MVA '98 IAPR Workshop on Machine Vision Applications, Nov. 17-19, 1998, Makuhari, Chiba, Japan

2—4 Integrated Vision/Control System for Autonomous Planetary Rovers

Terry Huntsberger, Toshiro Kubota, and John Rose

Intelligent Systems Laboratory, Department of Computer Science
University of South Carolina, Columbia, SC 29208
terry@cs.sc.edu, kubota@cs.sc.edu, rose@cs.sc.edu

Abstract

Current NASA plans call for extended year-long, multikilometer treks for the 2003 and 2005 Mars missions. A much greater amount of rover autonomy is required compared to the recent Sojourner mission, where the rover stayed within a 50 meter radius of the Pathfinder lander. A 2005 mission prototype system of a Sample Return Rover (SRR1) is currently being field tested at the Jet Propulsion Laboratory in Pasadena, CA USA.

BISMARC (**B**iologically **I**nspired **S**ystem for **M**ap-based **A**utonomous **R**over **C**ontrol) is a hybrid wavelet/neural network based system that is capable of such autonomy. Previous simulations demonstrated that the system is capable of control for multiple rovers involved in a multiple cache recovery scenario. This robust behavior was obtained through the use of a free-flow hierarchy (FFH) as an action selection mechanism.

A subsequent study extended BISMARC to include fault tolerance in the sensing and mechanical rover subsystems. The vision subsystem in the original BISMARC implementation relied on the generalization capabilities of a fuzzy self-organizing feature map (FSOFM) neural network. This paper describes a more robust vision subsystem based on camera models combined with tilt sensors that is fully integrated into the BISMARC framework. The results of preliminary simulation studies in a Mars environment are also reported.

1 Introduction

We have recently developed a control system for autonomous planetary rover control (shown in Figure 1) called BISMARC (**B**iologically **I**nspired **S**ystem for **M**ap-based **A**utonomous **R**over **C**ontrol) [2, 5, 3], that uses a hybrid neural network [4]/free flow hierarchy [11] architecture for action selection. BISMARC used the output of the FSOFM neural network to determine the possible set of actions for any given visual sensor inputs. During the course of simulation studies, failure modes occurred for highly uneven planetary terrain, where steep cliffs and deep chasms are present. Since the network was trained with no sense of depth or changes in elevation, rovers tended to fall over the edge of a cliff.

This paper presents a modification of the vision subsystem in BISMARC for integration of depth and surface slope information into the action generation process. It was felt that this would be the most robust solution to the problem, since the number of possible training scenarios for cliff avoidance would be quite large. The next section describes the overall organization of the BISMARC architecture. This is followed by a discussion of the vision subsystem and its interactions with the action generation system. The results of some failure mode scenarios in simulated planetary rover missions is described next, followed by a final summary section.

2 BISMARC Organization

The original three level BISMARC architecture uses a hybrid mix of neural networks and behavior-based approaches. The first level performs a wavelet transform on the rover's stereo image pair, the second level inputs these processed images into an action generation navigation network, and then to a third level action selection mechanism (ASM) network modeled after that of the Rosenblatt and Payton FFH [11]. This type of FFH was recently shown to be optimal within the multiple objective decision making (MODM) formalism, which produces an action through the maximization of a global objective function that includes all possible actions [10]. Some examples of the other external inputs would include internal temperature sensing, internal tilt sensing,

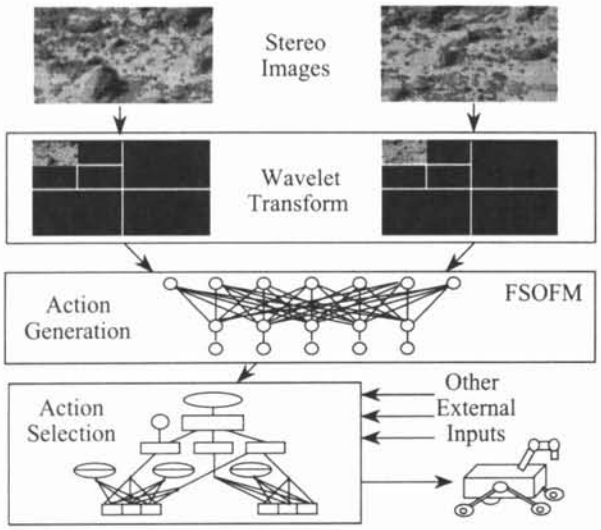

Figure 1: Multilevel system organization of BISMARC for autonomous rover control in planetary environments. Coefficients from the wavelet detail channels are used to generate actions with a FSOFM. An ASM then performs a combination operation on the possible actions for final navigation.

relative time of day, sun sensor positioning, and possible communications with other rovers.

