol. 2, pp. 519–522.
- [29] F. Hansen (2006). *Real-time Ethernet is reaching the field for industrial control* [Online]. Available: <http://www.rtc magazine.com/home/article.php?id=100489>
- [30] D. Jansen and H. Buttner, "Real-time Ethernet the EtherCAT solution," *IEE Comput. Contr. Eng. J.*, vol. 15, no. 1, pp. 16–21, 2004.
- [31] EtherCAT Technology Group. (2003). *EtherCAT—the Ethernet fieldbus*, [Online]. Available: [http://www.ethercat.org/pdf/pcc\\_ethercat\\_e.pdf](http://www.ethercat.org/pdf/pcc_ethercat_e.pdf)
- [32] Bernecker & Rainer (B&R). (2003). *ETHERNET powerlink user's manual* [Online]. Available: [http://www.br-automation.com/cps/rde/xchg/br-productcatalogue/hs.xml/services\\_66597\\_ENG\\_HTML.htm](http://www.br-automation.com/cps/rde/xchg/br-productcatalogue/hs.xml/services_66597_ENG_HTML.htm)
- [33] Bernecker & Rainer (B&R). (2004). *ETHERNET powerlink* [Online]. Available: [http://www.br-automation.com/cps/rde/xchg/br-automation\\_com/hs.xml/service\\_5129\\_ENG\\_HTML.htm](http://www.br-automation.com/cps/rde/xchg/br-automation_com/hs.xml/service_5129_ENG_HTML.htm)
- [34] *Real-Time Ethernet EPL (ETHERNET Powerlink), Proposal for a Publicly Available Specification for Real-Time Ethernet*, document IEC, 65C/356a/NP, 2004.
- [35] C. Schlegel (2005). *ETHERNET powerlink implementation strategies* [Online]. Available: <http://www.ethernet-powerlink.org/index.php?id=35>
- [36] *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specification[S]*. IEEE Standard 802.3x2002.
- [37] G. Held, *Ethernet Networks: Design, Implementation, Operation, Management*. Hoboken, NJ: Wiley, 2003.
- [38] Lantronix. (2004). *Networking tutorials—Network switching* [Online]. Available: <http://www.lantronix.cn/learning/tutorials/switching.html>
- [39] S. Johannessen, "Time synchronization in a local area network," *IEEE Control Sys. Mag.*, vol. 24, no. 2, pp. 61–69, 2004.
- [40] H. Melvin and L. Murphy, "Time synchronization for VOIP quality of service," *IEEE Internet Comput.*, vol. 6, no. 3, pp. 57–63, 2002.
- [41] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, p. 1482, 1991.
- [42] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Standard 1588–2004, Nov. 8, 2002.
- [43] Advanced Micro Devices Inc. (2004). *AMD Geode(R) GX processors data book* [Online]. Available: [http://www.amd.com/files/connectivitysolutions/geode/geode\\_gx/31505E\\_gx\\_databook.zip](http://www.amd.com/files/connectivitysolutions/geode/geode_gx/31505E_gx_databook.zip)
- [44] J. Bentham, *TCP/IP Lean: Web Servers for Embedded Systems*. 2nd ed., Lawrence, KS: CMP Books, 2002.
- [45] D. Golubovic and H. Hu, "A hybrid evolutionary algorithm for gait generation of Sony legged robots," in *Proc. IECON 2002*, pp. 2593–2598.
- [46] C. Asghari, D. Golubovic, B. Li, and H. Hu, "A hybrid software platform for Sony AIBO robots," in *7th Int. Symp. RobotCup*, Padua, Italy, 2003.
- [47] L.-W. Wu and J.-S. Hu. *Demo video: A distributed embedded control platform for robots using real-time Ethernet* [Online]. Available: <http://xlab.cn.nctu.edu.tw/Liwei/eRobot.htm>

**Li-Wei Wu** received his B.S. degree in mechanical engineering from Chinese Culture University, Taiwan, Republic of China, in 1998 and his M.S. degree in mechanical and aerospace engineering from Chung Hua University, Taiwan, Republic of China, in 2000. He received his Ph.D. degree from the Department of Electrical and Control Engineering at the National Chiao-Tung University in 2006. He received the Honorable Mention of the Chinese Taiwan Ministry of Education (championship) in the 2003 Embedded Software Design Contest, which was sponsored by the Ministry of Education Advisory Office and the SoC Consortium. He also received the Honorable Mention of the Chinese Taiwan Ministry of Education in the 2005 Intelligent Robot Contest. His research interests include robotics, RT network protocol, embedded Ethernet, RT systems, and embedded system design and implementation. Currently, he is a senior engineer at Compal Communications.

**Jwu-Sheng Hu** received the B.S. degree from the Department of Mechanical Engineering, National Taiwan University, Taiwan, in 1984 and the M.S. and Ph.D. degrees from the Department of Mechanical Engineering, University of California at Berkeley, in 1988 and 1990, respectively. He is currently a professor in the Department of Electrical and Control Engineering, National Chiao-Tung University, Taiwan, Republic of China. His current research interests include microphone array signal processing, active noise control, embedded system design, and robotics.

**Address for Correspondence:** Li-Wei Wu, Department of Electrical and Control Engineering, National Chiao-Tung University, 1001 Ta-Hsueh Road, Hsinchu 30010, Taiwan. Phone: +3 571 2121. Fax: +3 571 5998. E-mail: [liwei0503@gmail.com](mailto:liwei0503@gmail.com), [liwei.ece89g@nctu.edu.tw](mailto:liwei.ece89g@nctu.edu.tw).



# Surgical and Interventional Robotics

## *Core Concepts, Technology, and Design*

BY PETER KAZANZIDES, GABOR FICHTINGER, GREGORY D. HAGER,  
ALLISON M. OKAMURA, LOUIS L. WHITCOMB, AND RUSSELL H. TAYLOR

Two decades after the first reported robotic surgical procedure [1], surgical robots are just beginning to be widely used in the operating room or interventional suite. The da Vinci telerobotic system (Intuitive Surgical, Inc.), for example, has recently become more widely employed for minimally invasive surgery [2]. This article, the first in a three-part series, examines the core concepts underlying surgical and interventional robots, including the potential benefits and technical approaches, followed by a summary of the technical challenges in sensing, manipulation, user interfaces, and system design. The article concludes with a review of key design aspects, particularly in the areas of risk analysis and safety design. Note that medical care can be delivered in a surgical suite (operating room) or an interventional suite, but for convenience, we will henceforth use the term *surgical* to refer to both the surgical and interventional domains.

### **Core Concepts**

This section describes some of the potential benefits of surgical robots, followed by an overview of the two technical paradigms, surgical computer-aided design and computer-aided manufacturing (CAD/CAM) and surgical assistance, which will be the subjects of the second and third articles in this series.

### **Potential Benefits**

The development of surgical robots is motivated primarily by the desire to enhance the effectiveness of a procedure by coupling information to action in the operating room or interventional suite. This is in contrast to industrial robots, which were

developed primarily to automate dirty, dull, and dangerous tasks. There is an obvious reason for this dichotomy: medical care requires human judgment and reasoning to handle the variety and complexity of human anatomy and disease processes. Medical actions are chosen based on information from a number of sources, including patient-specific data (e.g., vital signs and images), general medical knowledge (e.g., atlases of human anatomy), and physician experience. Computer-assisted interventional systems can gather and present information to the physician in a more meaningful way and, via the use of robots, enable this information to influence the performance of an intervention, thereby potentially improving the consistency and quality of the clinical result. It is, therefore, not surprising that surgical robots were introduced in the 1980s, after the dawn of the information age, whereas the first industrial robot was used in 1961.

There are, however, cases where surgical robots share potential benefits with industrial robots and teleoperators. First, a robot can usually perform a task more accurately than a human; this provides the primary motivation for surgical CAD/CAM systems, which are described later in the “Surgical CAD/CAM” section. Second, industrial robots and teleoperators can work in areas that are not human friendly (e.g., toxic fumes, radioactivity, or low-oxygen environments) or not easily accessible to humans (e.g., inside pipes, the surface of a distant planet, or the sea floor). In the medical domain, inhospitable environments include radiation (e.g., X-rays) and inaccessible environments include space-constrained areas such as the inside of a patient or imaging system. This also motivates the development of surgical CAD/CAM systems and is one of the primary motivations for surgical assistant systems, described in the “Surgical Assistance” section.

In contrast to industrial robots, surgical robots are rarely designed to replace a member of the surgical or interventional team. Rather, they are intended to augment the medical staff by imparting superhuman capabilities, such as high motion accuracy, or to enable interventions that would otherwise be physically impossible. Therefore, methods for effective human-robot cooperation are one of the unique and central aspects of medical robotics.

### Technical Paradigms

In our research, we find it useful to categorize surgical robots as surgical CAD/CAM or surgical assistance systems, based on their primary mode of operation [3]. Note, however, that these categories are not mutually exclusive and some surgical robots may exhibit characteristics from both categories. The following sections briefly describe these categories, with representative examples.

