Alonzo Kelly, Ben Brown, Paul Klarer, Wendy Amai, Yasutake Fuke, and Luc Robert.

References

[1] R. Chatila, R. Alami, *et al*. Planet Exploration by Robots: From Mission Planning to Autonomous Navigation. In *Proc. Intl. Conf. on Advanced Robotics*, Tokyo, Japan, Nov. 1993.

[2] J. Garvey, A Russian-American Planetary Rover Initiative, AIAA 93-4088, Huntsville AL, Sept. 1993.

[3] E. Gat, R. Desai, *et al.* Behavior Control for Robotic Exploration of Planetary Surfaces. *IEEE Transactions on Robotics and Automation*, **10:4**, Aug. 1994.

[4] B. Hotz, Z. Zhang and P. Fua. Incremental Construction of Local DEM for an Autonomous Planetary Rover. In *Proc. Workshop on Computer Vision for Space Applications,* Antibes France, Sept. 1993.

[5] L. Katragadda, J. Murphy and W. Whittaker. Rover Configuration for Entertainment-Based Lunar Excursion. In *Intl. Lunar Exploration Conference*, San Diego, CA, Nov. 1994.

[6] A. Kelly. A Partial Analysis of the High Speed Autonomous Navigation Problem. Tech Report CMU-RI-TR-94-16. Robotics Institute, Carnegie Mellon University, 1994.

[7] E. Krotkov and M. Hebert, Mapping and Positioning for a Prototype Lunar Rover. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, Nagoya, Japan, May 1995.

[8] J. Purvis and P. Klarer. RATLER: Robotic All Terrain Lunar Exploration Rover. In *Proc. Sixth Annual Space Operations, Applications and Research Symposium*, Johnson Space Center, Houston TX, 1992.

[9] L. Robert, M. Buffa and M. Hebert. Weakly-Calibrated Stereo Perception for Rover Navigation. In *Proc. Image Understanding Workshop*, 1994.

[10] J. Rosenblatt and C. Thorpe, Combining Multiple Goals in a Behavior-Based Architecture, In *Proc. IEEE Conference on Intelligent Robots and Systems,* Pittsburgh PA, August 1995.

[11] R. Simmons. Structured Control for Autonomous Robots. *IEEE Transactions on Robotics and Automation*, **10:1**, Feb. 1994.

[12] B. Wilcox, L. Matthies, *et al.* Robotic Vehicles for Planetary Exploration. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, Nice France, May 1992.

the reliability and range of performance of the navigation system.

Most of the experiments were performed at a slag heap in Pittsburgh (Figure 8), on an undulating plateau featuring some sheer cliffs and sculpted features (mounds and ridges). While most of the experimental runs have been on the order of one to two hundred meters each, our longest contiguous run to date has been 1,078 m, where 94% of the distance was traversed in autonomous mode and the rest in direct teleoperation mode. Direct teleoperation is needed mainly to turn the rover around when it nears the limit of its radio transmission range, and to back the rover out of situations where it becomes trapped (since the current obstacle avoidance planner looks only several meters out, and cannot generate recommendations for traveling backwards).

The cycle time for stereo is about 1 second on a SPARC 10, and cycle time for the obstacle avoidance planner is 0.5 second (other computation times are minimal). Since the largest latency is in acquiring and transmitting image pairs (about 2 seconds), we are investigating using wireless Ethernet to speed this up. The overall cycle time, in which perception and planning is concurrent, is about 3 seconds. Average rover speed is between 10 to 20 cm/s. We are working to speed up the computations, in order to increase average speed to about 50 cm/s, the nominal speed for the anticipated 1000 km Lunar mission.

The experiments have revealed a number of areas for improvement. The most critical is that the obstacle avoidance planner must be less sensitive to noise and missing data in the stereo terrain map. We are developing a statistical approach to evaluating the traversability of paths, which should be more stable than the purely geometrical approach currently being used.

Another important improvement is to add proximity sensing, especially to detect drop-offs (cliffs and craters) in the area one to two meters in front of the rover. While the stereo-based navigation is used to steer the rover around obstacles (four to seven meters ahead), the proximity-based algorithms would halt the rover when imminently hazardous situations are detected. We have recently acquired a laser scanner for this purpose, and are working to develop simple algorithms to interpret the data reliably.