There are a number of factors beyond the visual sensory input that influence navigation of the rover. These include the health of the rover (internal/external temperature, battery power levels, accelerometers/gyro), time of day, and homogeneity of the terrain. For example, if the rover is approaching a highly uneven portion of the terrain rather late in the day, the control decision may be made to halt and wait for the next morning in order to recharge the batteries and to have enough light for visual sensing. The FFH automatically handles action selection in the presence of such conflicting behaviors.

A FFH is a directed graph of action and stimulus nodes that are combined using predetermined rules. These rules may include addition, multiplication, or more complicated means of combination. The FFH system for BISMARC is shown in Figure 2. Action nodes are drawn as rectangles, stimulus nodes as ellipses, and those with multi-directional characteristics are indicated using 8 directional bins. The combination rules are additive for a small filled rectangle above the node, multiplicative for a small filled triangle, and a more sophisticated rule is used for plain rectangular nodes [12].

Tyrrell introduced the temporal penalty (T-circle in Figure 2) to control action that will take an inordinate amount of time to complete [12]. Temporal penalty nodes increased the likelihood of satisfying the overall mission goal of maximizing the number of cache containers returned. In addition, the uncertainty penalty (U-circle in Figure 2) is used to control actions that are heavily dependent on external sensor inputs, which are usually noisy and imprecise.

The top level nodes in BISMARC generally relate to rover health (Avoid Dangerous Places, Sleep at Night, Warm Up, Cool Down, Get Power) or cache recovery/navigation (Scan for Cache, Get Cache, Keep Variance Low). These high level actions involve a complicated combination of internal control and assimilation of external sensor inputs. There are only five bottom level nodes: Sleep, Approach, Move, Look Around, and Rest. The bottom level node that is most sensitive to limitations in the visual subsystem is Move, since its activation level depends primarily on the combination of visual sensor inputs with additional input from higher levels in the hierarchy.

The direct integration of the obstacle avoidance behavior into BISMARC's route planning strategy is a necessary component of the design. Since obstacles are used as *landmarks* for sensory/action map making, detailed information about size, relative height, etc. are important. The necessary clearance for a rover to navigate around an obstacle is built into the training sets, and looming is used to differentiate between small obstacles that can be driven over and the larger ones that require course changes. Unfortunately, there is no way to adequately train the FSOFM network to include drastic changes in surface slope, since relative depth information is only crudely present in the looming phenomena. There are however tilt sensors within the rover that do contain this information, and integrated visual/tilt mechanism can be directly included in BISMARC through the action selection mechanism.

3 Vision Subsystem

Since the FSOFM is providing obstacle avoidance information based on the input of the stereo cameras, knowing the camera model will enable the recovery of relative depth information and error bounds [6, 8]. Dangerous places are areas where relatively large gradients in the planetary surface are present. Temporal integration of tilt sensor inputs can be used in these situations, but for a very sharp drop-off such as a cliff, the needed action selection inputs may be too late to prevent damage to the rover. However, two recent studies have demonstrated relatively good detection capabilities for such a case using only stereo camera inputs [7, 9]. The stereo algorithm used in these studies is the one currently implemented on the SRR1 prototype at JPL.

The stereo system has a 5 centimeter baseline with 130 degree FOV lenses, and uses grayscale images of 512×486 spatial resolution. A ten step pro-

