VOL. 9, NO. 1 MARCH 2000

SPIE's
International
Technical
Group
Newsletter

Robotics
and Machine Perception

Pattern recognition for goal-based rover navigation

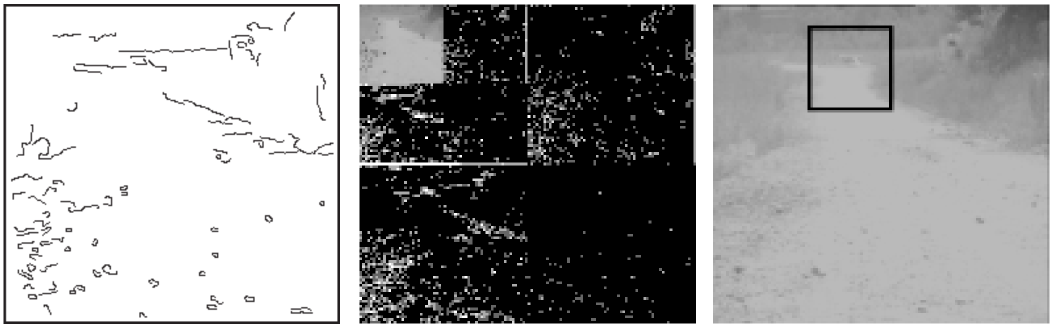


Figure 1. Horizontal line and wavelet derived features used for fusion to obtain lander heading from long distances (~125 meters). (Left) Canny derived edge features. (Middle) Wavelet coefficients with a 2 level transform. (Right) Detected lander labeled by boxed area.

Figure 2. Truss structure features used for mid-range navigation. (Left) Original image. (Middle) Edge features. (Right) Truss structure features.

The baseline '03/'05 Mars Sample Return missions require the return of a science rover to the lander for the transfer of sample cache containers to a Mars Ascent Vehicle or MAV. Along these lines, the newest mission guidelines for the Mars Sample Return call for a science rover to descend from the lander using ramps, acquire core samples from as far away as hundreds of meters from the lander, return to the lander, and then ascend the ramps to deposit these samples in the MAV. The return operation requires tracking and docking techniques for the development of necessary integrated rover capabilities key to the lander rendezvous operation. The science rover must autonomously recognize, track, and precisely rendezvous with the lander from distances as far away as hundreds of meters. The Sample Return Rover, or the SRR, is a rover prototype that was originally developed for the rapid retrieval of samples collected by longer ranging mobile science systems, and the return of these samples to an Earth ascent vehicle.

We have developed a multifeature fusion algorithm that integrates the outputs of horizontal line and wavelet-based visual area-of-interest operators[1] for lander detection from significant distances. The horizontal line detection algorithm is used to localize possible lander candidates based on detection of the lander deck. The wavelet transform is then performed on an acquired image and a texture signature is extracted in a local window of the wavelet coefficient space for each of the lander candidates found above. The multifeature fusion algorithm eliminates the false positives in this process arising from features in the surrounding terrain. This technique is coupled with a 3D visual terminal guidance algorithm that can extract and utilize cooperative features of the lander to accurately, iteratively estimate structure range-and-pose estimates, and then steer the rover to the bottom of the ramps.

We have tested, in the arroyo at JPL, a three-phase sequence that uses these pattern recognition algorithms for the lander-rendezvous portion of the Mars Sample Return missions. The first phase is the long-range traverse, in which the SRR initiates a search for the lander using images acquired by the goal camera

continued on p. 3

Newsletter now available on-line

Beginning with this issue, Technical Group members will be offered the option to receive the Robotics and Machine Perception Newsletter in an electronic format. An e-mail notice is being sent to all group members advising you of the web site location for this issue and asking you to choose between the electronic or printed version for future issues. If you have not yet received this e-mail message, then SPIE does not have your correct e-mail address in our database. To receive future issues of this newsletter in the electronic format please send your e-mail address to spie-membership@spie.org with the word ROBOTICS in the subject line of the message and the words "Electronic version" in the body of the message.

If you prefer to continue to receive the newsletter in the printed format, but want to send your correct e-mail address for our database, include the words "Print version preferred" in the body of your message.

Editorial

Modular robotics and distributed systems

Modular robotics is an emerging area of research focused on the configuration of robotic resources into reconfigurable structures and control systems. Recent work has aimed at developing a range of standard off-the-shelf mechatronic modules that can be plugged together to create diverse robot manipulator configurations. The CMU Reconfigurable Modular Manipulator System (RMMS)[1] is a representative example of research in this area.

The recent conference on Sensor Fusion and Decentralized Control in Robotics Systems[2] featured a number of sessions that addressed this important area and highlighted some of its most significant developments. Of particular note was the number of real systems that have now been fielded in order to investigate the practical feasibility of both configurable and reconfigurable systems. Many of these were simulations in the eye of the researchers a couple of years ago. Now they offer insights into a new era, where industry can expect robotics solutions that speak more directly to their needs.

The emergence of the modular systems approach in robotics, and indeed the development of layered robot architectures and cooperative robotics systems, has drawn the attention of the robotics community to message passing and networking models for communicating between local and distributed entities. We should take particular note, therefore, of the release of the Jini™ specification[3] from Sun Microsystems and the entertainment and media industry development of the HAVi specification[4] for home networking. These developments aim at providing integration of devices distributed across industrial, office and home networks.

The key issues of concern to all of these areas is the modelling and management of resources, the integration of resources into systems that perform useful functions, and the development of software toolkits to reflect and support these activities. Robotics can contribute to these developments, which offer in turn the potential for robotics and artificial intelligence technologies to penetrate new application areas in industry and the home.

Gerard McKee
Robotics and Machine Perception
Technical Group Chair
The University of Reading, UK
G.McKee@Reading.ac.uk

References

1. C.J. J. Paredis and P. K. Khosla, *Synthesis Methodology for Task Based Reconfiguration of Modular Manipulator Systems*, **Proc. 6th Int'l Symp. Robotics Research:** ISRR, 1993.
2. **Proc. SPIE 3839,** 1999.
3. The JiniTM Technology Specification, Sun Microsystems, at http://www.sun.com/jini/specs/.
4. HAVi: Home Audio/Video Interoperability. Information and specifications at http://www.havi.org/.

Self-repairing mechanical system

We have proposed the concept of a self-repairable machine made up of homogeneous mechanical units. Each unit is an active robotic element with some mobility to allow it to change its local connections, an ability that allows a collection of units to metamorphose into a desired shape through local cooperation (self-assembly). When some of the system units are damaged, they can be detected via communication error and then eliminated, after which spare units are brought in so that the shape can be reassembled (self-repair).

This kind of system can be used as a primary structure in space or deep-sea applications because it is able to change its shape and functionality according to environmental changes. It can also be used as a flexible robotic manipulator that can change its configuration according to required workspace, or change into various types of walking machine to cope with unknown terrain. Ordinary mechanical systems with fixed configurations cannot achieve this versatility.

We have developed several of these systems and verified their self-assembly and self-repair capabilities.[1,2] The first prototype system is made up of 2D units that connect with other units through electro-magnetic interaction (see Figure1). By switching the polarity of the electromagnets, they can change their local situation. Figure 2 shows the basic reconfiguration procedure. Repeating these local processes, a group of the units can change its global shape.

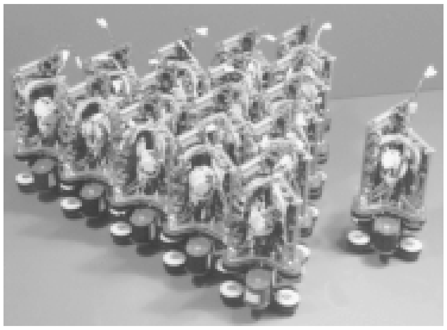


Figure 1. Two dimensional electro-magnetic unit system.

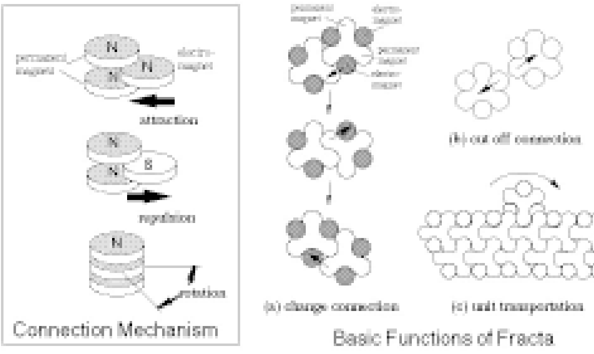


Figure 2. Basic procedures of self-assembly/self-repair.