Finally, we need to increase the stereo field of view. The current two-camera system is insufficient for making sharp turns, since it cannot view much to the sides of the robot. Using lenses with a wider field of view is not attractive because of the distortion produced in the images; putting the cameras on a pan mechanism adds too much complexity. Instead, to solve this problem we will use four cameras, one pair facing left and the other pair facing right, and have the stereo component alternate between pairs of images.

With these improvements to the system, we expect to be able to travel on the order of 10 km in Lunar-relevant terrain, using the safeguarded teleoperation mode. In addition, we are working with a social scientist to design an experiment to test the effects of safeguarding on remote, time-delayed teleoperation. The idea is to quantify the objective effects (e.g., time to complete a task, number of backups required) and subjective effects (e.g., fatigue and frustration) that result from adding safeguarding techniques that veto or alter the operator's steering recommendations. While we intuitively expect that safeguarding is more and more useful as the time delay grows, we feel that it is important to measure those effects rigorously before proceeding further along this research path.

Conclusions

It is inevitable that we will return to the Moon -- probably with robots leading the way. To navigate reliably and safely over long distances, various control strategies will be needed: direct teleoperation, autonomous navigation, and safeguarded teleoperation. This paper has presented an implemented software and hardware rover system that can produce the various navigation modes by judiciously combining the steering recommendations of a human operator with those of stereo-based navigation software. This arbitration scheme provides for great flexibility in controlling the rover, as evidenced by our successful experiments in driving in outdoor, natural terrain.

Our experiments have demonstrated basic competence in driving, but much more work needs to be done in order to produce a system that can behave reliably over many weeks and kilometers. In particular, we have targeted the area of proximity sensing and obstacle detection in order to reach our goal of a 10 km traverse in 1995.

It is important to realize that safeguarding and autonomous navigation can have profound impact on the ease and reliability of remote driving of a lunar rover. On the other hand, such systems admittedly add complexity to the hardware and software requirements of a rover. We need to perform careful experiments to quantify the value added by these technologies, in order to demonstrate their effectiveness for near-term lunar missions.

Acknowledgments

This research was partially sponsored by NASA, under grants NAGW-3863 and NAGW-1175. We gratefully acknowledge assistance from Lalit Katragadda,

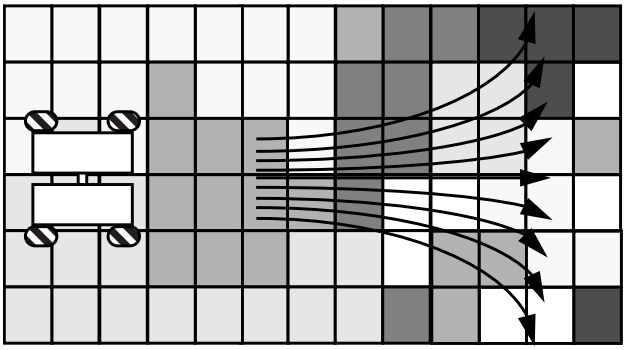

Figure 7: Evaluating Potential Steering Directions

computed using a normalized correlation. Disparity resolution is increased by interpolating the correlation values of the two closest disparities. The normalized correlation method is relatively robust with respect to differences in exposure between the two images, and can be used to produce confidence measures in the disparity values.

Much of the research effort for the stereo component has been in minimizing the number of outlier values (caused by false stereo matches). We use several methods to achieve the level of reliability required for navigation [7]. One method eliminates low-textured areas using lower bounds on the acceptable correlation values and variance in pixel intensity. Another method eliminates ambiguous matches (caused by occlusion boundaries or repetitive patterns) by rejecting matches that are not significantly better than other potential matches. Finally, the values are smoothed to reduce the effect of noise. All these methods help to produce elevation maps that accurately reflect the actual surrounding terrain, with only a few centimeters of error.