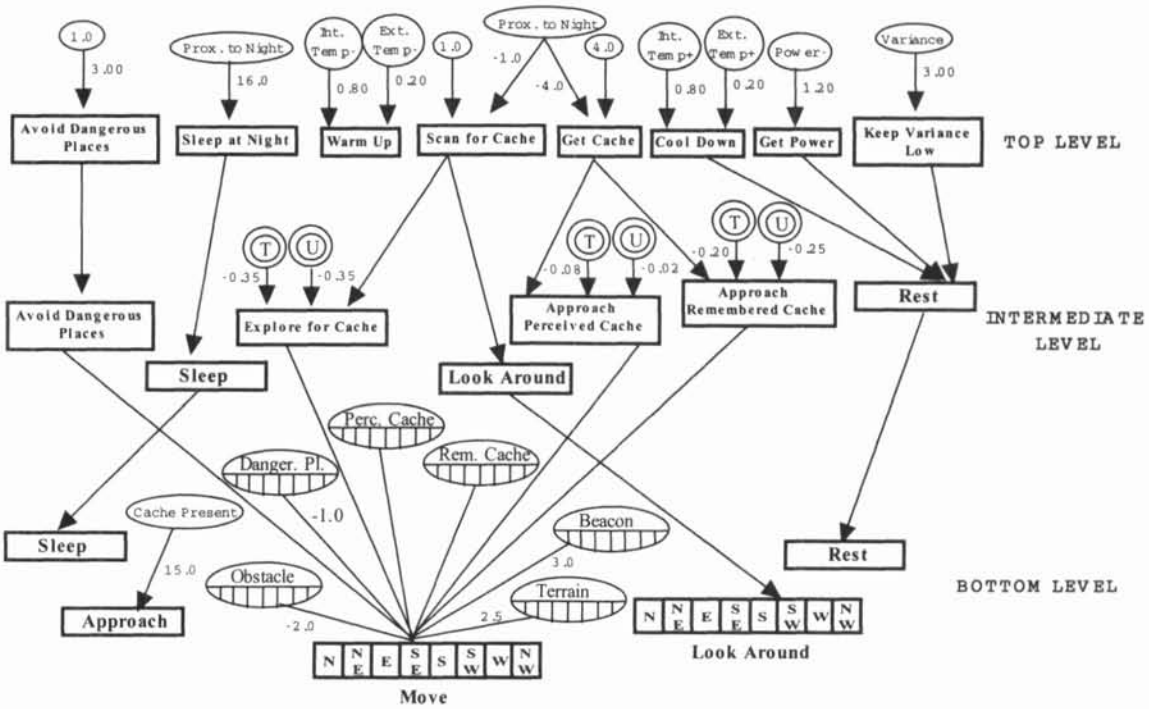

Figure 2: FFH system for BISMARC. All weights on the arcs are 1.0 unless otherwise indicated. Notation for symbols is that of Tyrrell [12]. See text for a detailed discussion.

cess is used to determine the range map and perform obstacle avoidance for 50 centimeters in front of the rover, taking about 5 seconds on the current 100 Mhz 80486 processor. This limits the forward speed of the rover to a maximum of 10 cm/sec for safety sake. A recent study performed in the Planetary Robotics Lab at JPL demonstrated that the algorithm gives good rover localization when combined with extended Kalman filtering of odometry data [1]. The FSOFM in BISMARC has been replaced with this improved sensor model. The tradeoff here is a more computationally intensive front-end to BISMARC in return for better sensory warning of potentially destructive terrain features.

4 Experimental Study

Here we report the results of some preliminary simulation studies. We ran four trials using a randomly generated heightfield with some steep 500 meter drops in elevation. The area encompassed about 1KM by 1KM with a grid decomposition resolution of 5 cm. Each trial put the rover within 50 meters of a steep incline. The goal was to return to the start position with a cache container whose placement site was known to within a 200 meter radius using a beacon. The top speed on the rover was set at 10 cm/sec, which is consistent with SRR1. In order to simulate wheel slippage, we set a 10% loss of traction when climbing over a rock or traversing rocky terrain. The battery lifetime was set at one week on the rover with a timestep of 0.1 sec. The rover was forced to sleep during the night hours of the simulations, since there were no infrared sensors on the rover. Cache acquisition time using the 5 DOF manipulator on the rover prior to return to the lander site was assumed to be one hour.

The rover succeeded in avoiding the steep incline in all four of the trials. The paths that were followed to the cache containers for the trials are shown in Figure 3, where the cache container position is shown as a circle, and the cliff area is indicated using a hatched pattern. The rovers followed the beacon until the action selection mechanism indicated that a dangerous place/obstacle (cliff) was in close proximity. Mission 4 is particularly notable in that the rover had to traverse a relatively narrow bridge of land with a cliff on either side.