Each unit has an onboard CPU (Z80) that controls its motion. It also has digital communication channels to adjacent units, thus allowing the full collective to work as a distributed intelligent system. We have developed a homogeneous distributed algorithm to allow the units to form a target shape and then repair the shape when the system contains a faulty unit. Here, "homogeneous" means that each unit is driven by an identical program and data set. In other words, they do not know their role in the system at the initial time, thus they have to differentiate to their appropriate roles in the system through cooperation via inter-unit communication.

A 3D unit system has also been developed (see Figure 3). This fully mechanical system has one DC motor that rotates six orthogonal arms, with a drive connection mechanism at the end of each arm. The motion of the unit is rather slow (it takes 60s to change connection points), which is due to the mechanical connection be-

Pattern recognition

(continued from cover)

mounted on its arm. A raster grid search pattern is implemented by serving the arm. The images are analyzed using the multifeature fusion algorithm that has been tuned to the specific characteristics of the lander. This operation is shown in Figure 1, where the outputs of the horizontal line algorithm and the wavelet based area-of-interest operator are shown on the left and the final targeting is shown on the right. The rover was 125m from the lander at this point in time. After the SRR visually acquires the lander, the traverse is started with visual reacquisition of the lander every 5m. This mode of operation continues until SRR is within 25m of the lander based on fusion of wheel odometry data and visual angle sizing of the lander.

During the second phase, the SRR traverses from 25m to a ramp standoff location of 5m. Since the relative distance is more important than absolute position during this phase, we use parallel line features visually extracted from the lander truss structure. In order to correctly distinguish the truss structure itself, the lander deck is detected first, which appears as a set of nearly leveled lines. Any parallel lines below the deck are considered as part of the truss structure. The average distance between the parallel pairs is then used to estimate the distance, and the center of mass of these parallel lines is used to compute the heading direction. This operation is shown in Figure 2, where the original image is shown on the right, and the visually extracted features are shown on the left.

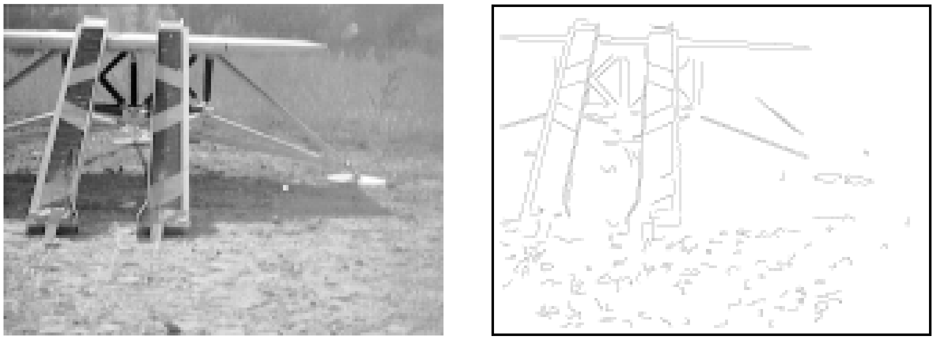


Figure 3. Ramp pattern features used for close-range navigation. (Left) Original image. (Right) Detected ramp pattern features.

During the third phase prior to climbing the ramps, SRR uses a distinctive pattern of stripes on the ramps to position itself within 10cm of the base of the ramps. Once the SRR is in around 5m range, it will circle around the lander. The close-range ramp alignment algorithm includes three major steps: feature extraction, feature match, and pose estimation. Features are the six stripes that were deliberately arranged on the ramps so that any two strips have a unique combination spatially and topologically. This unique configuration greatly reduced uncertainty and computational complexity. An edge detection algorithm (Canny edge detection) is applied first, then all straight-line segments are extracted. In order to find the stripes, we first look for the ramps, which are defined by a set of long straight and nearly parallel lines.

With a single detected stripe in the image, a linear affine transformation can be constructed based on the four corners. If this match is correct, the transformation can help to find other matches. Because there are relatively few stripes (six on the ramps), an exhaustive search is used to pick up the best match. Once the matches are found, the pose and orientation are estimated by using the outside corners of the stripes. A minimum of four stripes is used for safe navigation. This operation is shown in Figure 3, where the original image is shown on the right, and the visually extracted features are shown on the left.

Using the SRR, we have demonstrated pattern recognition algorithms for the lander return phase of the planned Mars Sample Return missions. The SRR autonomously acquired the lander from 125m away in an outdoor environment using multifeature fusion of line and texture features, traversed the intervening distance using features visually derived from the lander truss structure, and aligned itself with the base of the ramps prior to climbing using precise pattern matching for determining rover pose and orientation.

This research was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Yang Cheng and **Terry Huntsberger**
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109, USA
E-mail: {yang.cheng, terry.huntsberger}
@jpl.nasa.gov

Reference

1. F. Espinal, T. Huntsberger, B. D. Jawerth, and T. Kubota, *Wavelet-based fractal signature analysis for automatic target recognition,* **Optical Engineering 37** (1), p. 166, January, 1998.

---

ing very stiff. We have built four units to verify the reconfiguration functionality. Homogeneous software for self-assembly and self-repair for 3D systems has also been developed in simulation.

We are now working on new systems. One of them is a homogeneous modular system capable of self-assembly, self-repair, and also motion generation as a robotic system.[3] As part of a lattice system (such as those described previously), the modules can metamorphose into various configurations without outside help, then function as a robotic system to generate a group motion such as walking or wall climbing. The other is a miniaturized unit system. By using a shape memory alloy (SMA) actuator, we have built a small (4cm) 2-D unit.[4]

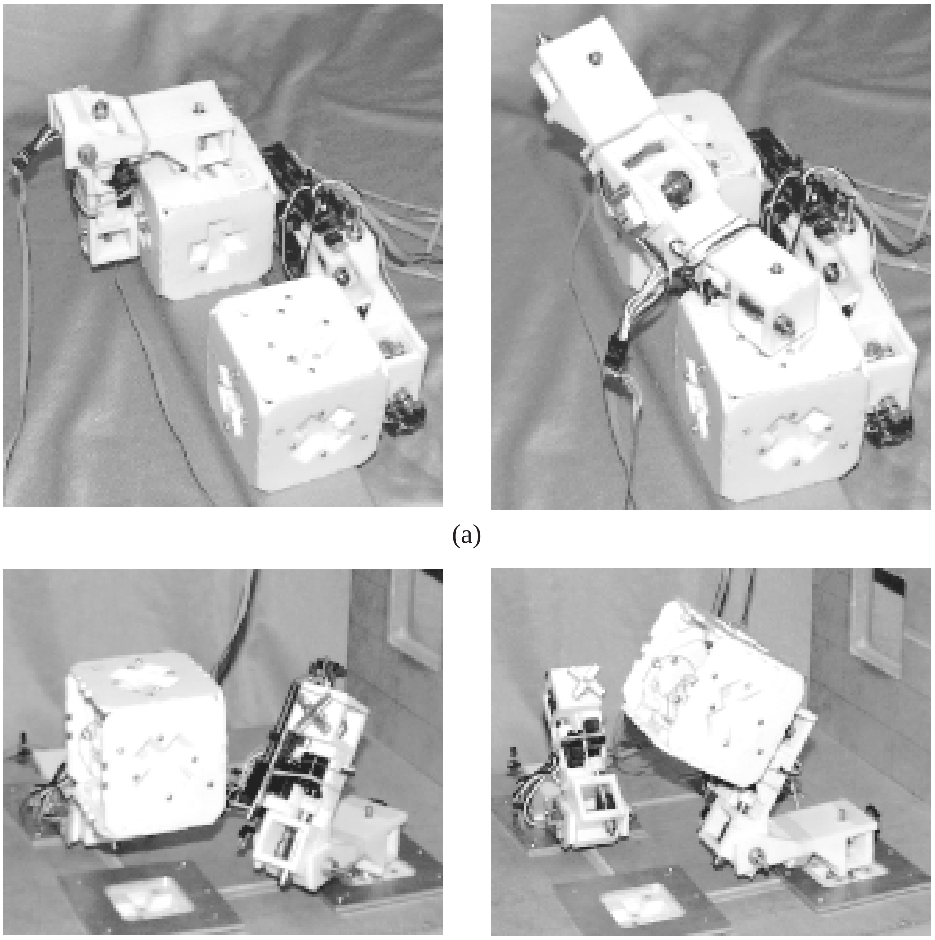
Figure 3. Three dimensional mechanical unit system.