#### Surgical CAD/CAM

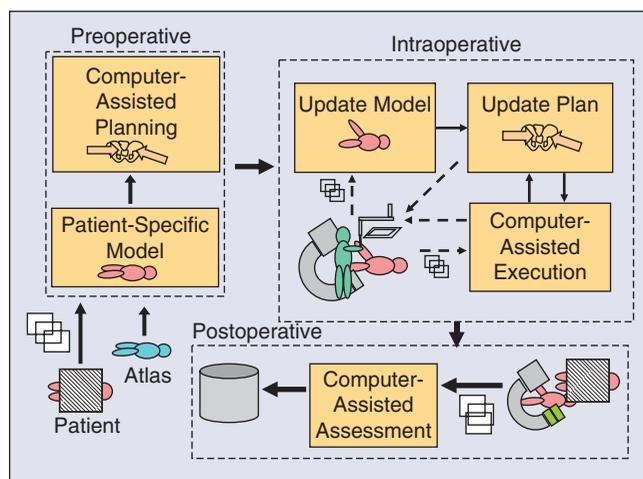
The basic tenet of CAD/CAM is that the use of a computer to design a part creates a digital blueprint of the part, and so it is natural to use a computer-controlled system to manufacture it, i.e., to translate the digital blueprint into physical reality. In the medical domain, the planning that is often performed prior to, or during, an intervention corresponds to CAD, whereas the intervention represents CAM. To take the analogy further, postoperative assessment corresponds to total quality management (TQM). We refer to the closed-loop process of 1) constructing a patient-specific model and interventional plan; 2) registering the model and plan to the patient; 3) using technology to assist in carrying out the plan; and 4) assessing the result, as surgical CAD/CAM, again emphasizing the analogy between computer-integrated medicine and computer-integrated manufacturing (Figure 1).

The most well-known example of a surgical CAD/CAM system is ROBODOC (ROBODOC, a Curexo Technology Company; formerly Integrated Surgical Systems, Inc.) [4], [5]. ROBODOC was developed for total hip and total knee replacement surgeries (Figure 2). In these surgeries, the patient's joint is replaced by artificial prostheses: for hip surgery, one prosthesis is installed in the femur and another in the acetabulum (pelvis) to create a ball and socket joint; for knee surgery, one prosthesis is installed in the femur and the other in the tibia to create a sliding hinge joint. Research on ROBODOC began in the mid-1980s as a joint project between IBM and the University of California, Davis. At that time, the conventional technique for hip and knee replacement surgery consisted of two-dimensional (2-D) planning

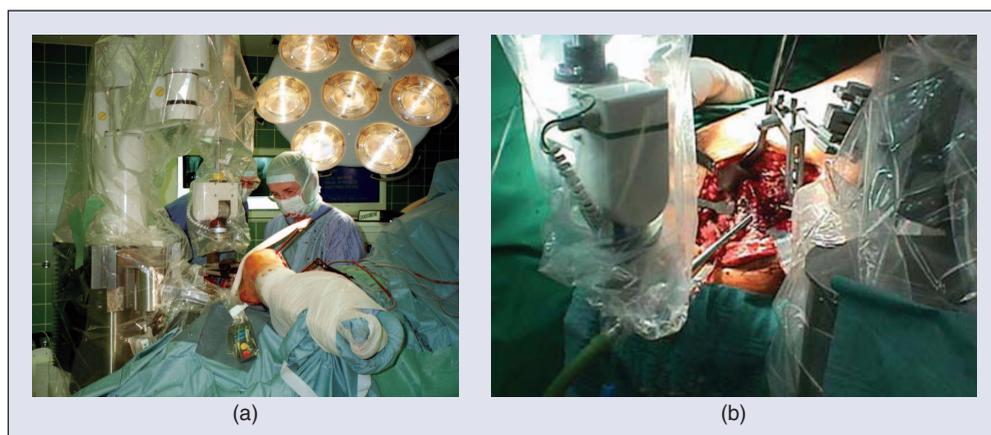
(using X-rays) and manual methods (handheld reamers and broaches) for preparing the bone. The motivation for introducing a robot was to improve the accuracy of this procedure—both the placement accuracy (to put the prostheses in the correct places) and the dimensional accuracy (to get a good fit to the bones). The technical approach of the system is to use computed tomography (CT) for three-dimensional (3-D) planning and a robot for automated bone milling. The planning (surgical CAD) is performed on the ORTHODOC workstation, which enables the surgeon to graphically position a 3-D model of the prosthesis (or prostheses) with respect to the CT image, thereby creating a surgical plan. In the operating room (Surgical CAM), the robot is registered to the CT image so that the surgical plan can be transformed from the CT coordinate system to the robot coordinate system. The robot then machines the bone according to the plan, using a high-speed milling tool.

#### Surgical Assistance

Medical interventions are highly interactive processes, and many critical decisions are made in the operating room and



**Figure 1.** Architecture of a surgical CAD/CAM system, where the preoperative phase is CAD, the intraoperative phase is CAM, and the postoperative phase is TQM.



**Figure 2.** The ROBODOC system for orthopedic surgery. (a) The robot is being used for total hip replacement surgery. (b) Close-up of robotic milling of femur.

## The da Vinci telerobotic system has recently become more widely employed for minimally invasive surgery.

executed immediately. The goal of computer-assisted medical systems, including surgical robots, is not to replace the physician with a machine but, rather, to provide intelligent, versatile tools that augment the physician's ability to treat patients. There are many forms of technological assistance. In this section, we focus on robotic assistance. Some nonrobotic technologies are reviewed in the "Other Technologies" section.

There are two basic augmentation strategies: 1) improving the physician's existing sensing and/or manipulation, and 2) increasing the number of sensors and manipulators available to the physician (e.g., more eyes and hands). In the first case, the system can give even average physicians superhuman capabilities such as X-ray vision, elimination of hand tremor, or the ability to perform dexterous operations inside the patient's body. A special subclass is remote telesurgery systems, which permit the physician to operate on patients at distances ranging from a few meters to several thousand kilometers. In the second case, the robot operates side by side with the physician and performs functions such as endoscope holding, tissue retraction, or limb positioning. These systems typically provide one or more direct control interfaces such as joysticks, head trackers, or voice control but could also include intelligence to demand less of the physician's attention during use.

The da Vinci system (Intuitive Surgical, Inc.) is a telesurgery system that demonstrates both of these augmentation approaches [2]. As shown in Figure 3, the system consists of a patient-side slave robot and a master control console. The slave robot has three or four robotic arms that manipulate a stereo endoscope and dexterous surgical instruments such as scissors, grippers, and needle holders. The surgeon sits at the master control console and grasps handles attached to two dexterous master manipulator arms, which are capable of exerting limited

amounts of force feedback to the surgeon. The surgeon's hand motions are sensed by the master manipulators, and these motions are replicated by the slave manipulators. A variety of control modes may be selected via foot pedals on the master console and used for such purposes as determining which slave arms are associated with the hand controllers. Stereo video is transmitted from the endoscope to a pair of high-quality video monitors in the master control console, thus providing high-fidelity stereo visualization of the surgical site. The display and master manipulators are arranged so that it appears to the surgeon that the surgical instruments (inside the patient) are in the same position as his or her hands inside the master control console. Thus, the da Vinci system improves the surgeon's eyes and hands by enabling them to (remotely) see and manipulate tissue inside the patient through incisions that are too small for direct visualization and manipulation. By providing three or four slave robot arms, the da Vinci system also endows the surgeon with more than two hands.

### Other Technologies

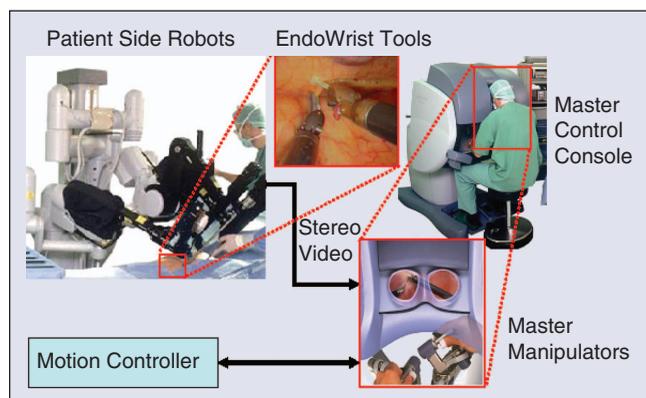
Robotics is not the only manner in which computers can be used to assist medical procedures. One important, and widely used, alternative is a navigation system, which consists of a sensor (tracker) that can measure the position and orientation of instruments in 3-D space (typically, the instruments contain special tracker targets). If the tracker coordinate system is registered to a preoperative or intraoperative image (see the "Registration" section), the navigation system can display the position and orientation of the instrument with respect to the image. This improves the physician's visualization by enabling him or her to see the internal structure, molecular information, and/or functional data, depending on the type of image. This can also enable the physician to execute a preoperative plan (surgical CAD/CAM), e.g., by aligning an instrument with respect to a target defined in the preoperative image. Currently, the most widely used tracking technology is optical because of its relatively high accuracy, predictable performance, and insensitivity to environmental variations. The primary limitation of optical trackers is that they require a clear line of sight between the camera and the instruments being tracked. This precludes their use for instruments inside the body. Electromagnetic tracking systems are free from line-of-sight constraints but are generally less accurate, especially due to field distortions caused by metallic objects.

### Technology and Challenges

Surgical robots present a unique set of design challenges due to the requirements for miniaturization, safety, sterility, and adaptation to changing conditions. This section reviews current practices and challenges in manipulation, sensing, registration, user interfaces, and system design.

### Manipulation

Surgical robots must satisfy requirements not found in industrial robotics. They must operate safely in a workspace shared with humans; they generally must operate in a sterile environment; and they often require high dexterity in small spaces. An



**Figure 3.** The da Vinci surgical system (courtesy Intuitive Surgical, Inc.).

additional challenge occurs when the robot must operate in the proximity of a magnetic resonance imaging (MRI) scanner, whose high magnetic field precludes the use of many conventional robotic components.

The topic of safety design is covered in detail in the “Safety Design” section. There are, however, certain safety factors that should be considered during the design of a surgical manipulator. First, unlike industrial robots, where speed and strength are desirable attributes, a surgical robot should only be as fast and strong as needed for its intended use. In most cases, the robot should not be capable of moving faster or with more force than the physician. An obvious exception could occur for a robot that operates on a rapidly moving organ, such as a beating heart. Even in this case, there are innovative solutions that do not require rapid motion, such as Heartlander [6], which is designed to attach to a beating heart using suction and move along it with inchworm locomotion. Another safety-related design parameter is the robot’s workspace, which ideally should only be as large as needed. This is difficult to achieve in practice, given the high variability between patients and the differences in the way that physicians perform procedures. Some researchers have reported parallel manipulators, which have smaller workspaces (and higher rigidity) than serial robots [7]–[10].