Obstacle Avoidance Planner

To decide where it is safe to drive, we have adapted techniques developed in ARPA's Unmanned Ground Vehicle (UGV) program for cross-country navigation [6]. The basic idea is to evaluate the hazards along a discrete number of paths (corresponding to a set of steering commands) that the rover could possibly follow in the next few seconds of travel. The evaluation produces a set of "votes" for each path/steering angle, including "vetoes" for paths that are deemed too hazardous, that are then sent to the arbiter to be combined with the human operator's recommendations.

The obstacle avoidance planner first merges individual elevation maps produced by the stereo system to produce a 25 cm resolution grid map up to seven meters in front of the rover. Map merging is necessary because the limited fields

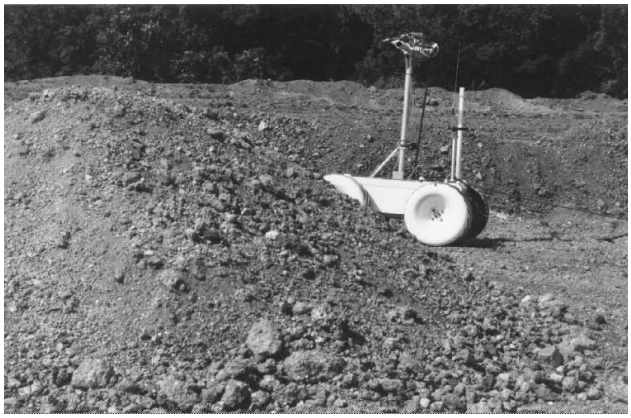

Figure 8: The Ratler at the Pittsburgh Slag Heap

of view of the cameras do not allow a single image to view sufficient terrain. Currently, we use a rather simple approach that transforms the new map based on the average deviation of elevations between the new and old maps.

To speed up the overall system cycle time, the planner requests only a small segment of the stereo image, at reduced resolution (skipping rows and columns in the image). Experiments show that only about 2% of the image is needed for reliably detecting features on the order of 30 cm high. The planner dynamically chooses which portion of the image that the stereo system should process, based on the current vehicle speed, stopping distance, and expected cycle time of the perception/planning/control loop. Typically, stereo is asked for points lying from 4 to 7 meters in front of the rover, at an 8 cm resolution.

To evaluate the potential steering commands, the planner uses a detailed model of the vehicle's kinematics and dynamics to project forward in time the expected path of the rover on the terrain. This produces a set of paths, one for each potential steering direction (Figure 7). The planner then evaluates, at each point along the path, the elevations underneath the wheels, from which it computes the rover's roll and the pitch of each body segment. The overall merit of a path depends on the maximum roll or pitches along the path, together with how known is the underlying terrain (in practice, there are often unknown terrain areas that are either occluded from view by obstacles, or are low-texture areas, which are not reliably processed by the stereo algorithm).

Experimental Results

To date, our research has focused on the controller components (on-board and off-board), the autonomous navigation aspects (stereo and obstacle avoidance planner), and the integration of the overall system. We have done extensive testing in outdoor, natural terrain to determine

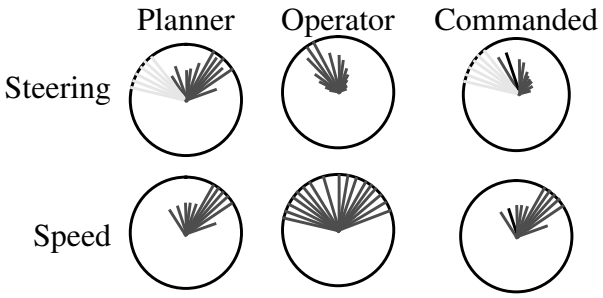

Figure 5: Arbitrating Operator & Planner Commands

rover out of a crater. This mode would be reserved for experienced drivers in exceptional situations. In contrast, in the *autonomous mode,* the software system has complete control.

The third mode, *safeguarded teleoperation*, is seen as the standard way in which the lunar rover will be operated. In this mode, input from the human and the rover are combined: the operator presents a desired direction to travel, and the rover can either veto it, causing the robot to refuse to travel in that direction, or can alter the command slightly to steer around obstacles. The idea is that the software safeguards should prevent the operator from damaging the rover, but should otherwise interfere only minimally. The user interface is designed to make it easy to switch between modes,. In particular, if the operator chooses not to provide input, only the rover's inputs are used to make steering decisions. In this way, operator fatigue can be reduced by letting the robot operate on its own when it is in benign terrain, while still enabling the operator to take over control at any moment.