Satoshi Murata
Mechanical Engineering Laboratory
AIST, MITI
1-2 Namiki, Tsukuba, 305-8564 Japan
Phone: +81-298-58-7125
Fax.:+81-298-58-7091
E-mail: murata@mel.go.jp

References

1. S. Murata, H. Kurokawa, and S. Kokaji, *Self-assembling Machine,* **Proc. IEEE Int'l. Conf. Robotics and Automation,** pp. 441-448, 1994.
2. S. Murata, et al., *A 3-D Self-reconfigurable Structure,* **Proc. IEEE Int'l. Conf. Robotics and Automation,** pp. 432-439, 1998.
3. S. Murata, et al., *Self-reconfigurable Modular Robotic System, Intl. Workshop on Emergent Synthesis,* IWES '99, pp. 113-118, 1999.
4. E.Yoshida, et al., *Miniaturized Self-reconfigurable System using Shape Memory Alloy,* **Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems: IROS99,** pp.1579-1585, 1999.

I-Cubes: A modular self-reconfigurable bipartite system


(a)

(b)

Figure 1. (a) A link transferring from one cube to another. (b) Two links exchanging a cube.

Statically stable gaits currently available for mobile robots include using wheels, treads, and similar devices that limit the robot's locomotion capabilities. A robot on wheels is usually incapable of climbing a set of stairs, or moving over large obstacles. Recent trends in autonomous mobile robot research have tried to find solutions for this problem by designing robots with different gait mechanisms that can providing locomotion in non-ideal environments. These efforts are aimed at slowly emerging applications that use small inexpensive robots to accomplish tasks in unstructured environments and narrow spaces.

The I-Cubes (or ICES-Cubes)[1] project comes out of a vision of a modular self-reconfigurable group of robots comprised of two module types with different characteristics. When combined as a single entity, sufficient modules will be capable of self-reconfiguring into defined shapes, which in turn will provide a new type of locomotion gait that may be combined with other capabilities. A large group of modules that can change its shape according to the locomotion, manipulation or sensing task at hand will then be capable of transforming into a snake-like robot to travel inside a air duct or tunnel, a legged robot to move on unstructured terrain, a climbing robot that can climb walls or move over large obstacles, a flexible manipulator for space applications, or an extending structure to form a bridge.

Designing identical elements for a modular system has several advantages over large and complex robotic systems. The units can be mass-produced, and their homogeneity can provide faster production at a lower cost. A large system consisting of many elements is less prone to mechanical and electrical failures, since it would be capable of replacing nonfunctioning elements by removing them from the group and reconfiguring its elements. Homogeneous groups of modules that are capable of self-reconfiguring into different shapes also provide a manufacturing solution at the design phase where identical elements are considered, while providing a modular system that can be rearranged for different tasks.

To obtain the advantages listed above, a modular system must have several essential properties, such as geometric, physical, and mechanical compatibility among individual modules. Several design issues need to be considered for a modular self-reconfiguring system to be truly autonomous.

I-Cubes are a bipartite robotic system, which is a collection of independently controlled mechatronic modules (links) and passive connection elements (cubes). Links are capable of connecting to and disconnecting from the faces of the cubes. Using this property, they can move themselves from one cube to another. In addition, while attached to a cube on one end, links can move a cube attached to the other end (Figure 1). We envision that all active (link) and passive (cube) modules are capable of permitting power and information flow to their neighboring modules. A group of links and cubes does in fact form a dynamic 3-D graph where the links are the edges, and the cubes are the nodes. When the links move, the structure and shape of this graph changes.

This self-reconfiguring system has the following properties:

- Elements can be independently controlled; only the cube attached to the moving end of a link is affected by link motions.
- All elements have the same characteristics and are mechanically/computationally compatible, i.e., any link can connect to any cube.
- The 3-D graph formed by the elements fits a cubicle lattice to guarantee interlocking of neighboring elements
- Active elements have sufficient degrees of freedom to complete motions in three-dimensional space.

Since all the actuation for self-reconfiguration (with the exception of the attachment mechanism) is provided by the links, cubes can be used to provide computation, sensing and power resources. If the modules are designed to exchange power and information, the cubes can be equipped with on-board batteries, microprocessors and sensing modules to become the "brains" of the system. Furthermore, it is possible to remove some of the attachment points on the cubes to provide these modules with different and faster gaits, such as wheeled locomotion. Specifically, we envision small

continued on p. 7

Modular robotics development at MMS

Modular Motion Systems Inc (MMS) in Palos Verdes, California, is engaged in the development and marketing of modular robotic equipment and control software for industrial and research applications. The company, working with many local organizations providing manufacturing services, holds foreign and domestic patents covering basic modular robotic concepts and offers a wide range of robotic modules, software for application programming, simulation and runtime operation, application engineering services ,and customer support.

The MMS system offers a family of robotic modules that allow application engineers or researchers to build manipulator systems that fit their needs. There are two types of MMS module: *joints* and *links*, both provided in a range of sizes (5, 7, 10, 14 and 20cm diameter). The joints are the active elements of the system, and come in two types: rotary joints and linear joints. These active joints are connected together with passive link modules that provide the structural geometry of manipulator configurations (see Figure 1). The user may configure as many degrees of freedom (i.e. number of joints), in any desired arrangement (links), as are desired for the application.

The configurations are programmed and operated by a single desktop PC control computer (ACP) which offers the user a programming environment called the Model Manager. This presents an array of graphical programming tools for online and off-line operations. Central to the system architecture, the Model Manager embodies software designs that host the integration of several machine intelligence technologies, including computer vision, planning, simulation and multi-agent cooperation.

The joint modules are completely self-contained. Each has its own control processor to decode motion commands, perform kinematic solution calculations, generate precise trajectory interpolations, and manage power flow to the DC servomotor. Each joint also has an onboard power amplifier driving the servomotor in pulse-width-modulated (PWM) mode, generating high power levels with high positioning accuracy and very low levels of waste heat. The only connection required for each module, 48V dc with RF communication superimposed, is made automatically as the modules are assembled. As seen in Figure 2, there are no external cables.

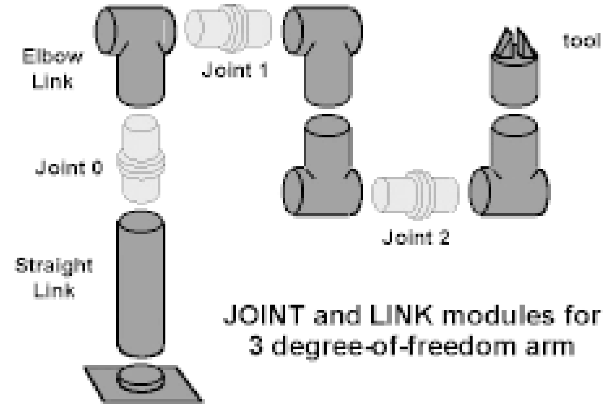


Figure 1. The two MMS Module types.

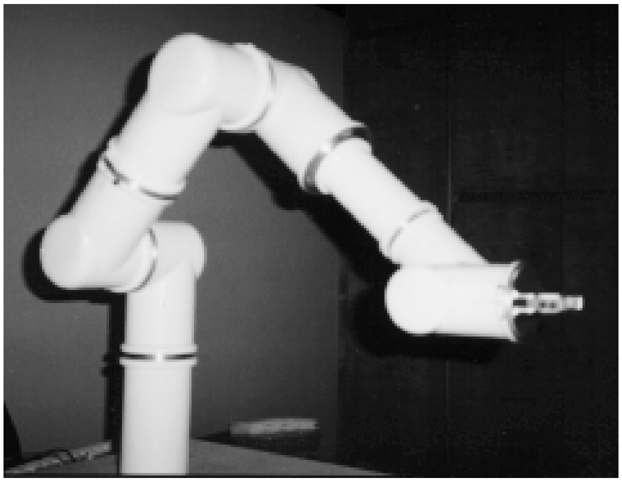


Figure 2. Modules assembled into PUMA configuration.

The modular architecture offers many advantages in practice. The original objective was to provide a cost-effective means of building either simple or complex systems that are not commercially available. Substantial cost savings have been discovered, however, when building standard robot configurations like PUMA or SCARA units. These advantages derive from the use of much simpler system building blocks, i.e. the 1-DOF modules, which are easy to manufacture and maintain.

Simple MMS configurations might include one degree-of-freedom (DOF) arrangements to steer a vehicle or control a plant process, two DOF arrangements to gimbal a camera or antenna, or three or four DOF systems to perform simple repetitive tasks. While hard automation devices that require engineering and fabrication might be more appropriate for some long term applications, rapid deployment and easy reuse of modules make MMS more cost effective for shorter term needs.

Complex MMS configurations, on the other hand, can be assembled and deployed at much lower cost than would be spent to develop a special purpose design. Such applications as multi-legged walkers, branching arms with multiple end-effectors, mobile platforms hosting diverse activities, or multiple arms working in coordination on a single task, are all examples of complex systems that might be deployed at much lower cost with a modular system. In addition, the modular architecture makes maintenance quicker and easier. Even the standard factory configurations can be serviced in minutes instead of hours with a simple module replacement, the Model Manager control software knows when and where problems may exist.