Sterility is a major design challenge. It is not easy to design reusable devices that can withstand multiple sterilization cycles. One common solution is to create a disposable device that only needs to be sterilized once, usually by the manufacturer. This is practical for low-cost parts. Another issue with a reusable device is that it must be cleaned between procedures. Thus, crevices that can trap blood or other debris should be avoided. The most common approach is to design the surgical robot so that its end effector (or tool) can be removed and sterilized, while the rest of the robot is covered with a disposable sterile drape or bag (e.g., as illustrated for ROBODOC in Figure 2). This is particularly difficult when the end effector or tool includes electromechanical components.

Size matters for surgical robots. Operating rooms and interventional suites are usually small, and, thus, a large robot can take too much space. This has been a complaint for many commercially available systems, such as daVinci and ROBODOC, which are large floor-standing robots. In orthopedics, there have been recent examples of smaller, bone-mounted robots [7]–[9]. Size is especially critical when the robot, or part of it, must work inside the body. For example, although the da Vinci system is large, its robotic EndoWrist tools, with diameters from 5–8 mm, are a marvel of miniaturization and can pass into the body via small entry ports.

The design of MRI-compatible robots is especially challenging because MRI relies on a strong magnetic field and radio frequency (RF) pulses, and so it is not possible to use components that can interfere with, or be susceptible to, these physical effects. This rules out most components used for robots, such as electric motors and ferromagnetic materials. Thus, MRI-compatible robots typically use nonmetallic links and piezoelectric, pneumatic, or hydraulic motors. This topic will be discussed in greater detail in a subsequent part of this tutorial.

## Sensing

Besides internal sensors, such as joint encoders, a surgical robot often needs external sensors to enable it to adapt to its relatively unstructured and changing environment. Common examples are force sensors and vision systems, which translate naturally into the human senses of touch and sight. For this reason, they are often used for surgical assistants. For example, the da Vinci system provides exquisite stereo video feedback, although it is often criticized for not providing force feedback (a component of haptic feedback). Without force feedback, the surgeon must use visual cues, such as the tautness of a suture or the deflection of tissue, to estimate the forces. If these cues are misread, the likely outcome is a broken suture or damaged tissue [11].

Real-time imaging such as ultrasound, spectroscopy, and optical coherence tomography (OCT) can provide significant benefits when they enable the physician to see subsurface structures and/or tissue properties. For example, when resecting a brain tumor, this type of sensing can alert the surgeon before he or she accidentally cuts a major vessel that is obscured by the tumor. Preoperative images, when registered to the robot, can potentially provide this information, but only if the anatomy does not change significantly during the procedure. This is rarely the case, except when working with rigid structures such as bones. Once again, it is necessary to overcome challenges in sterility and miniaturization to provide this sensing where it is needed, which is usually at or near the instrument tip.

Sensors that directly measure physiologic properties, such as tissue oxygenation, are also useful. For example, a smart retractor that uses pulse oxymetry principles to measure the oxygenation of blood can detect the onset of ischemia (insufficient blood flow) before it causes a clinical complication [12].

## Registration

Geometric relationships between portions of the patient’s anatomy, images, robots, sensors, and equipment are fundamental to all areas of computer-integrated medicine. There is an extensive literature on techniques for determining the transformations between the associated coordinate systems [13], [14]. Given two coordinates  $\vec{v}_A = [x_A, y_A, z_A]$  and  $\vec{v}_B = [x_B, y_B, z_B]$  corresponding to comparable features in two coordinate systems  $\text{Ref}_A$  and  $\text{Ref}_B$ , the process of registration is simply that of finding a function  $\mathbf{T}_{AB}(\dots)$  such that

$$\vec{v}_B = \mathbf{T}_{AB}(\vec{v}_A).$$

Although nonrigid registrations are becoming more common,  $\mathbf{T}_{AB}(\dots)$  is still usually a rigid body transformation of the form

$$\vec{v}_B = \mathbf{T}_{AB}(\vec{v}_A) = \mathbf{R}_{AB} \bullet \vec{v}_A + \vec{p}_{AB},$$

where  $\mathbf{R}_{AB}$  represents a rotation and  $\vec{p}_{AB}$  represents a translation.

The simplest registration method is a paired-point registration in which a set of  $N$  points ( $N \geq 3$ ) is found in the first coordinate system and matched (one to one) with  $N$

## *The development of surgical robots is motivated primarily by the desire to enhance the effectiveness of a procedure.*

corresponding points in the second coordinate system. The problem of finding the transformation that best aligns the two sets of points is often called the Procrustes problem, and there are well-known solutions based on quaternions [15] and rotation matrices [16], [17]. This method works best when it is possible to identify distinct points in the image and on the patient. This is usually straightforward when artificial fiducials are used. For example, ROBODOC initially used a fiducial-based registration method, with three metal pins (screws) inserted into the bone prior to the CT scan. It was easy to locate the pins in the CT image, via image processing, due to the high contrast between metal and bone. Similarly, it was straightforward for the surgeon to guide the robot's measurement probe to physically contact each of the pins.

Point-to-surface registration methods can be employed when paired-point registration is not feasible. Typically, this involves matching a cloud of points that is collected intraoperatively to a 3-D surface model that is constructed from the pre-operative image. The most widely used method is iterative closest point (ICP) [18]. Briefly, ICP starts with an initial guess of the transformation, which is used to transform the points to the same coordinate system as the surface model. The closest points on the surface model are identified and a paired-point registration method is used to compute a new estimate of the transformation. The process is repeated with the new transformation until a termination condition is reached. Although ICP often works well, it is sensitive to the initial guess and can fail to find the best solution if the guess is poor. Several ICP variations have been proposed to improve its robustness in this case, and other techniques, such as an unscented Kalman filter [19], have recently been proposed. These methods can also be used for surface-to-surface registration by sampling one of the surfaces.

Nonrigid (elastic or deformable) registration is often necessary because many parts of the anatomy (e.g., soft tissue and organs) change shape during the procedure. This is more difficult than rigid registration and remains an active area of research. To date, most surgical CAD/CAM systems have been applied to areas such as orthopedics, where deformations are small and rigid registration methods can be employed.

### **User Interfaces and Visualization**

Standard computer input devices, such as keyboards and mice, are generally inappropriate for surgical or interventional environments because it is difficult to use them in conjunction with other medical instrumentation and maintain sterility. Foot pedals are often used because they do not interfere with whatever the physician is doing with his or her hands, and they

do not require sterilization. Handheld pendants (button boxes) are also used; in this case, the pendant is either sterilized or covered by a sterile drape. It is important to note, however, that the robot itself can often provide a significant part of the user interface. For example, the da Vinci system relies on the two master manipulators (one for each hand), with foot pedals to change modes. The ROBODOC system not only includes a five-button pendant to navigate menus but also uses a force-control (hand guiding) mode that enables the surgeon to manually move the robot.

Computer output is traditionally provided by graphical displays. Fortunately, these can be located outside the sterile field. Unfortunately, the ergonomics are often poor because the physician must look away from the operative site (where his or her hands are manipulating the instruments) to see the computer display. Some proposed solutions include heads-up displays, image overlay systems [20], [21], and lasers, which project information onto the operative field [22].

### **Surgical Robot System Design**

A surgical robot includes many components, and it is difficult to design one from scratch. There is no off-the-shelf surgical robot for research, and it is unlikely that one robot or family of robots will ever satisfy the requirements of the diaspora of medical procedures. In the software realm, however, there are open source software packages that can help. The most mature packages are for medical image visualization and processing, particularly the Visualization Toolkit (VTK, [www.vtk.org](http://www.vtk.org)) and the Insight Toolkit (ITK, [www.itk.org](http://www.itk.org)). Customizable applications, such as 3-D Slicer ([www.slicer.org](http://www.slicer.org)), package VTK, ITK, and a plethora of research modules.

Few packages exist for computer-assisted interventions. The Image Guided Surgery Toolkit (IGSTK, [www.igstk.org](http://www.igstk.org)) enables researchers to create a navigation system by connecting a tracking system to a computer. At Johns Hopkins University, we are creating a software framework for a surgical assistant workstation (SAW), based on our Computer-Integrated Surgical Systems and Technology (CISST) libraries [23] ([www.cisst.org](http://www.cisst.org)), which focus on the integration of robot control and real-time sensing with the image processing and visualization toolkits described previously.

### **Surgical Robot Design Process**

This section presents a detailed discussion of the risk analysis, safety design, and validation phases of the design process. Although these topics are not unique to surgical robots, they are obviously of extreme importance.

### **Risk Analysis**

Safety is an important consideration for both industrial and surgical robots [24]. In an industrial setting, safety can often be achieved by keeping people out of the robot's workspace or by shutting down the system if a person comes too close. In contrast, for surgical robots it is generally necessary for human beings, including the patient and the medical staff, to be inside the robot's workspace. Furthermore, the robot may be holding a potentially dangerous device, such as a cutting instrument,

that is supposed to actually contact the patient (in the correct place, of course). If the patient is anesthetized, it is not possible for him or her to actively avoid injury.