5 Conclusions

This paper modified the rover control system called BISMARC to include a better sensory model for action generation. The FFH used for action se-

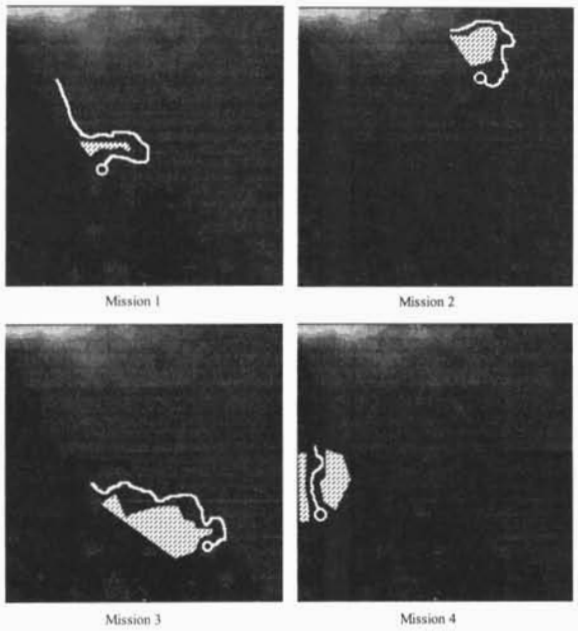

Figure 3: Rover paths for four simulation studies with close proximity to cliff. Cache position is shown as a circle and cliff is shown as hatched area.

lection in BISMARC maintained its structure and weights, but the sensor perception models now were capable of detecting steep drops in terrain elevation that correspond to cliffs. The preliminary results of four missions indicate that the rover now successfully avoids these types of obstacles and completes the cache retrieval operation. We are currently performing more simulation studies and hope to test the improved version of BISMARC on the SRR1 prototype at JPL.

6 Acknowledgments

This research was supported in part under ONR Grant N00014-97-1-1163 and ARO Grant DAAH04-96-10326. The first author would like to thank Paul Schenker for hosting me under the ASEE Visiting Faculty Program at the NASA JPL Planetary Robotics Laboratory during the summer of 1997 and for numerous visits since then.

References

[1] B. D. Hoffman, E. T. Baumgartner, T. Huntsberger, and P. S. Schenker. Improved rover state estimation in challenging terrain. *Autonomous Robots*, 1998. in press.

[2] T. L. Huntsberger. Autonomous multirover system for complex planetary retrieval operations. In P. S. Schenker and G. T. McKee, editors, *Proc. SPIE Symposium on Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, pages 221–227, Pittsburgh, PA, October 1997.

[3] T. L. Huntsberger. Fault tolerant action selection for planetary rover control. In P. S. Schenker and G. T. McKee, editors, *Proc. SPIE Symposium on Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, Boston, MA, November 1998. to appear.

[4] T. L. Huntsberger and P. Ajjimarangsee. Parallel self-organizing feature maps for unsupervised pattern recognition. *International Journal of General Systems*, 16:357–372, 1990. Also reprinted in *Fuzzy Models for Pattern Recognition*, (Eds. J. C. Bezdek, S. K. Pal), IEEE Press, Piscataway, NJ, 1992, pp. 483–495.

[5] T. L. Huntsberger and J. Rose. BISMARC: A Biologically Inspired System for Map-based Autonmous Rover Control. *Neural Networks*, 1998. in press.

[6] L. Matthies. Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation. *Intern. J. Computer Vision*, 8(1):71–91, July 1992.

[7] L. Matthies and E. Brown. Machine vision for obstacle detection and ordnance recognition. In *Proc. 1996 Annual meeting of the Assocation for Unmanned Vehicle Systems (AUVSI'96)*, Orlando, FL, July 1996.

[8] L. Matthies and P. Grandjean. Stochastic performance modeling and evaluation of obstacle detectability with imaging range sensors. *IEEE Transactions on Robotics and Automation, Special Issue on Perception-based Real World Navigation*, 10(6), December 1994.

[9] L. Matthies, A. Kelly, T. Litwin, and G. Tharp. Obstacle detection for unmanned ground vehicles: A progress report. In *Robotics Research 7*. Springer-Verlag, 1996.

[10] P. Pirjanian and H. I. Christensen. Behavior coordination using multiple-objective decision making. In P. S. Schenker and G. T. McKee, editors, *Proc. SPIE Symposium on Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, pages 78–89, Pittsburgh, PA, October 1997.

[11] J. K. Rosenblatt and D. W. Payton. A fine-grained alternative to the subsumption architecture for mobile robot control. In *Proc. IEEE/INNS Joint Conf. on Neural Networks*, pages 317–324, Washington, DC, June 1989.

[12] T. Tyrrell. The use of hierarchies for action selection. *Journal of Adaptive Behavior*, 1(4), 1993.