Several design improvements are in development for the next generation of MMS units. The new module interconnect latch is practically impossible to attach incorrectly and cannot loosen from activity. The new electrical interconnect scheme associated with the new latch is simpler and more reliable. The use of fiber composite materials is further reducing the cost and weight of the link modules. Every module configuration requires a unique kinematic solution: a set of equations translating Cartesian end-effector position into joint angles. An algorithm based on Fuzzy Associative Memory (FAM) methods has been developed to automate this procedure. Most important, the Model Manager control software is continuously expanding to include more services. It has also been rehosted from Lisp to a Java platform and the simulation functions now use Java3D.

More detailed information is available through the web address below and the reference given.[1]

William Schonlau
Modular Motion Systems
31107 Marne Drive
Rancho Palos Verdes, CA 90275
E-mail: info@mms-robots.com
http://www.mms-robots.com

Reference

1. William Schonlau, *MMS: a modular robotic system and model-based control architecture,* **Proc. SPIE 3839,** pp. 289-296, 1999.

PolyBot and Proteo: Two modular reconfigurable robot systems

Modular robotic systems are composed of modules that can be disconnected and reconnected in different arrangements to form a new system enabling new functionalities. Self-reconfigurability arises when the system can rearrange its own modules. We are interested in leveraging the repeated use of one or two module types giving a large variety of configurations of increasing complexity without a high cost in design. The resulting systems promise to be versatile (by adapting their shape to the task), robust (through redundancy), and low cost (via batch fabrication).[1-4]

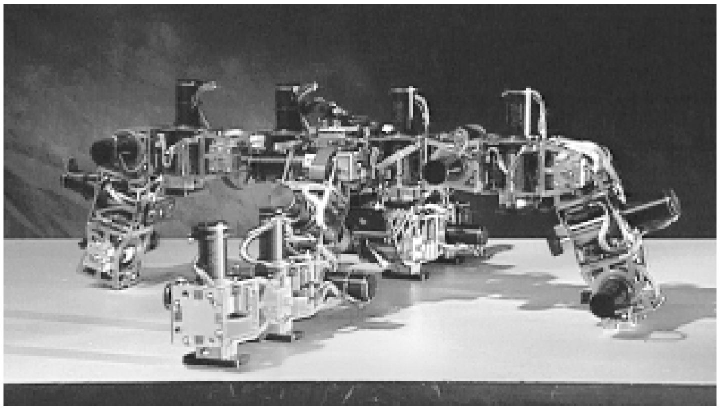


Figure 1. PolyBot with 25 modules in a four-legged spider configuration.

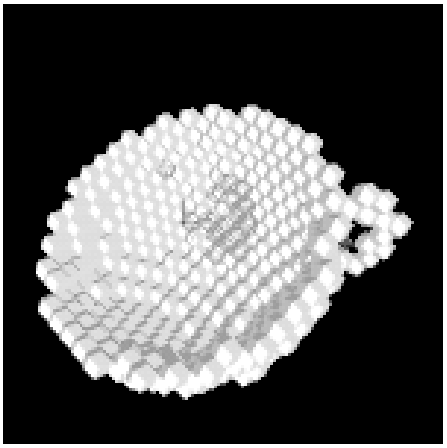


Figure 2. Proteo simulation with 441 modules in a teacup formation.

After prototyping several module types over the past eight years to prove the basic concepts in,[4,6] as well as more recent development work,[2,3,7,8] it is clear that the feasibility of these systems has been established. While these projects have involved the construction of only tens of modules at the most, we are now interested in exploring issues that arise as the numbers of modules are increased to hundreds or thousands while maintaining a small number of module types. We are exploring two basic systems: one called PolyBot, the most recent system under development, which involves snake-like or octopus-like structures and motions; and another called Proteo, which can form rigid structures (somewhat like autonomous bricks) that can rearrange themselves into different shapes.

PolyBot

PolyBot is comprised of two types of modules, one called a *segment*, which contains most of the functionality, the other called a *node*. It is easiest to visualize the modules as cube-shaped. Segments have two connection plates (for both mechanical and communications and power coupling), and one rotational degree of freedom. Each segment contains a low-speed high torque motor, a Motorola Power PC 555 micro-controller, motor current and position sensing, and proximity sensing. The node is rigid with six connection plates (one on each side of the cube). It serves two purposes: it allows more than serial chain configurations, and the space inside the cube can house such things as more powerful computing resources and batteries.

An earlier project, Polypod demonstrated three different classes of locomotion (a tumbling motion, an earthworm-like gait, and a caterpillar-like gait) with the 11 modules that were constructed. In addition, another five classes of locomotion were simulated.[9] Its successor PolyBot, has demonstrated another four classes of locomotion (a rolling-track, a sinusoidal snake-like gait, a platform of with up to six legs, and a four-legged spider) with up to 32 modules connected together. In addition, PolyBot has demonstrated distributed manipulation of paper, boxes and tennis balls.[10] These demonstrations show the versatility of modular systems.

In addition to particular tasks and configurations, we have demonstrated reconfiguration from a loop to a snake and from a snake into a four-legged spider configuration. This last reconfiguration involved 24 modules and was guided by a human teleoperator. The spider configuration is shown in Figure 1. A similar reconfiguration sequence was simulated with 180 modules.[11]

Proteo

Proteo is the name of a system that uses substrate or lattice reconfiguration. The motions of the modules are along one dimensional paths stopping only at positions that form a lattice. In addition each module can only move itself relative to one neighbor. These constraints ease the computational problems of inverse and forward kinematics and self-collision detection by restricting the motions to be local and discrete.[12] The current design of Proteo involves a rhombic dodecahedron (RD) shape which packs in a face-centered-cubic fashion. Actuation would occur by the rotation of the RD about an edge into one of the 12 adjacent positions. In contrast to PolyBot, the Proteo modules were intended to be simpler and smaller since they need little sensing, and only discrete actuation. In practice, it is turning out to be difficult to construct.

Since one module can only move itself (or, equivalently, one neighbor) to an adjacent position, the functionality is essentially the reconfiguration of the conglomerate shape. We have been exploring the planning problem of forming arbitrary shapes,[13] and have developed and simulated six algorithms. The latest is the combination of two methods: the first is a goal-ordered method in which every module attempts to reach the closest available goal position in the target configuration according a pre-planned order; and the second is a simulated heat flow method in which module motion is directed by the flow of heat which is generated at goal positions and propagated between modules. This system has proven to be able to successfully plan the motion of up to 600 modules with only local communications and state knowledge between modules[14] (see Figure 2).

Future work

By the end of year 2001 we hope to have demonstrated 200 PolyBot modules performing autonomous reconfiguration and locomotion. We should also have demonstrated Proteo modules reconfiguring into multiple shapes as well as simulating thousands of modules. While the modules will each fit inside a cube of 5cm on a side by the 2001 deadline, the long term goal is to shrink the modules down as far as possible while increasing the numbers past the thousands: perhaps to millions.

This work is funded in part by the Defense

Advanced Research Project Agency (DARPA) contract # MDA972-98-C-0009. Contributors to this effort include Sunil Agrawal, Arancha Casal, David Duff, Craig Eldershaw, Rob Gorbet, Leo Guibas, Maia Hoeberechts, Sam Homans, Kea Kissner, An Thai Nguyen, Kimon Roufas, John Suh, Kevin Wooley, and Ying Zhang.