Proper safety design begins with a risk (or hazard) analysis. A failure modes effects analysis (FMEA) or failure modes effects and criticality analysis (FMECA) are the most common methods [25]. These are bottom-up analyses, where potential component failures are identified and traced to determine their effect on the system. Methods of control are devised to mitigate the hazards associated with these failures. The information is generally presented in a tabular format (see Table 1). The FMECA adds the criticality assessment, which consists of three numerical parameters: the severity (S), occurrence (O), and detectability (D) of the failure. A risk priority number (RPN) is computed from the product of these parameters, which determines whether additional methods of control are required. The FMEA/FMECA is a proactive analysis that should begin early in the design phase and evolve as hazards are identified and methods of control are developed. Another popular method is a fault tree analysis (FTA), which is a top-down analysis and is generally more appropriate for analyzing a system failure after the fact.

### Safety Design

As an illustrative example of how to apply these methods in the design phase, consider a multilink robot system where each link is driven by a feedback-controlled motor, as shown in Figure 4. The error,  $e(t)$ , between the desired position  $x_d(t)$  and the measured position  $x_a(t)$  is computed and used to determine the control output  $u(t)$  that drives the motor. An encoder failure will cause the system to measure a persistent steady-state error and therefore continue to drive the motor to attempt to reduce this error. An amplifier failure can cause it to apply an arbitrary voltage to the motor that is independent of the control signal  $u(t)$ . The controller will sense the increasing error and adjust  $u(t)$  to attempt to compensate, but this will have no effect.

These failure modes are shown in the FMEA presented in Table 1. The result in both these cases is that the robot will move until it hits something (typically, the effect on system is more descriptive and includes application-specific information, such as the potential harm to the patient). This is clearly unacceptable for a surgical robot, and so methods of control are necessary. One obvious solution, shown in Table 1, is to allow the control software to disable the motor power, via a relay, whenever the error,  $e(t)$ , exceeds a specified threshold. This will prevent a catastrophic, headline-grabbing runaway robot scenario, but is the robot safe enough for surgical use? The answer is that it depends on the application and on the physical parameters of the system. To illustrate this, consider the case where the power amplifier fails and applies maximum voltage to the motor. As shown in Figure 5, if  $E$  is the error threshold (i.e., the point at which the control software disables motor power via the relay), the final joint position error,  $\Delta P_{\max}$ , is given by

$E + V_{\max} \times \Delta T + \Delta P_{\text{off}}$ , where  $\Delta T$  is the control period,  $V_{\max}$  is the maximum joint velocity (assuming the robot had sufficient time to accelerate), and  $\Delta P_{\text{off}}$  is the distance the robot travels after power off due to inertia or external forces. The actual value of  $\Delta P_{\max}$  depends on the robot design, but it is not uncommon for this to be several millimeters. Although a one-time glitch of this magnitude may be tolerable for some surgical procedures, it is clearly not acceptable in others. In those cases, it is necessary to make design modifications to decrease  $\Delta P_{\max}$ , e.g., by decreasing  $V_{\max}$ , or to forgo the use of an active robot. This safety analysis was a prime motivation for researchers who developed passive robots such as Cobots [26] and PADyC [27].

There are safety issues that must be considered regardless of whether a robot is active or passive. One example occurs when the robot's task is to accurately position an instrument

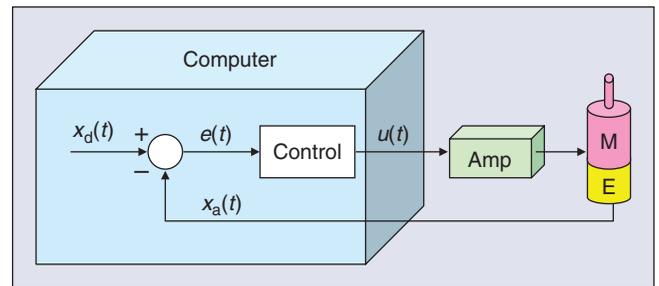


Figure 4. Computer control of a robot joint, showing the motor (M), encoder (E), and power amplifier (Amp).

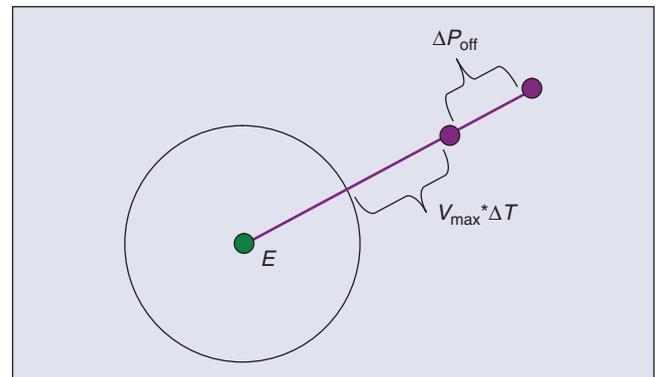


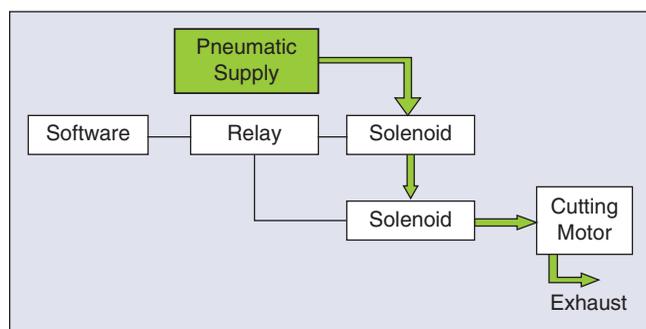
Figure 5. Illustration of maximum possible error:  $E$  is the error threshold,  $V_{\max}$  is the maximum velocity,  $\Delta T$  is the control period, and  $\Delta P_{\text{off}}$  is the robot stopping distance.

Table 1. Excerpt from a sample FMEA.			
Failure Mode	Effect on System	Causes	Methods of Control*
Robot out of control	Robot may hit something	Encoder failure, broken wire	Trip relay when error tolerance exceeded
Robot out of control	Robot may hit something	Amplifier failure	Trip relay when error tolerance exceeded
*Methods of control can initially be empty and then populated during the design phase.			

## Surgical robots present a unique set of design challenges due to the requirements for miniaturization, safety, sterility, and adaptation to changing conditions.

or instrument guide. The position of a robot-held tool is typically determined by applying the robot's forward kinematic equations to the measured joint positions. An inaccurate joint sensor (e.g., an incremental encoder that intermittently gains or loses counts) can cause a large position error. One method of control is to introduce a redundant sensor and use software to verify whether both sensors agree within a specified tolerance. Practical considerations dictate the need for a tolerance to account for factors such as mechanical compliance between the sensors and differences in sensor resolution and time of data acquisition. This limits the degree with which accuracy can be assured. Note also that although redundant sensors remove one single point of failure (i.e., sensor failure), it is necessary to avoid a single point of failure in the implementation. For example, if both sensors are placed on the motor shaft, they cannot account for errors in the joint transmission, e.g., due to a slipped belt.

A final point is that redundancy is not sufficient if failure of one component cannot be detected. For example, consider the case where the robot is holding a pneumatic cutting tool, and a solenoid is used to turn the tool on and off. If the solenoid fails in the open (on) state, the cutting tool may be activated at an unsafe time. It is tempting to address this hazard by putting a second solenoid in series with the first, as shown in Figure 6. This is not an acceptable solution, however, because if one solenoid fails in the open state, the system will appear to operate correctly (i.e., the software can still turn the cutter on or off). Therefore, this system once again has a single point of failure. This is not a hypothetical scenario—it actually appeared in the risk analysis for the ROBODOC system, which uses a pneumatic cutting tool. The concern was that a failed solenoid could cause



**Figure 6.** Example of poorly designed redundant system. The second solenoid does not provide sufficient safety because the system cannot detect when either solenoid has failed in the open state.

injury to the surgeon if the failure occurred while the surgeon was inserting or removing the cutting bit. ROBODOC adopted a simple method of control, which was to display a screen instructing the surgeon to physically disconnect the pneumatic supply prior to any cutting tool change.

### Validation

Validation of computer-integrated systems is challenging because the key measure is how well the system performs in an operating room or interventional suite with a real patient. Clearly, for both ethical and regulatory reasons, it is not possible to defer validation until a system is used with patients. Furthermore, it is difficult to quantify intraoperative performance because there are limited opportunities for accurate postoperative assessment. For example, CT scans may not provide sufficient contrast for measuring the postoperative result, and they expose the patient to additional radiation. For these reasons, most computer-integrated systems are validated using phantoms, which are objects that are designed to mimic (often very crudely) the relevant features of the patient.

One of the key drivers of surgical CAD/CAM is the higher level of accuracy that can be achieved using some combination of computers, sensors, and robots. Therefore, it is critical to evaluate the overall accuracy of such a system. One common technique is to create a phantom with a number of features (e.g., fiducials) whose locations are accurately known, either by precise manufacturing or measurement. Some of these features should be used for registration, whereas others should correspond to targets. The basic technique is to image the phantom, perform the registration, and then locate the target features. By convention, the following types of error are defined [28] as follows:

- ◆ *fiducial localization error (FLE)*: the error in locating a fiducial in a particular coordinate system (i.e., imaging system or robot system)
- ◆ *fiducial registration error (FRE)*: the root mean square (RMS) residual error at the registration fiducials, i.e.,

$$\text{FRE} = \sqrt{\frac{1}{N} \sum_{k=1}^N \|\vec{\mathbf{b}}_k - \mathbf{T} \cdot \vec{\mathbf{a}}_k\|^2}$$

where  $\mathbf{T}$  is the registration transform and  $(\vec{\mathbf{a}}_k, \vec{\mathbf{b}}_k)$  are matched pairs of homologous fiducials ( $k = 1, \dots, N$ )

- ◆ *target registration error (TRE)*: the error in locating a feature or fiducial that was not used for the registration; if multiple targets are available, the mean error is often reported as the TRE.

Although it is necessary to validate that a surgical robot meets its requirements, including those related to accuracy, it is important to realize that higher accuracy may not lead to a clinical benefit. Validation of clinical utility is often possible only via clinical trials.