Arbiter

The arbiter component provides a straightforward way to incorporate steering recommendations from various sources in a modular and asynchronous fashion [10]. The arbiter accepts evaluations for a set of steering angles from the user interface and obstacle avoidance planner components, and combines the evaluations to choose the overall best steering angle.

Each evaluation consists of a steering angle, value, and speed (Figure 5). If the value is "veto" (lightly shaded in the figure) then the arbiter eliminates that steering angle from consideration. Otherwise, it combines the recommendations from all sources using a weighted sum (the weights can be changed in the user interface).

Rather than choosing the absolute best evaluation, the arbiter actually chooses the steering angle which is at the center of the largest contiguous set of evaluations that are all close to the maximum value. In this way, the arbiter is biased towards wide, easily traversable areas over

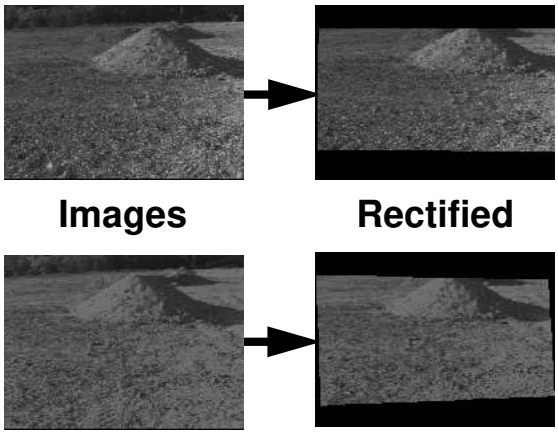

Figure 6: Rectified Stereo Images

directions that might be a bit more traversable, but have less leeway for error if the rover fails to track the path precisely (an added safety measure).

The operator's evaluations are generated by using a Gaussian distribution, centered at the actual steering angle chosen by the operator. The spread (variance) of the Gaussian dictates how much leeway the system has to deviate from the operator's intentions: If the variance is zero, then the user interface sends just the chosen steering angle, and the obstacle avoidance planner can either accept or veto it. If the variance is wide, the user interface sends a number of recommendations (whose values decrease the further they are from the operator's choice), and the arbiter is then free to choose steering angles on either side of the nominal one selected by the operator.

The recommendations sent to the arbiter are also tagged with a robot pose. If the tagged pose differs significantly from the rover's current pose, then those path evaluations are ignored. If the evaluations from all the processes are invalidated in this way, the arbiter commands the rover to stop. In this way, the arbiter safeguards against other modules failing to provide timely inputs (such as when they crash).

Stereo

The stereo component, used to produce terrain maps for local obstacle avoidance, takes its input from two black-and-white CCD cameras, mounted on a motion-averaging mast. The output is (x,y,z) triples, given in the camera coordinate frame, along with the pose of the robot at the time the images were acquired. Using the pose, the (x,y,z) values are transformed into world coordinates to form a (non-uniformly distributed) terrain elevation map.

The stereo images are first rectified (Figure 6) to ensure that the scan lines of the image are the epipolar lines [9]. The best disparity match within a given window is then