Mark Yim
Xerox Palo Alto Research Center
3333 Coyote Hill Rd
Palo Alto, CA 94304
E-mail: yim@parc.xerox.com

References

1. T. Fukuda and S. Nakagawa, *Dynamically Reconfigurable Robotic System,* **Proc. IEEE Int'l Conf. on Robotics and Automation,** pp. 1581-1586, 1988.
2. S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji, *A 3D SelfReconfigurable Structure,* **Proc. IEEE Int'l Conf. on Robotics and Automation,** pp. 432-439, May 1998.
3. A. Pamecha, C. Chiang, D. Stein, and G.S. Chirikjian, *Design and Implementation of Metamorphic Robots,* **Proc. ASME Design Eng. Tech. Conf. and Computers in Eng. Conf.,** Irvine, California, August 1996.
4. M. Yim, *New Locomotion Gaits,* **Proc. IEEE Int'l Conf. on Robotics and Automation,** pp. 2508-2514, May 1994.
5. M. Yim, *A Reconfigurable Modular Robot with Many Modes of Locomotion,* **Proc. of JSME Int'l Conf. on Advanced Mechatronics,** pp.283-288, 1993
6. M.Yim, D.G. Duff, and K.D. Roufas, *PolyBot: A Modular Reconfigurable Robot,* accepted to, **Proc. of the IEEE Intl. Conf. on Robotics and Automation,** San Francisco 2000
7. D. Rus and M. Vona, *Self-reconfiguration Planning with Compressible Unit Modules,* **Proc. IEEE Int'l Conf. on Robotics and Automation,** pp. 25132520, May 1999.
8. P. Will, A. Castano, and W.-M. Shen, *Robot modularity for selfreconfiguration,* **Proc. SPIE 3839,** pp. 236-245, September 1999.
9. http://www.parc.xerox.com/modrobots (online since 1997)
10. M. Yim, J. Reich, and A.A. Berlin, *Two Approaches to Distributed Manipulation,* **Distributed Manipulation,** K. Bohringer and H. Choset (eds.), Kluwer Academic Publishing, 2000.
11. http://robotics.stanford.edu/users/mark/multimode.html (online since 1994)
12. A. Casal and M. Yim, *Self-Reconfiguration Planning for a Class of Modular Robots,* **Proc. SPIE 3839,** pp. 246-257, September 1999.
13. A.T. Nguyen, L.J. Guibas, and M. Yim, *Controlled Module Density Helps Reconfiguration Planning,* accepted **4th Int'l Workshop on Algorithmic Foundations of Robotics: WAFR**, 2000.
14. Y. Zhang, M. Yim, J. Lamping, and E. Mao, *Distributed Control for 3D Shape Metamorphosis,* submitted **Autonomous Robots, special issue on self-reconfigurable robots,** 1999.

The IRIS Laboratory

continued from p. 12

walls, roads, opponent robots, and obstacles. All of these objects may be included in a simulation by editing an easily understandable human-readable description file.

TeamBots supports the low-cost platform in simulation and in hardware (Figure 2).

Multirobot formations

The potential-field approach is a well-known strategy for robot navigation. In this paradigm, repulsive and attractive fields are associated with important objects in the environment (eg. goal locations or obstacles to avoid). To navigate, the robot computes the value of the vectors corresponding to each relevant field, then combines them (usually by summation) to compute a movement vector based on its current position. The result is emergent navigational behavior reflecting numerous c constraints and/or intentions encoded in the robot's task-solving behavior.

We have extended the mechanism to multiple robots so that the potential field impacting a robot's path is shaped by the presence of team or opponent robots. We call these potential functions *social potentials*. This approach provides an elegant means for specifying team strategies in tasks like foraging, soccer and cooperative navigation.[6] This work is continuing as we seek to formalize and carefully analyze the various types of potentials appropriate for various multirobot tasks.

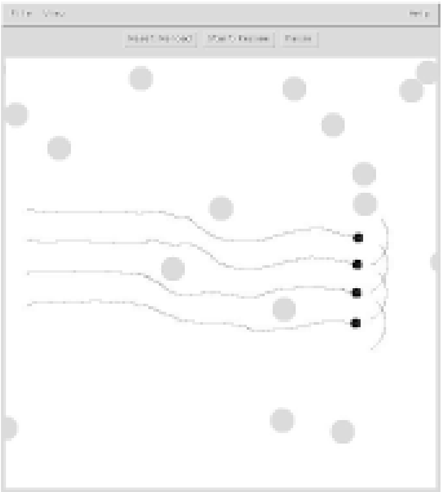


Figure 3. A formation of four simulated robots navigates using the social potentials approach, initially developed at Georgia Tech and extended at the MultiRobot Lab.

Tucker Balch
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15217
Phone: 412/268-1780
E-mail: trb@cs.cmu.edu
http://www.cs.cmu.edu/~trb

References

1. http://www.cs.cmu.edu/~multirobotlab
2. S. Lenser and M. Veloso, *Sensor Resetting Localization,* **IEEE Int'l Conf. Robotics and Automation,** 2000, *to appear*.
3. T. Balch, *Hierarchic Social Entropy: An Information Theoretic Measure of Robot Group Diversity,* **Autonomous Robots 8** (3), *to appear.*
4. http://www.cs.cmu.edu/~coral/minnow
5. http://www.teambots.org
6. T. Balch, *Social Potentials for Scalable Multi-Robot Formations,* **IEEE Int'l Conf. Robotics and Automation 2000,** *to appear.*

I-Cubes

continued from p. 4

groups of robots that can reposition themselves to form a group capable of changing its gait in order to move over obstacles that single elements cannot overtake. Similar scenarios that require reconfiguration include climbing stairs and traversing pipes.

Path planning for modular self-reconfigurable systems has proven to be difficult at best, due to the total number of spatial configurations for a given number of modules: the possibilities grow exponentially with this number. In addition to the large number of configurations, motion-planning algorithms need to be able to consider multiple sequences of these configurations in order to find the one that will lead from a given initial position/shape to a final one. We are currently testing heuristic and knowledge-based algorithms to find intelligent solutions that can be implemented in real time.

Cem Ünsal
Carnegie Mellon University
Institute for Complex Engineered Systems (ICES)
5000 Forbes Avenue
Pittsburgh, PA 15213-3890
E-mail: unsal@ri.cmu.edu
http://www.cs.cmu.edu/~unsal/research/ices/cubes.html

Reference

1. C. Ünsal, H. Kiliÿÿöte, and P.K. Khosla, *I(CES)-Cubes: A Modular SelfReconfigurable Bipartite Robotic System,* **Proc. SPIE 3839,** pp. 258-269, 1999.

Modelling modular robotics systems

Robotic systems can be created by combining the functionality of a set of separate system components such as sensors, effectors, and computational resources. The effort of the programmer in these circumstances is often spent in making explicit the inherent functionality that can emerge when these components are combined. Benefits, at the very least in terms of saved programming effort, can be achieved if an automated system can reason about these interactions. In order to achieve this, the functionality provided by the resources must be described in a manner that allows reasoning to be performed about their suitability for a particular task and the consequences of their interaction with other resources. At a more practical level, resources must also be provided with facilities that support their interconnection, including considerations of data requirements and appropriate control structures.

The MARS model, developed at the University of Reading, addresses these issues. Its chief contribution is the introduction of an explicit, declarative description of the individual resources and their target configuration in a modular robot system. A reasoning model exploits this description to identify and resolve the consequences, both beneficial and detrimental, of the interactions between the resources. MARS exploits annotations, semantic descriptions added to source code as comments, to repackage the software defining the robotics resources as 'modules'; the basic unit the model recognizes. MARS identifies physical (sensors and effectors) and non-physical (algorithmic) modules, and provides notation for describing both the modules and the physical, data, and control relationships between them. A configuration defining a modular robotics system comprises an itemized list of these relationships.

The model resolves consequences through the introduction of additional modules to produce a specification that is the basis for realizing the target robot system. It extracts the annotations from the source code and generates a module, with source code, for the corresponding modular robot system. This programming model has been implemented in an environment called DEIMOS.

The practical benefit of the approach represented by MARS is the ability to move swiftly from specification to implementation without the requirement of a programmer either to glue the components together or to articulate the inherent functions that can emerge when modules interact.

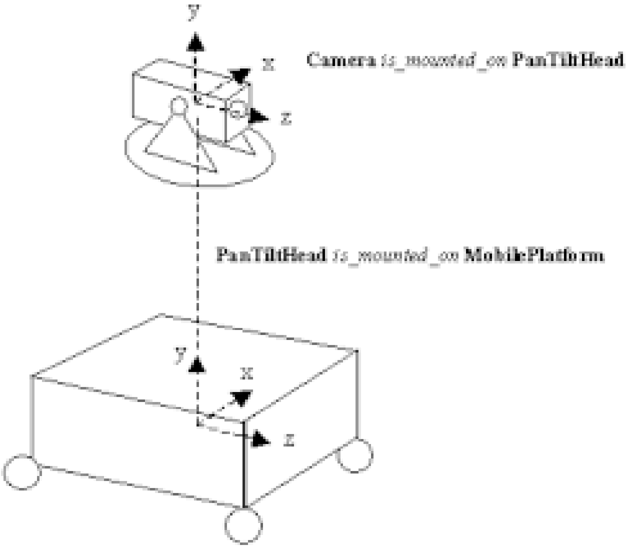


Figure 1. Modular robot system.

Exploiting object-oriented concepts

MARS exploits object-oriented concepts in its reasoning. The most important example of this is in the inheritance of motion functionality. Central to this is the notion that sensors and end-effectors have an inherent potential for motion in six degrees of freedom, even though they may not be able to express that motion directly. The MARS model allows these components—and tools in general—to inherit motion functionality from the effectors on which they are mounted. When, for example, a camera is mounted on a pan-tilt head can inherit the pan and tilt motions from the head. If the latter is in turn mounted on a mobile base, the camera can also inherit its motion functionality. This includes the inheritance of similar rotational functions from the pan-tilt head (pan) and the mobile base (rotate).