### Summary

This article presents the first of a three-part tutorial on surgical and interventional robotics. The core concept is that a surgical robot couples information to action in the operating room or

interventional suite. This leads to several potential benefits, including increased accuracy and the ability to intervene in areas that are not accessible with conventional instrumentation. We defined the categories of surgical CAD/CAM and surgical assistance. The former is intended to accurately execute a defined plan. The latter is focused on providing augmented capabilities to the physician, such as superhuman or auxiliary (additional) eyes and hands. These categories will be the focus of the final two parts of this tutorial.

There are numerous challenges in surgical manipulation, sensing, registration, user interfaces, and system design. Many of these challenges result from the requirements for safety, sterility, small size, and adaptation to a relatively unstructured (and changing) environment. Some software toolkits are available to facilitate the design of surgical robotics systems.

The design of a surgical robot should include a risk analysis. Established methodologies such as FMEA/FMECA can be used to identify potential hazards. Safety design should consider and eliminate single points of failure whenever possible. Validation of system performance is critical but is complicated by the difficulty of simulating realistic clinical conditions.

Surgical robotics is a challenging field, but it is rewarding because the ultimate goal is to improve the health and quality of human life.

### Acknowledgments

The authors gratefully acknowledge the National Science Foundation for supporting our work in this field through the Engineering Research Center for Computer-Integrated Surgical Systems and Technology (CISST ERC), NSF Grant EEC 9731748. Related projects have also been supported by Johns Hopkins University, the National Institutes of Health, the Whitaker Foundation, the Department of Defense, and our CISST ERC industrial affiliates.

### Keywords

Surgical robots, medical robots, robot safety.

### References

- [1] Y. S. Kwok, J. Hou, E. A. Jonckheere, and S. Hayati, "A robot with improved absolute positioning accuracy for CT-guided stereotactic brain surgery," *IEEE Trans. Biomed. Eng.*, vol. 35, no. 2, pp. 153–160, 1988.
- [2] G. S. Guthart and J. K. Salisbury, "The intuitive telesurgery system: Overview and application," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA 2000)*, San Francisco, vol. 1, pp. 618–621.
- [3] R. H. Taylor and D. Stoianovici, "Medical robotics in computer-integrated surgery," *IEEE Trans. Robot. Automat.*, vol. 19, no. 3, pp. 765–781, 2003.
- [4] R. H. Taylor, B. D. Mittelstadt, H. A. Paul, W. Hanson, P. Kazanzides, J. F. Zuhars, B. Williamson, B. L. Musits, E. Glassman, and W. L. Bargar, "An image-directed robotic system for precise orthopaedic surgery," *IEEE Trans. Robot. Automat.*, vol. 10, no. 3, Jun. 1994.
- [5] P. Kazanzides, "Robot assisted surgery: The ROBODOC experience," in *Proc. 30th Int. Symp. Robotics (ISR)*, Tokyo, Japan, Nov. 1999, pp. 261–286.
- [6] N. Patronik, C. Riviere, S. E. Qarra, and M. A. Zenati, "The Heart-Lander: A novel epicardial crawling robot for myocardial injections," in *Proc. 19th Int. Congr. Computer Assisted Radiology and Surgery*, 2005, vol. 1281C, pp. 735–739.

- [7] M. Shoham, M. Burman, E. Zehavi, L. Joskowicz, E. Batkalin, and Y. Kunicher, "Bone-mounted miniature robot for surgical procedures: Concept and clinical applications," *IEEE Trans. Robot. Automat.*, vol. 19, no. 5, pp. 893–901, Oct. 2003.
- [8] A. Wolf, B. Jaramaz, B. Lisien, and A. M. DiGioia, "MBARS: Mini bone-attached robotic system for joint arthroplasty," *Int. J. Med. Robot. Comp. Assist. Surg.*, vol. 1, no. 2, pp. 101–121, Jan. 2005.
- [9] J. H. Chung, S. Y. Ko, D. S. Kwon, J. J. Lee, Y. S. Yoon, and C. H. Won, "Robot-assisted femoral stem implantation using an intramedulla gauge," *IEEE Trans. Robot. Automat.*, vol. 19, no. 5, pp. 885–892, Oct. 2003.
- [10] G. Brandt, A. Zimolong, L. Carrat, P. Merloz, H. W. Staudte, S. Lavallee, K. Radermacher, and G. Rau, "CRIGOS: A compact robot for image-guided orthopedic surgery," *IEEE Trans. Inform. Technol. Biomed.*, vol. 3, no. 4, pp. 252–260, Dec. 1999.
- [11] A. M. Okamura, "Methods for haptic feedback in teleoperated robot-assisted surgery," *Ind. Robot*, vol. 31, no. 6, pp. 499–508, 2004.
- [12] G. Fischer, T. Akinbiyi, S. Saha, J. Zand, M. Talamini, M. Marohn, and R. H. Taylor, "Ischemia and force sensing surgical instruments for augmenting available surgeon information," in *Proc. IEEE Int. Conf. Biomedical Robotics and Biomechanics (BioRob 2006)*, Pisa, Italy, 2006, pp. 1030–1035.
- [13] J. B. Maintz and M. A. Viergever, "A survey of medical image registration," *Med. Image Anal.*, vol. 2, no. 1, pp. 1–37, 1998.
- [14] S. Lavallee, "Registration for computer-integrated surgery: methodology, state of the art," in *Computer-Integrated Surgery*, R. H. Taylor, S. Lavallee, G. Burdea, and R. Mosges, Eds. Cambridge, MA: MIT Press, 1996, pp. 77–98.
- [15] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Am. A*, vol. 4, no. 4, pp. 629–642, 1987.
- [16] K. Arun, T. Huang, and S. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no. 5, pp. 698–700, 1987.
- [17] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 4, pp. 376–380, 1991.
- [18] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 2, pp. 239–256, 1992.
- [19] M. H. Moghari and P. Abolmaesumi, "Point-based rigid-body registration using an unscented Kalman filter," *IEEE Trans. Med. Imag.*, vol. 26, no. 12, pp. 1708–1728, Dec. 2007.
- [20] M. Blackwell, C. Nikou, A. M. DiGioia, and T. Kanade, "An image overlay system for medical data visualization," *Med. Image Anal.*, vol. 4, no. 1, pp. 67–72, 2000.
- [21] G. Fichtinger, A. Deguet, K. Masamune, E. Balogh, G. S. Fischer, H. Mathieu, R. H. Taylor, S. J. Zinreich, and L. M. Fayad, "Image overlay guidance for needle insertion on CT scanner," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 8, pp. 1415–1424, Aug. 2005.
- [22] T. Sasama, N. Sugano, Y. Sato, Y. Momoi, T. Koyama, Y. Nakajima, I. Sakuma, M. G. Fujie, K. Yonenobu, T. Ochi, and S. Tamura, "A novel laser guidance system for alignment of linear surgical tools: Its principles and performance evaluation as a man-machine system," in *Proc. 5th Int. Conf. Medical Image Computing and Computer-Assisted Intervention*, 2002, vol. 2489, pp. 125–132.
- [23] A. Kapoor, A. Deguet, and P. Kazanzides, "Software components and frameworks for medical robot control," in *Proc. IEEE Conf. Robotics and Automation (ICRA)*, Orlando, FL, May 2006, pp. 3813–3818.
- [24] B. Davies, "A discussion of safety issues for medical robots," in *Computer-Integrated Surgery*, R. Taylor, S. Lavallee, G. Burdea, and R. Moesges, Eds. Cambridge, MA: MIT Press, 1996, pp. 287–296.
- [25] R. E. McDermott, R. J. Mikulak, and M. R. Beauregard, "The Basics of FMEA," New York, Quality Resources, 1996.

- [26] M. A. Peshkin, J. E. Colgate, W. Wannasuphprasit, C. A. Moore, R. B. Gillespie, and P. Akella, "Cobot architecture," *IEEE Trans. Robot. Automat.*, vol. 17, no. 4, pp. 377–390, Aug. 2001.
- [27] O. Schneider and J. Troccaz, "A six-degree-of-freedom passive arm with dynamic constraints (PADyC) for cardiac surgery applications: Preliminary experiments," *Comput. Aided Surg.*, vol. 6, no. 6, pp. 340–351, 2001.
- [28] C. Maurer, J. Fitzpatrick, M. Wang, R. Galloway, R. Maciunas, and G. Allen, "Registration of head volume images using implantable fiducial markers," *IEEE Trans. Med. Imag.*, vol. 16, no. 4, pp. 447–462, Aug. 1997.

**Peter Kazanzides** received the B.S., M.S., and Ph.D. degrees in electrical engineering from Brown University in 1983, 1985, and 1988, respectively. He worked on surgical robotics in March 1989 as a postdoctoral researcher at the International Business Machines (IBM) T.J. Watson Research Center. He cofounded Integrated Surgical Systems (ISS) in November 1990 to commercialize the robotic hip replacement research performed at IBM and the University of California, Davis. As the director of robotics and software, he was responsible for the design, implementation, validation and support of the ROBODOC System. He joined the Engineering Research Center for Computer-Integrated Surgical Systems and Technology (CISST ERC) in December 2002, and currently, he is an assistant research professor of computer science at Johns Hopkins University.

**Gabor Fichtinger** received his B.S. and M.S. degrees in electrical engineering and his Ph.D. degree in computer science from the Technical University of Budapest, Hungary, in 1986, 1988, and 1990, respectively. He has developed image-guided surgical interventional systems. He specializes in robot-assisted image-guided needle-placement procedures, primarily for cancer diagnosis and therapy. He is an associate professor of computer science, electrical engineering, mechanical engineering, and surgery at Queen's University, Canada, with adjunct appointments at the Johns Hopkins University.