The MARS model also identifies data and control consequences associated, respectively, with distributing data to multiple client modules and arbitrating between multiple control commands converging on a single effector. These provide scope for a range of control architectures, including centralized, decentralized, hybrid and agent-based architectures.

We can illustrate functional inheritance in the MARS model using the mobile camera example above. For each of the three components, a corresponding MARS module is defined. We will denote these, respectively, as the Camera module, the PanTiltHead module and the MobilePlatform module. Each has a well-defined interface: the Camera module for grabbing images, and the PanTiltHead and MobilePlatform modules for pan-tilt motions and drive-rotate motions respectively. We assume that the vertical axes of rotation for the pan and rotate motions of the latter are coincident (Figure 1). The configuration definition for the system is then trivial, and defined as follows:

```
@configuration
@modules  Camera, PanTiltHead,
MobilePlatform
Camera  is_mounted_on  PanTiltHead
PanTiltHead  is_mounted_on
        MobilePlatform
```

The definition identifies the modules involved in the configuration and their physical relationship. The MARS reasoning model then identifies the inherited functionality available to the camera from the two modules on which it is mounted. The specification that makes this explicit is defined as follows:

```
@specification
@modules  Camera, PanTiltHead,
MobilePlatform, InheritanceNode1
Camera  is_mounted_on  PanTiltHead
PanTiltHead  is_mounted_on
    MobilePlatform
Camera   uses_motion_from
    InheritanceNode1 Rotn_X, Rotn_Y,
    Trans_Z  InheritanceNode1 {
        Rotn_X: PanTiltHead
        Rotn_Y: PanTiltHead, MobilePlatform
        Trans_Z: MobilePlatform
        resolve Rotn_Y:  UseOnly PanTiltHead
}
```

The InheritanceNode in the specification collects together the components of the functional inheritance and specifies how the rotational motions required by the camera should be resolved between the head and the base. In this case it is simply resolved in favor of the head. Other resolution strategies are possible. This specification now forms the basis, using tools within the DEIMOS software environment, for the creation of source code for the composite system. The MARS model is still in early stages of development. Further research will evaluate the scope of the model and will investigate extensions to enhance its capability to deal with diverse robotics and computational systems.

G. T. McKee and **J. A. Fryer**
The University of Reading, UK.
E-mail: G.McKee@Reading.ac.uk

References

1. J. A. Fryer and G. T. McKee, *Resource Modelling and Combination in Modular Robotics Systems,* **Proc. IEEE Int'l Conf. Robotics and Automation,** pp. 3167-3172, Leuven. Belgium, 1998.

The IRIS Laboratory: 3D imaging and data fusion at the University of Tennessee

The Imaging, Robotics, and Intelligent Systems (IRIS) Laboratory, in the University of Tennessee's Department of Electrical and Computer Engineering, performs research in 3D imaging and data fusion. Applications of this research include mapping of large facilities for robotic navigation and automated construction of virtual environments for real time simulation. IRIS Lab core technologies are classified into the three areas illustrated in Figure 1 and described below.

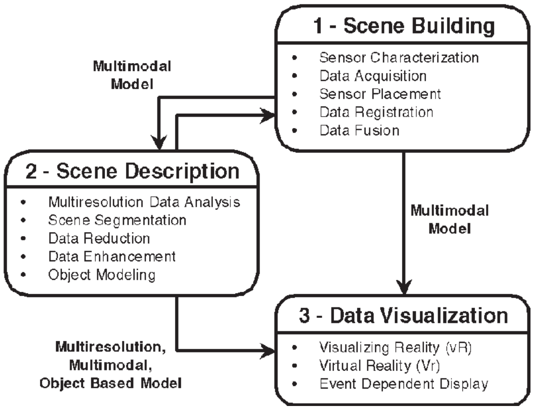


Figure 1. IRIS Lab core technologies.

Scene building

To acquire usable data, accurate sensor characterization is obviously important. In many scenarios, sensor placement must be addressed to ensure that the captured data is as accurate and complete as possible. Data sets captured from different viewpoints, perhaps with different sensor modalities, must be registered to a common coordinate system. Once registered, the data can be fused to improve the information available from a single sensor or to capture additional information, such as texture or thermal characteristics. The final outcome of this task is a dense lattice of 3D, multimodal data that describes the scene geometry and other spatial and spectral characteristics.

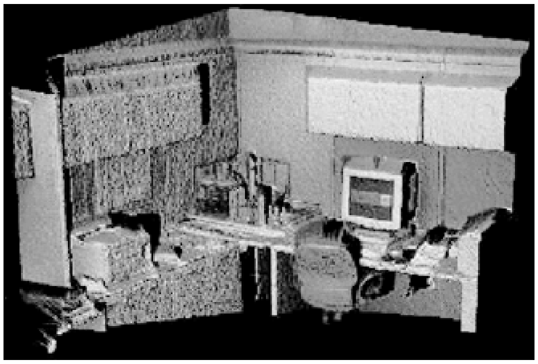


Figure 2. 3D scene automatically reconstructed from unregistered range and video.

Scene description

Once constructed, a scene generally requires further processing to be of practical use. Objects of interest might need to be segmented from other objects, clutter, and/or background. After segmentation, objects may be modeled to aid in manipulation, recognition, and/or visualization. The amount of data captured in the scene building process often exceeds the needs or capabilities of the application, thereby necessitating data reduction. On the other hand, data enhancement may be required for the examination of small details. A multiresolution analysis of the data can benefit the solution of each of these problems. Referring to Figure 1, note that information flows in both directions between Scene Building and Scene Description. In many cases, intermediate scene description results can be used to modify the operation of the scene building module.

Data visualization

Current visualization activities are focused on examining the results from scene building and scene description. As these results are often used for virtual reality (Vr) or to visualize reality (vR), research must account for varying visualization requirements. The objective is to provide appropriate data so that "event" dependent visualization can be achieved. Example events might include viewpoint, hardware limitations, constant frame rate, and/or object specific interest.

The IRIS Lab is presently supported by the U.S. Department of Energy (DOE) through the University Research Program in Robotics (URPR) and by the U.S. Army's Tank-automotive & Armaments Command (TACOM) through the National Automotive Center (NAC) and the Automotive Research Center (ARC).

Mongi A. Abidi and Jeffery R. Price
328 Ferris Hall
University of Tennessee
Knoxville, TN 37996-2100, USA
E-mail: {abidi, jrp}@utk.edu
http://www.iristown.engr.utk.edu

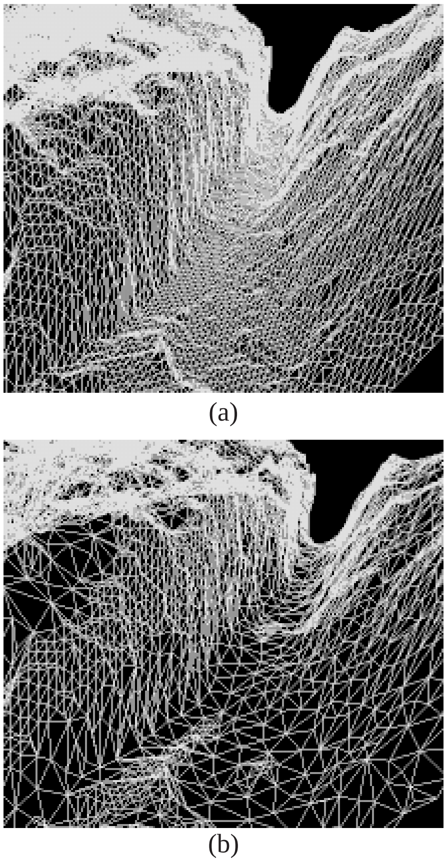

(a)
(b)

Figure 3. Mesh reduction using a multiresolution, wavelet analysis: (a) original terrain mesh; (b) mesh after reduction.

References

1. C. Gourley, C. Dumont, and M. Abidi, *Enhancement of 3D models for fixed rate interactivity during display,* **Proc. IASTED Int'l Conf. Robotics and App.,** pp. 174-179, October 1999.
2. L. Wong, C. Dumont, and M. Abidi, *Next best view system in a 3D object modeling task,* **Proc. IEEE Int'l Symp. Computational Intelligence in Robotics and Automation: CIRA'99,** pp. 306-311, November 1999.
3. F. Boughorbal, D. Page, and M. Abidi, *Automatic reconstruction of large 3D models of real environments from unregistered data sets,* **Proc. SPIE 3958,** 2000.
4. H. Gray, C. Dumont, and M. Abidi,*Integration of multiple range and intensity image pairs using a volumetric method to create textured 3D models,* **Proc. SPIE 3966,** 2000.
5. E. Juarez, C. Dumont, and M. Abidi, *Object modeling in multiple-object 3D scenes using deformable simplex meshes,* **Proc. SPIE 3958,** 2000.
6. M. Toubin, D. Page, C. Dumont, F. Truchetet, and M. Abidi, *Multi-resolution wavelet analysis for simplification and visualization of multi-textured meshes,* **Proc. SPIE 3960,** 2000.
7. Y. Zhang, Y. Sun, H. Sari-Sarraf, and M. Abidi, *Impact of intensity edge map on segmentation of noisy range images,* **Proc. SPIE 3958,** 2000.

Mobile robots: Part of a smart environment

Smart environments consist of numerous agents perceiving the environment, communicating with other systems and humans, reasoning changes to the environment, and performing those changes accordingly. A key characteristic for a system operating in a smart environment is “situatedness”: a concept that was introduced to the fields of AI and robotics in the 1980s. A situated system is in tight interaction with its environment: analyzing the situation at hand and calculating appropriate responses in timely fashion. This is the opposite of the traditional approach to intelligent systems which emphasized computationally expensive symbolic models and planning. In our research, the concept of situatedness is enlarged into the field of mobile computing. In addition to a robot, a situated system can be a mobile phone, a wearable computer, a handheld navigational aid for the elderly, or some other mobile information system. In all these mobile devices serving a user, situatedness conveys the idea that the operation of the device depends on the state of the user and the state of the local environment.

Intelligent mobile robots

The general goal of our research in this area is to develop components for the next generation of intelligent robots that will operate in our normal living environment and cooperate with human beings and other machines. These robots are an essential part of the smart environment.

We have developed a control architecture for a mobile robot operating in a dynamic environment. An early version of the control system, PEMM, was applied to the control of an intelligent and skilled paper roll manipulator.[1,2] The research resulted in a practical, full-sized implementation capable of handling paper rolls in warehouses and harbors. A key characteristic of the more recent version of the control architecture, Samba, is its ability to both reason about actions based on task constraints and to react quickly to unexpected events in the environment.[3,4] That is, the architecture is goal-oriented and situated.

Samba

Samba continuously produces reactions to all the important objects in the environment. These reactions are represented as action maps. An action map describes, for each possible action, how advantageous it is from the perspective of behaving towards the object in a certain way (e.g. avoiding or chasing). The preferences are shown by assigning a weight to each action. By combining action maps, an action producing a satisfactory reaction to several objects simultaneously can be found. Figure 1 shows two robots and an action map for the robot in front to catch the ball. The ridge describes the actions by which the robot reaches the ball. The actions that would result in a collision with the other robot before the ball is reached are inhibited.

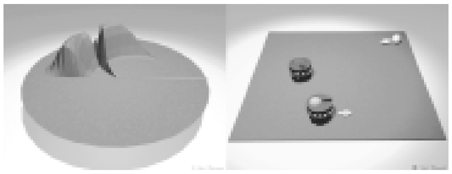


Figure 1. An action map and two robots catching the ball.

Recently, our main activities have been in developing the Samba control architecture and techniques for vision-based environment modeling. We have applied the control architecture in playing simulated soccer, and participated the first and second world championships of robotic soccer, RoboCup, held in Nagoya in 1997 and Paris in 1998.[5,6] We are also trying to build a mobile robot to support the independent living of the elderly and disabled. In this work, we have demonstrated the use of a teleoperated mobile robot for domestic help.

Vision-based environment modeling

In vision-based environment modeling, a technique is being developed that combines 3D information provided by structured lighting with the 2D information provided by intensity images.[7] As an application, the creation of the model of a building for virtual reality applications is considered. This model is created while a robot is moving inside the target building. Because the model will have 3D range and 2D (color) intensity information, it is possible to include versatile information about the structures inside the building such as surface colors and materials. Figure 2 shows an image taken by the color range scanner developed in our laboratory. This image is combination of range and reflectance images.

Interacting with mobile robots and smart environments

In developing a mobile robot capable of executing complex tasks in a dynamic environment, we selected the approach of first building a teleoperated robot and then gradually shifting tasks from the human to the robot. This approach enables complex tasks to be performed as soon as the teleoperation is operational. This speeds up the research on human-robot interaction, as complex interactions can be analyzed at an early stage of the research. Furthermore, in many applications, human operators can be left in the control loop to analyze the data collected by the robot or to solve tasks that are too difficult for it.

In the future, interaction between humans, mobile robots and other devices will become increasingly important as machines will, more and more, operate as assistants in our everyday life. Because of this, there is a need for a general-purpose technique that we can use to interact with robots and other embedded systems in smart environments. We have developed a technique based on a mobile code paradigm, that allows humans to use a single handheld control device to interact with mobile robots and ubiquitous embedded systems.[8]

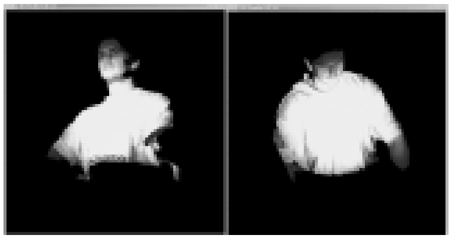


Figure 2. A combination of range and reflectance images.

Juha Röning and **Jukka Riekki**
Infotech Oulu and Department of Electrical Engineering
University of Oulu
PO Box 4500
FIN-90014, Finland
E-mail: Juha.Roning@oulu.fi
http://www.ee.oulu.fi/research/isg/

References

1. T. Heikkilä and J. Röning, *PEM-Modelling: A Framework for Designing Intelligent Robot Control,* **J. Robotics and Mechatronics 4** (5), pp. 437444, 1992.
2. P. Vähä, T. Heikkilä, J. Röning, and J. Okkonen, *Machine of the Future: An Intelligent Paper Roll Manipulator,* **Mechatronics: Mechanics Electronics Control 4** (8), pp. 861-877, 1994.
3. J. Riekki and J. Röning, *Reactive Task Execution by Combining Action Maps,* **IEEE/RSJ Int’l Conf. on Intelligent Robots and Systems: IROS’97,** pp. 224-230, Grenoble, France, 8-12 September 1997.
4. J. Riekki, *Reactive task execution of a mobile robot,* **Acta Univ. Oul. C 129** (PhD Thesis), University of Oulu, Oulu, Finland, 1999. http://herkules.oulu.fi/isbn9514251318/
5. J. Riekki and J. Röning, *Playing Soccer by Modifying and Combining Primitive Reactions,* **RoboCup-97: Robot Soccer World Cup I, Lecture Notes in Artificial Intelligence 1395,** H. Kitano (ed.), pp. 74-87, Springer-Verlag, Nagoya, 1997.
6. J. Riekki, J. Pajala, A. Tikanmäki, and J. Röning, *CAT Finland: Executing primitive tasks in parallel,* **RoboCup-98: Robot Soccer World Cup II, Lecture Notes in Artificial Intelligence 1604,** M. Asada and H. Kitano (eds.), Springer-Verlag, pp. 396-401, 1999.
7. J. Haverinen and J. Röning, *A 3-D Scanner Capturing Range and Color,* **Proc. Scandinavian Symposium on Robotics,** pp. 144-147, Oulu, Finland, 14-15 October 1999.
8. J. Röning and K. Kangas, *Controlling Personal Robots and Smart Environments,* **Proc. SPIE 3837,** pp. 26-39, 1999.

Join the Technical Group

...and receive this newsletter

Membership Application

Please Print ☐ Prof. ☐ Dr. ☐ Mr. ☐ Miss ☐ Mrs. ☐ Ms.

First Name, Middle Initial, Last Name ____________________

Position ____________________ SPIE Member Number ____________________

Business Affiliation ____________________

Dept./Bldg./Mail Stop/etc. ____________________

Street Address or P.O. Box ____________________

City/State ____________________ Zip/Postal Code ____________ Country ____________

Telephone ____________________ Telefax ____________________

E-mail Address/Network ____________________

Technical Group Membership fee is $30/year, or $15/year for full SPIE members.

☐ Robotics & Machine Perception
Total amount enclosed for Technical Group membership $ ____________

☐ **Check enclosed.** Payment in U.S. dollars (by draft on a U.S. bank, or international money order) is required. Do not send currency. Transfers from banks must include a copy of the transfer order.

☐ **Charge to my:** ☐ VISA ☐ MasterCard ☐ American Express ☐ Diners Club ☐ Discover

Account # ____________________ Expiration date ____________________

Signature ____________________
(required for credit card orders)

This newsletter is printed as a benefit of membership in the **Robotics & Machine Perception Technical Group.** Technical group membership allows you to communicate and network with colleagues worldwide.