**Gregory D. Hager** is a professor of computer science at Johns Hopkins University. He received the B.A. degree, *summa cum laude*, in computer science and mathematics from Luther College, in 1983, and the M.S. and Ph.D. degrees in computer science from the University of Pennsylvania in 1985 and 1988, respectively. From 1988 to 1990, he was a Fulbright junior research fellow at the University of Karlsruhe and the Fraunhofer Institute IITB in Karlsruhe, Germany. From 1991 to 1999, he was with the Computer Science Department at Yale University. In 1999, he joined the Computer Science Department at Johns Hopkins University, where he is the deputy director of the Center for Computer Integrated Surgical Systems and Technology. He has authored more than 180 research articles and books in the area of robotics and computer vision. His current

research interests include visual tracking, vision-based control, medical robotics, and human-computer interaction. He is a Fellow of the IEEE.

**Allison M. Okamura** received the B.S. degree from the University of California at Berkeley, in 1994, and the M.S. and Ph.D. degrees from Stanford University in 1996 and 2000, respectively, all in mechanical engineering. She is currently an associate professor of mechanical engineering and the Decker Faculty Scholar at Johns Hopkins University. She is the associate director of the Laboratory for Computational Sensing and Robotics and a thrust leader of the National Science Foundation Engineering Research Center for Computer-Integrated Surgical Systems and Technology. Her awards include the 2005 IEEE Robotics Automation Society Early Academic Career Award, the 2004 National Science Foundation Career Award, the 2004 Johns Hopkins University George E. Owen Teaching Award, and the 2003 Johns Hopkins University Diversity Recognition Award. Her research interests include haptics, teleoperation, medical robotics, virtual environments and simulators, prosthetics, rehabilitation engineering, and engineering education.

**Louis L. Whitcomb** completed his B.S. and Ph.D. degrees at Yale University in 1984 and 1992, respectively. His research focuses on the design, dynamics, navigation, and control of robot systems. He has numerous patents in the field of robotics, and he is a Senior Member of the IEEE. He is the founding director of the Johns Hopkins University Laboratory for Computational Sensing and Robotics. He is a professor at the Department of Mechanical Engineering, with joint appointment in the Department of Computer Science, at the Johns Hopkins University.

**Russell H. Taylor** received his Ph.D. degree in computer science from Stanford in 1976. He joined IBM Research in 1976, where he developed the AML robot language and managed the Automation Technology Department and (later) the Computer-Assisted Surgery Group before moving in 1995 to Johns Hopkins University, where he is a professor of computer science, with joint appointments in mechanical engineering, radiology and surgery. He is the Director of the NSF Engineering Research Center for Computer-Integrated Surgical Systems and Technology. He is the author of more than 200 refereed publications. He is a Fellow of the IEEE and AIMB and is a recipient of the Maurice Müller award for excellence in computer-assisted orthopedic surgery.

**Address for Correspondence:** Peter Kazanzides, Department of Computer Science, CSEB 120, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA. E-mail: pkaz@jhu.edu.

## Lethality and Autonomous Systems: Survey Design and Results

According to a recent report by Moshkina and Arkin of the Georgia Institute of Technology, already there are armed robot systems deployed or being deployed in Afghanistan and Iraq [1], [2], the Israeli-Palestinian Border [3], and the Korean Demilitarized Zone [4]. According to them, there is also a likelihood of an increasing role of autonomy for these battlefield robots as humans are gradually moved further and further out of the loop [5], [6].

When we consider the use of deadly force by robots, it is crucial that we ask the question, "What is acceptable?" As a part of an effort to "understand, define, and shape expectations regarding battlefield robotics?" Moshkina and Arkin conducted a survey of researchers, policymakers, military personnel, and the general public to ascertain the current point of view maintained by various demographic groups on this subject.

As part of this study, the team conducted an online public opinion survey on the use of robots capable of lethal force in warfare. The main objective of the survey was to determine the level of acceptance by various demographic groups of the employment of potentially lethal robots in warfare as well as their attitudes toward related ethical issues.

The questions dealt with the use of human soldiers, robots as extensions of the human soldier, and autonomous robots in a variety of roles and situations. The researchers also compared the attitudes of the different community types, i.e., those who identified themselves as having had an experience in the military, as policymakers, or as robotics professionals. Those with none of these experiences were classified as general public.

The authors summarized the survey responses as follows:

- ◆ As far as the community types are concerned, regardless of roles or situations, in most cases, the general public found employment of soldiers and robots less acceptable than any other community type, and, conversely, military and policymakers found such employment more acceptable.
- ◆ The most acceptable role for using both types of robots in is reconnaissance; the least acceptable is for crowd control.
- ◆ With respect to levels of autonomy, regardless of roles or situations, the more the control shifts away from the human, the less such an entity is acceptable to the participants; a human soldier was the most acceptable entity in warfare, followed by a robot as an extension of the warfighter, with autonomous robot being the least acceptable.
- ◆ As far as the situations are concerned, covert operations were less acceptable to the entire set of participants

than open warfare for all three entities: soldiers and both types of robots (whether on home or foreign territory).

- ◆ The majority of participants, regardless of the community type, agreed that the ethical standards, namely, Laws of War, Rules of Engagement, Code of Conduct, and Additional Moral Standards, do apply to both soldiers (84%) and robots (72%).
- ◆ More military and policymakers were in favor of the same standards for both soldiers and robots than general public and roboticists, who were more in favor of higher standards for robots.
- ◆ 59% of the participants believed that an autonomous robot should have a right to refuse an order it finds unethical, thus in a sense admitting that it may be more important for a robot to behave ethically than to stay under the control of a human.
- ◆ As the control shifts away from the soldier, the robot and its maker should take more responsibility for its actions, according to the participants. A robot designer was blamed 31% less for the mistakes of robot as an extension than for those of an autonomous robot.
- ◆ "Saving lives of soldiers" was considered the most clear-cut benefit of employing robots in warfare; the main concern was that of risking civilian lives by their use.
- ◆ The majority of the participants (69%) believe that it would be easier to start wars if robots were employed in warfare.
- ◆ Sympathy was considered to be beneficial to a military robot by over half of the participants (59%) and guilt by just under a half (49%).

Although, in general, the difference in opinions between the community types was slight, it is interesting to note that the general public and roboticists were less likely to identify "saving civilian lives" as a benefit than politicians or military, and fewer roboticists believed that robots could help produce better battlefield outcomes.

Since resource constraints prevented the authors from mailing surveys to a randomly distributed population, participants were recruited using a variety of means and venues, most of them were online based. Of the 430 participants who completed the survey, 234 (54%) self-identified themselves as having had robotics research experience, 69 having had policymaking experience, 127 having had military experience, and 116 having had none of the aforementioned (therefore categorized as general public). Participants were primarily male (89%) and ranged in age from 18 to older than 66, with 43% between 21 and 30 and 22% between 31 and 40; most were well educated (only 4% had no post-secondary education) and international (45% were raised outside the United States).

2008

**20–23 June** ICIA 2008: IEEE International Conference on Information Automation. Zhangjiajie, Hunan, China. <http://www.ieee-icia.info>

**25–28 June** RSS 2008: Robotics: Science and Systems. Zurich, Switzerland. <http://www.roboticsconference.org>

**2–5 July** AIM 2008: IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Xi'an, China. <http://www.aim2008.info>

**1–3 Aug.** RO-MAN 2008: IEEE International Symposium on Robot and Human Interaction. Munich, Germany. <http://www.ro-man2008.org>

**5–8 Aug.** ICMA 2008: IEEE International Conference on Mechatronics and Automation. Takamatsu, Japan. <http://www.icma2008.org/>

**20–22 Aug.** MFI 2008: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. Seoul, Korea. <http://www.mfi2008.org/>

**22 Aug.** SICE 2008: Annual Conference. Chofu, Japan. <http://www.sice.or.jp/sice2008/>

**23–26 Aug.** IEEE-CASE 2008: 4th International Conference on Automation Science and Engineering. Washington, District of Columbia, USA. <http://www.ieee-case.org>

**1–3 Sept.** ICAL 2008: IEEE International Conference on Automation and Logistics. Qingdao, China. <http://myweb.dal.ca/jgu/ical08/>

**16–17 Sept.** IEEE-CYBER: IEEE International Conference on Automation, Control, and Intelligent Systems. Shenyang, China. (Contact: E-mail: [tarn@wuauto.wustl.edu](mailto:tarn@wuauto.wustl.edu))

**22–26 Sept.** IROS 2008: IEEE/RSJ International Conference on Intelligent Robots and Systems. Nice, France. <http://www.iros.org>

**19–22 Oct.** BioRob 2008: International Conference on Biomedical Robotics and Biomechatronics. Scottsdale, Arizona, USA. <http://www.biorob2008.org>

**28–31 Oct.** SSRR 2008: IEEE International Workshop on Safety, Security, and Rescue Robotics. Sendai, Japan. <http://www.rm.is.tohoku.ac.jp/ssr2008/cfp.html>

**10–12 Nov.** TePRA 2008: IEEE International Conference on Technologies for Practical Robot Applications. Woburn, Massachusetts, USA. <http://www.ieeerobot-tepra.org/>

**17–19 Nov.** DARS 2008: 9th International Symposium on Distributed Autonomous Robotics Systems. Tsukuba, Japan.

**2–5 Dec.** ICARV: 10th International Conference on Control, Automation, Robotics and Vision. Hanoi, Viet Nam. <http://www.icarvcv.org/2008/>

**4 Dec.** SI International 2008: IEEE/SICE International Symposium on System Integration. Nagoya, Japan. <http://www.rm.is.tohoku.ac.jp/SIInt08/>

**14–17 Dec.** ROBIO'08: IEEE International Conference on Robotics and Biomimetics. Bangkok, Thailand. <http://www.ee.cuhk.edu.hk/~qhmeng/robio/ROBIO2008-CFP.pdf>

2009

**11–13 Mar.** HRI 2009: ACM/IEEE International Conference on Human-Robot Interaction. San Diego, California, USA. <http://www.hri2009.org/>

Digital Object Identifier 10.1109/MRA.2008.922383

INDUSTRY / RESEARCH NEWS

(continued from page 131)

Thus, the authors describe the data presented as mostly descriptive and qualitative, presenting a big picture rather than a more rigorous statistical analysis, although they did some more intensive statistical analysis of the robot professionals who made up nearly half the respondents.

The Georgia Tech team is currently working on the second thrust of the study, which addresses the question “what can be done?” They are designing a “computational implementation of an ethical code within an existing autonomous robotic system, i.e., an ‘artificial conscience,’ that will be able to govern an autonomous system’s behavior in a manner consistent with the rules of war.”

A detailed report of the survey and its finding is reported in technical report GIT-GVU-07-16, which is available online at <http://www.cc.gatech.edu/ai/robotlab/onlinepublications/ArkinMoshkinaISTAS.pdf>.

References

- [1] Foster-Miller Inc. (2007). *Products & service: TALON military robots, EOD, SWORDS, and Hazmat Robots* [Online]. Available: <http://www.foster-miller.com/lemming.htm>
- [2] *Reaper moniker given to MQ-9 unmanned aerial vehicle*. (2006). Official Web site of the US Air Force. Available: <http://www.af.mil/news/story.asp?storyID=123027012>
- [3] B. Opal-Rome. (2007, June 1). *Israel wants robotic guns, missiles to guard Gaza* [Online]. Available: <http://Defensenews.com>
- [4] J. Kumagai, “A robotic sentry for Korea’s demilitarized zone,” *IEEE Spectrum*, vol. 44, no. 3, p. 16, Mar. 2007.
- [5] R. C. Arkin, “Governing lethal behavior: Embedding ethics in a hybrid deliberative/reactive robot architecture,” GVU, Atlanta, GA, Tech. Rep. GIT-GVU-07-11, 2007.
- [6] R. C. Arkin and L. Moshkina, “Lethality and autonomous robots: An ethical stance,” in *Proc. IEEE Int. Symp. Technology and Society*, Las Vegas, NV, June 2007. Available: <http://www.cc.gatech.edu/ai/robot-lab/online-publications/ArkinMoshkinaISTAS.pdf>

By Javier Ruiz-del-Solar and Alfredo Weitzenfeld

IEEE Region 9, Latin America, is the fastest growing of the IEEE's ten geographic regions. Currently, there are 497 IEEE Robotics and Automation Society (RAS) Members in the region. With the leadership of Alfredo Weitzenfeld, chair of the Mexico Chapter, and Javier Ruiz-del-Solar, chair of the Chile RAS Chapter, and with the help of the 2007 RAS New Initiative Grant and the support of the Region 9 IEEE Sections, RAS Members and others in the region have created the IEEE RAS Latin American Robotics Council, with the mission to unify, coordinate, and strengthen robotics activities in the region.

The US\$9,200 RAS grant was used to support primarily student travel and provide prizes for the first Latin American (LA) Summer School on Robotics, the sixth Latin American Robotics Competition in Monterrey, Mexico, 5–7 November 2007, and the sixth Chilean Student Robotics Contest, in Santiago, Chile, 4–5 October 2007.

The first IEEE RAS LA Summer School on Robotics was organized jointly with the fourth IEEE Computational Intelligence Society (CIS) LA Summer School on Computational Intelligence. Both events took place at the Universidad de Chile, on 12–14 December 2007. The organizing chair of the IEEE RAS LA Summer School on Robotics was Prof. Javier Ruiz-del-Solar, while the organizing chair of the IEEE CIS LA Summer School on Computational Intelligence was Prof. Pablo Estévez.

The two summer schools featured 11 plenary talks and 13 tutorials. The 150 participants included 100 undergraduate and graduate students. The summer school also included a student paper competition, which had six entries. The judges were Alicia Casals and Kim Boyer. The winning papers are as follows:

- ◆ **First Place:** Matías Arenas and Rodrigo Verschae for “Detection of Humanoid Robots Using Cascades of Boosted Classifiers”
- ◆ **Second Place:** Berardi Sensale, Pablo Romero, Diego Astessiano, and Rafael Canetti for “Development of an Autonomous Robotic Fish—Preliminary Results”
- ◆ **Third Place:** Roberto Cabrera for “Intelligent Self-Parking System.”

### The Sixth Latin American Robotics Competition

The Sixth Latin American Robotics Competition was organized in collaboration with the third RoboCup Latin American Open 2007 in Monterrey, Mexico, 5–7 November 2007.

Forty-seven teams with a total of 171 participants competed in the two events. Although most of the competitors were from the Latin American countries of Mexico (with the largest number), Colombia, Chile, Brazil, and Venezuela, teams from other countries such as China, the Netherlands, Japan, and Iran also took part. About 1,500 members of the general public came to watch the competitions.

### The Sixth Chilean Student Robotics Contest

This competition took place at the Universidad de Santiago de Chile on 4–5 October 2007. The main organizer was Prof. Renato Salinas. The competition had 36 participants and was organized into nine teams (six in the open category and three in the LEGO category). About 250 members of the general public also attended this event.

The council also carried out the following activities successfully during this year:

- ◆ National Activities
  - ◆ Fifth Brazilian Student Robotics Contest, Florianópolis, Brazil, 7–12 October 2007
  - ◆ Fourth Colombian Student Robotics Contest, Cartagena, Colombia, August 2007
  - ◆ Third Colombian Workshop on Robotics and Automation, Cartagena, Colombia, August 2007
  - ◆ Third Mexican Student Robotics Contest, Puebla, Mexico, 27–31 August 2007
  - ◆ First Peruvian Student Robotics Contest, Lima, Peru, 11–12 September 2007
  - ◆ Third Venezuelan Student Robotics Contest, Maracaibo, Venezuela, 8–10 October 2007
- ◆ Regional Activities
  - ◆ Sixth Latin American Robotics Competition (LARC) 2007, Monterrey, Mexico, 5–7 November 2007
  - ◆ Third RoboCup Latin American Open 2007, Monterrey, Mexico, 5–7 November 2007
  - ◆ Fourth Latin American Robotics Symposium (LARS) 2007, Monterrey, Mexico, 8–9 November 2007
  - ◆ First and the Fourth Latin American Summer School on Robotics
  - ◆ School on Computational Intelligence, Santiago, Chile, 12–14 December

For detailed information about all national and regional activities, visit the council's Web site <http://ewh.ieee.org/reg/9/robotica/>.



# ICRA2009

2009 IEEE International Conference  
on Robotics and Automation  
Kobe, Japan, May 13-17, 2009

#### Sponsors:

The IEEE Robotics and Automation Society  
The Science Council of Japan

#### Technical Co-Sponsors:

The Japan Society of Mechanical Engineers  
The Robotics Society of Japan  
The Society of Instrument and Control Engineers

#### Organizing Committee Chair

Fumio Harashima (Tokyo Denki Univ.)

#### General Chair

Kazuhiro Kosuge (Tohoku Univ.)

#### Program Chair

Katsushi Ikeuchi (Univ. Tokyo)

#### Program Co-Chairs

Kevin Lynch (North Western Univ.)

Raja Chatila (LAAS-CNRS)

Shigeki Sugano (Waseda Univ.)

#### Tutorials and Workshops Chairs

Yoky Matsuoka (Univ. Washington)

Cecilia Laschi (Scuola Superiore Sant'Anna)

Makoto Kaneko (Osaka Univ.)

#### Videos Chair

Peter Corke (CSIRO)

#### Science/Industry Forum Chairs

Hajime Asama (Univ. Tokyo)

Oussama Khatib (Stanford Univ.)

Ruediger Dillmann (Univ. Karlsruhe)

#### Publication Chairs

C. S. George Lee (Purdue Univ.)

Kazuhito Yokoi (AIST)

Zhidong Wang (Chiba Inst. Tech.)

#### Awards Chair

John M. Hollerbach (Univ. Utah)

#### Publicity Chairs

Ning Xi (Michigan State Univ.)

Max Meng (Chinese Univ. Hong Kong)

Hideki Hashimoto (Univ. Tokyo)

#### E-media Chair

Kenji Suzuki (Univ. Tsukuba)

#### Exhibitions Chair

Tatsuo Arai (Osaka Univ.)

Frank Park (Seoul National Univ.)

#### Finance Chairs

Hiromi Mochiyama (Univ. Tsukuba)

Xiaoping Yun (Naval Postgraduate School)

#### Local Arrangement Chairs

Satoshi Tadokoro (Tohoku Univ.)

Koichi Osuka (Kobe Univ.)

#### Secretariat

Yasushi Nakauchi (Univ. Tsukuba)

Yasuhsa Hirata (Tohoku Univ.)

#### Contact E-mail:

info@icra2009.org

## Preliminary Call for Contributions

The 2009 IEEE International Conference on Robotics and Automation (ICRA2009) will be held in Kobe, Japan, during May 13 - 17, 2009. The theme of the conference is "**Robotics and IRT for Livable Societies**", reflecting the ever growing interests in research, development and applications in the dynamic and exciting areas of robotics and automation. Kobe is a fascinating city, and its attractions include one of the largest ports in the world, the beautiful natural setting of the Rokko Mountains and picturesque streets. Located in the central part of the Japanese archipelago, Kobe is 30 minutes from Kyoto and three hours from Tokyo by Shinkansen (the super express train).