Technical group member benefits include a semi-annual copy of the *Robotics & Machine Perception* newsletter, SPIE's monthly newspaper, *OE Reports*, membership directory, and discounts on SPIE conferences and short courses, books, and other selected publications related to robotics.

SPIE members are invited to join for the reduced fee of $15. If you are not a member of SPIE, the annual membership fee of $30 will cover all technical group membership services. For complete information and an application form, contact SPIE.

Send this form (or photocopy) to:
SPIE • P.O. Box 10
Bellingham, WA 98227-0010 USA
Phone: (1) 360/676-3290
Fax: (1) 360/647-1445
E-mail: info-robo@spie.org
Anonymous FTP: spie.org

Please send me
☐ Information about full SPIE membership
☐ Information about other SPIE technical groups
☐ FREE technical publications catalog

ROBOTICSONLINE

info-robo@spie.org

Here's an easy way to reach your colleagues around the world—instantly. SPIE's **Robotics Listserv** is an automated e-mail server that connects you to a network of engineers, scientists, vendors, entrepreneurs, and service providers.

SPIE's Robotics Listserv has its roots in the well-established and active Robotics & Machine Perception Technical Group, and is yet another way to use SPIE's technical resources.

To join the Robotics Listserv, send an e-mail message to `info-robo-request@spie.org` with the words `subscribe info-robo` in the message body.

For detailed instructions, as well as information about other online SPIE services, such as Abstracts Online, Employment Services, and Technical Programs Online, send a message to `info-spie-request@spie.org` with the word `help` in the message body.

Robotics & Machine Perception

The Robotics & Machine Perception newsletter is published semi-annually by SPIE—The International Society for Optical Engineering for its International Technical Group on Robotics & Machine Perception.

Technical Group Chair Gerard McKee
Technical Editor Sunny Bains
Managing Editor Linda DeLano
Advertising Sales Bonnie Peterson

SPIE is an international technical society dedicated to advancing engineering, scientific, and commercial applications of optical, photonic, imaging, electronic, and optoelectronic technologies. Its members are engineers, scientists, and users interested in the development and reduction to practice of these technologies. SPIE provides the means for communicating new developments and applications information to the engineering, scientific, and user communities through its publications, symposia, education programs, and online electronic information services.

SPIE—The International Society for Optical Engineering, P.O. Box 10, Bellingham, WA 98227-0010 USA. Phone: (1) 360/676-3290. Fax: (1) 360/647-1445.

European Office: Contact SPIE International Headquarters.

In Japan: c/o O.T.O. Research Corp., Takeuchi Bldg. 1-34-12 Takatanobaba, Shinjuku-ku, Tokyo 160, Japan. Phone: (81 3) 3208-7821. Fax: (81 3) 3200-2889. E-mail: otoresco@gol.com

In Russia/FSU: 12, Mokhovaja str., 121019, Moscow, Russia. Phone/Fax: (095) 202-1079. E-mail: edmund@spierus.msk.su.

Want it published? Submit work, information, or announcements for publication in future issues of this newsletter to Sunny Bains at sunny@spie.org or at the address above. Articles should be 600 to 800 words long. Please supply black-and-white photographs or line art when possible. Permission from all copyright holders for each figure and details of any previous publication should be included. All materials are subject to approval and may be edited. Please include the name and number of someone who can answer technical questions. Calendar listings should be sent at least eight months prior to the event.

The CMU MultiRobot Lab

In the MultiRobot Lab at CMU, we are interested in building and studying teams of robots that operate in dynamic and uncertain environments.[1] Our research focuses specifically on issues of multiagent communication, cooperation, sensing and learning. We experiment with and test our theories in simulation and on a number of real robot testbeds.

Cooperative multirobot behaviors and localization

Planning for real robots to act in dynamic and uncertain environments is a challenging problem. A complete model of the world is not viable and an integration of deliberation and behavior-based control is most appropriate for goal achievement and uncertainty handling. In recent work, we have successfully integrated perception, planning, and action for Sony quadruped legged robots (Figure 1).[2]

The quadruped legged robots running our software are fully autonomous with onboard vision, localization and agent behavior. Our perception algorithm automatically categorizes objects by color in real time. The output of the image processing step is provided to our Sensor Resetting Localization (SRL) algorithm (an extension of Monte Carlo Localization). SRL is robust to movement modelling errors and to limited computational power. The system can estimate the position of a robot on a 2m by 4m field within 5cm.

Our team of robots were entered in the RoboCup-99 robot soccer world championship. They only lost one game and placed third overall.

Behavioral diversity

Behavioral diversity provides an effective means for robots on a team to divide a task between them; robots can specialize by assuming different roles on the team. In earlier work we developed quantitative measure of the extent of behavioral diversity in a team and we have used this measure in experimental evaluations of robot groups.[3] One result of our research in behavioral diversity is data that indicates the usefulness of diversity depends on the team task. For instance, in soccer simulations, teams using heterogeneous behaviors perform best, while in robot foraging experiments, groups using homogeneous behaviors work best. We are continuing this work by investigating the appropriate level of diversity for various tasks and hardware platforms.

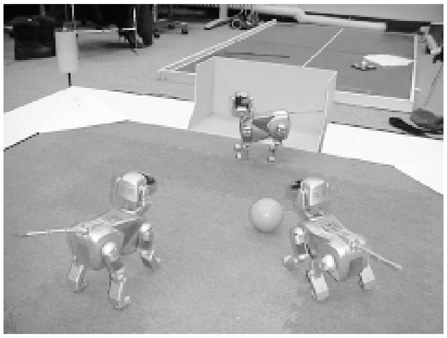


Figure 1. Sony quadruped robots. Robust vision-based localization software has been developed for these robots at the MultiRobotLab.

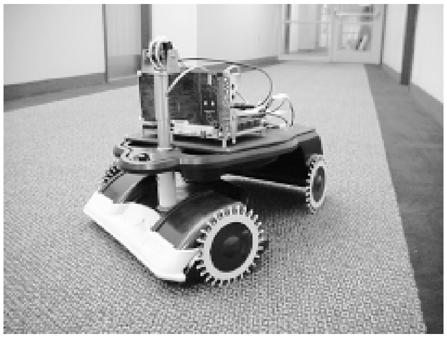
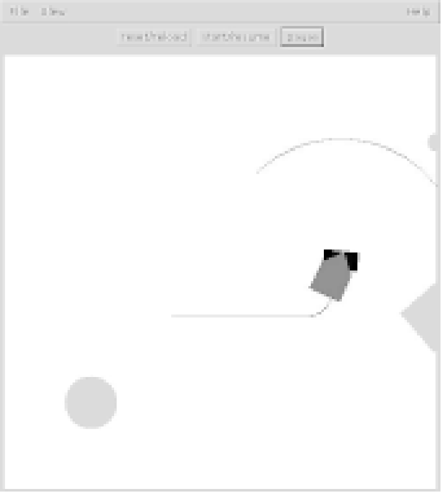


Figure 2. The low-cost Minnow robot platform (above). A simulation of the Minnow (right). ⟶

Hardware and software for multirobot research

Because we are interested in the behavior of large numbers of cooperating robots it is important to address the scalability of multirobot systems. Robot cost and reliability, and software portability and scalability, are crucial to the goal of scalable multirobot systems. To reduce robot cost and to improve reliability we have designed a low-cost mobile robot platform using only commercial-off-theshelf components (Figure 2). The Minnow robot is a fully autonomous indoor vehicle equipped with color vision, a Linux-based computer with hard disk, and wireless Ethernet. The entire robot costs less than $3000.[4] This low cost-perrobot will enable us to scale up to five or ten robots over the next year.

Of course scalable robot software is just as important as hardware. To provide for software reuse and portability we have developed TeamBots, our multirobot development environment, in Java.[5] To our knowledge, this is the first substantial robot control platform written in Java. One of the most important features of the TeamBots environment is that it supports prototyping in simulation of the same control systems that can be run on mobile robots. This is especially important for multirobot systems research because debugging, or even running, a multirobot system is often a great challenge. A simulation prototyping environment can be a great help.



The TeamBots simulation environment is extremely flexible. It supports multiple heterogeneous robot hardware running heterogeneous control systems. Complex (or simple) experimental environments can be designed with

continued on p. 7

SPIE Society of Photo-Optical Instrumentation Engineers

P.O. Box 10 • Bellingham, WA 98227-0010 USA

Change Service Requested

DATED MATERIAL

Non-Profit Org.
U.S. Postage Paid
Society of
Photo-Optical
Instrumentation
Engineers