#### Contributed Papers:

Prospective authors should submit their papers electronically in PDF format. Six pages in the standard ICRA format are allowed for each paper, including figures. A maximum of two additional pages is permitted. Detailed instructions for paper submission will be available soon on the conference website.

#### Invited Sessions:

Invited sessions on new topics or innovative applications will be considered, consisting of four to six related papers submitted through the regular review process. Prospective organizers should include a brief statement of purpose for the sessions as well as the abstracts of the papers to be included.

#### Videos:

Videos of 1.5 to 3 minutes illustrating new and exciting results are sought for special video sessions. Videos should be submitted according to the instructions on the website.

#### Tutorials & Workshops:

Proposals for half-day or full-day tutorials and workshops should be submitted according to the instructions on the conference website. Prospective organizers should include: statement of objectives, intended audience, list of speakers, and list of topics.

#### Exhibits:

There will be an exhibition site at the conference. Information for prospective exhibitors will be available on the web site.

#### Important Dates: (Tentative)

Submission of contributions:	September 15, 2008
Notification of acceptance:	January 7, 2009
All final contributions due:	February 8, 2009

[www.icra2009.org](http://www.icra2009.org)



**2008**

**20–23 June** ICIA 2008: IEEE International Conference on Information Automation. Zhangjiajie, Hunan, China. <http://www.ieee-icia.info>

**25–28 June** RSS 2008: Robotics: Science and Systems. Zurich, Switzerland. <http://www.roboticsconference.org>

**2–5 July** AIM 2008: IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Xi'an, China. <http://www.aim2008.info>

**1–3 Aug.** RO-MAN 2008: IEEE International Symposium on Robot and Human Interaction. Munich, Germany. <http://www.ro-man2008.org>

**5–8 Aug.** ICMA 2008: IEEE International Conference on Mechatronics and Automation. Takamatsu, Japan. <http://www.icma2008.org/>

**20–22 Aug.** MFI 2008: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. Seoul, Korea. <http://www.mfi2008.org/>

**22 Aug.** SICE 2008: Annual Conference. Chofu, Japan. <http://www.sice.or.jp/sice2008/>

**23–26 Aug.** IEEE-CASE 2008: 4th International Conference on Automation Science and Engineering. Washington, District of Columbia, USA. <http://www.ieee-case.org>

**1–3 Sept.** ICAL 2008: IEEE International Conference on Automation and Logistics. Qingdao, China. <http://myweb.dal.ca/jgu/ical08/>

**16–17 Sept.** IEEE-CYBER: IEEE International Conference on Automation, Control, and Intelligent Systems. Shenyang, China. (Contact: E-mail: [tarn@wuauto.wustl.edu](mailto:tarn@wuauto.wustl.edu))

**22–26 Sept.** IROS 2008: IEEE/RSJ International Conference on Intelligent Robots and Systems. Nice, France. <http://www.iros.org>

**19–22 Oct.** BioRob 2008: International Conference on Biomedical Robotics and Biomechatronics. Scottsdale, Arizona, USA. <http://www.biorob2008.org>

**28–31 Oct.** SSRR 2008: IEEE International Workshop on Safety, Security, and Rescue Robotics. Sendai, Japan. <http://www.rm.is.tohoku.ac.jp/ssr2008/cfp.html>

**10–12 Nov.** TePRA 2008: IEEE International Conference on Technologies for Practical Robot Applications. Woburn, Massachusetts, USA. <http://www.ieeerobot-tepra.org/>

**17–19 Nov.** DARS 2008: 9th International Symposium on Distributed Autonomous Robotics Systems. Tsukuba, Japan.

**2–5 Dec.** ICARV: 10th International Conference on Control, Automation, Robotics and Vision. Hanoi, Viet Nam. <http://www.icarvcv.org/2008/>

**4 Dec.** SI International 2008: IEEE/SICE International Symposium on System Integration. Nagoya, Japan. <http://www.rm.is.tohoku.ac.jp/SIInt08/>

**14–17 Dec.** ROBIO'08: IEEE International Conference on Robotics and Biomimetics. Bangkok, Thailand. <http://www.ee.cuhk.edu.hk/~qhmeng/robio/ROBIO2008-CFP.pdf>

**2009**

**11–13 Mar.** HRI 2009: ACM/IEEE International Conference on Human-Robot Interaction. San Diego, California, USA. <http://www.hri2009.org/>

Digital Object Identifier 10.1109/MRA.2008.922383

I N D U S T R Y / R E S E A R C H N E W S

(continued from page 131)

Thus, the authors describe the data presented as mostly descriptive and qualitative, presenting a big picture rather than a more rigorous statistical analysis, although they did some more intensive statistical analysis of the robot professionals who made up nearly half the respondents.

The Georgia Tech team is currently working on the second thrust of the study, which addresses the question “what can be done?” They are designing a “computational implementation of an ethical code within an existing autonomous robotic system, i.e., an ‘artificial conscience,’ that will be able to govern an autonomous system’s behavior in a manner consistent with the rules of war.”

A detailed report of the survey and its finding is reported in technical report GIT-GVU-07-16, which is available online at <http://www.cc.gatech.edu/ai/robotlab/onlinepublications/ArkinMoshkinaISTAS.pdf>.

**References**

- [1] Foster-Miller Inc. (2007). *Products & service: TALON military robots, EOD, SWORDS, and Hazmat Robots* [Online]. Available: <http://www.foster-miller.com/lemming.htm>
- [2] *Reaper moniker given to MQ-9 unmanned aerial vehicle*. (2006). Official Web site of the US Air Force. Available: <http://www.af.mil/news/story.asp?storyID=123027012>
- [3] B. Opal-Rome. (2007, June 1). *Israel wants robotic guns, missiles to guard Gaza* [Online]. Available: <http://Defensenews.com>
- [4] J. Kumagai, “A robotic sentry for Korea’s demilitarized zone,” *IEEE Spectrum*, vol. 44, no. 3, p. 16, Mar. 2007.
- [5] R. C. Arkin, “Governing lethal behavior: Embedding ethics in a hybrid deliberative/reactive robot architecture,” GVU, Atlanta, GA, Tech. Rep. GIT-GVU-07-11, 2007.
- [6] R. C. Arkin and L. Moshkina, “Lethality and autonomous robots: An ethical stance,” in *Proc. IEEE Int. Symp. Technology and Society*, Las Vegas, NV, June 2007. Available: <http://www.cc.gatech.edu/ai/robot-lab/online-publications/ArkinMoshkinaISTAS.pdf>



### Special Issue on Rehabilitation Robotics

Rehabilitation Robotics aims at the development of novel medical solutions for assisted motor therapy and functional assessment of patients with reduced motor and/or cognitive abilities in order to ultimately favor the best achievable functional recovery. Robot-assisted therapy emphasizes the central role of the patient during the motor exercise. This poses major technical challenges for the design of safe and effective robotic platforms. Typical requirements include high backdriveability, easy adaptation to different anthropometric parameters, adaptive control schemes for interaction control, friendly human-machine interfaces for motivating the patient and for allowing customization of robot performance by the doctor or therapist. A new generation of rehabilitation robots, to be conceived and developed in tight cooperation with medical experts and end users, is expected to come in the near future. Researchers working in the academy or industry are invited to submit papers to this Special Issue of the *IEEE Transactions on Robotics* (T-RO) on the peculiar theoretical, technological and experimental aspects of design, development, and validation of novel rehabilitation robotic systems.

#### Topics

- Novel robotic systems for application to rehabilitation
- Human-centred design methods and case studies of rehabilitation robots
- Robotic platforms for functional assessment and human behavioral analysis
- Exoskeletons and operational machines for lower and/or upper limb rehabilitation
- Robotic systems for telerehabilitation and homecare
- Portable robotics systems for ubiquitous rehabilitation
- Backdriveable mechanisms, compliant actuators and other innovative components for rehabilitation robots
- Physical human-robot interaction control in rehabilitation applications
- Impedance control, adaptive motor control and learning in rehabilitation robotics
- Multimodal and natural human-robot interfaces for rehabilitation
- Robotic systems for cognitive rehabilitation and for diagnosis and treatment of neurodevelopmental disorders
- Magnetic-imaging (MI) compatible robotic systems
- Application of robotic systems for biomechanical modeling of the human body
- Robotic systems for prevention of age-related motor disabilities

#### Important dates

<b>May 15, 2008:</b>	Call for Papers
<b>September 15, 2008:</b>	Deadline for Paper Submission
<b>December 31, 2008:</b>	Completion of First Review
<b>March 15, 2009:</b>	Completion of Final Review
<b>June 2009 (tentative):</b>	Publication

#### Guest Editors

**Dr. Takanori Shibata**

Intelligent Systems Research Inst.  
AIST Tsukuba Central 2  
Tsukuba, Ibaraki 305-8568, Japan  
Email: [shibata-takanori@aist.go.jp](mailto:shibata-takanori@aist.go.jp)

**Dr. Michelle J. Johnson**

Dept. of Physical Medicine and Rehabilitation  
Medical College of Wisconsin  
Milwaukee, WI 53226, USA  
Email: [mjohnso@mcw.edu](mailto:mjohnso@mcw.edu)

**Prof. Eugenio Gugliemelli**

Lab. of Biomedical Robotics and Biomicrosystems  
Università Campus Bio-Medico  
00128 Roma, Italy  
Email: [e.guglielmelli@unicampus.it](mailto:e.guglielmelli@unicampus.it)

#### Submission and Review of Papers

Author information is available at the T-RO web site <http://www.ieee-ras.org/tro>. Submissions should go to T-RO PaperCept at <http://ras.papercept.net/journals/tro>. T-RO considers also accompanying multimedia material. Papers submitted to the Special Issue undergo the usual T-RO review process. For further information on the Special Issue, please check the IEEE RAS TC on Rehabilitation & Assistive Robotics web site at <http://tab.ieee-ras.org/>.