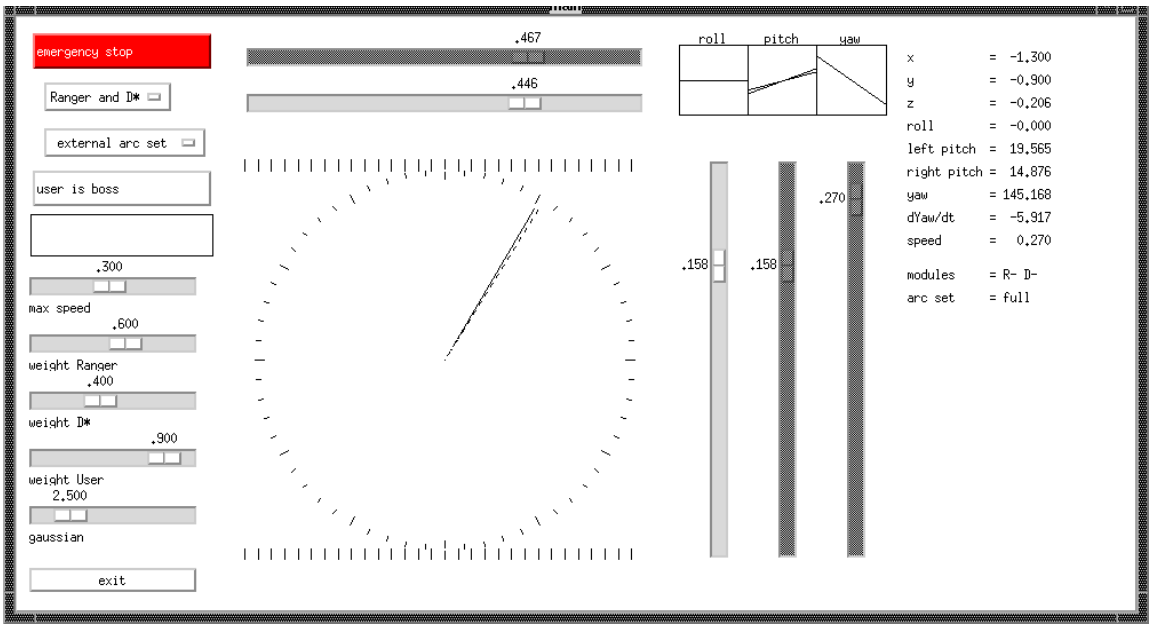

Figure 4: The Graphical User Interface

own obstacle avoidance planner to choose the best direction to travel, and then forwards steering and velocity commands to the off-board controller (and then to the on-board controller). The obstacle avoidance planner, in turn, bases its recommendations on analyses of terrain elevation maps produced by the stereo component. All components operate concurrently, and receive their inputs from other components asynchronously.

The following subsections describe each of the components depicted in Figure 3.

Controller

The on-board controller accepts velocity commands for the left and right pairs of wheels. It uses feedback from the wheel encoders to maintain the commanded velocity over a wide range of terrain conditions. The on-board controller also reports the various sensor readings (compass, gyro, inclinometers, encoders). It expects a "heart-beat" message from the off-board controller, and will halt all motions if not received periodically.

The off-board controller accepts desired steering and velocity commands, and converts these to wheel velocities for the on-board controller. It provides for several safety mechanisms, such as stopping the rover if roll or pitch inclinometers exceed certain thresholds, or if it does not receive a new command before the Ratler has traveled a specified distance.

The controller also merges encoder, inclinometer, compass and turn-rate sensor readings to estimate the position and orientation of the rover. In particular, extensive filtering and screening is performed on the data to reduce noise and eliminate outliers. For example, the compass signal is corrupted by random noise. Based on a spectral analysis of the data, which revealed a cut-off frequency of 0.25 Hz, we implemented several low-pass filters (Butterworth and Bessel). These are effective in suppressing the noise, although they also introduce a 2-3 cycle delay between the filtered value and the signal.

User Interface

While our focus has been on the technical, rather than the human-factors, aspects of safeguarded teleoperation, we have still tried to create a graphical user interface that facilitates mixed-mode teleoperation. The user interface consists of an "electronic joystick," which utilizes the computer mouse to command the robot's direction and speed, and a number of textual and graphical indicators of pertinent information, such as commanded and instantaneous robot speeds, roll and pitches, position, and status (Figure 4). Visualization of the terrain is provide by a color camera mounted toward the rear of the Ratler, which is transmitted over a microwave radio link to a monitor that sits next to the user interface workstation.

The user interface supports several driving modes. In the *direct teleoperation mode*, the human has full control over the rover — almost all safeguarding is turned off. Direct teleoperation is necessary when the rover gets into situations where the software would otherwise prevent motion. For instance, there may be occasions where the pitch limits must temporarily be exceeded to drive the

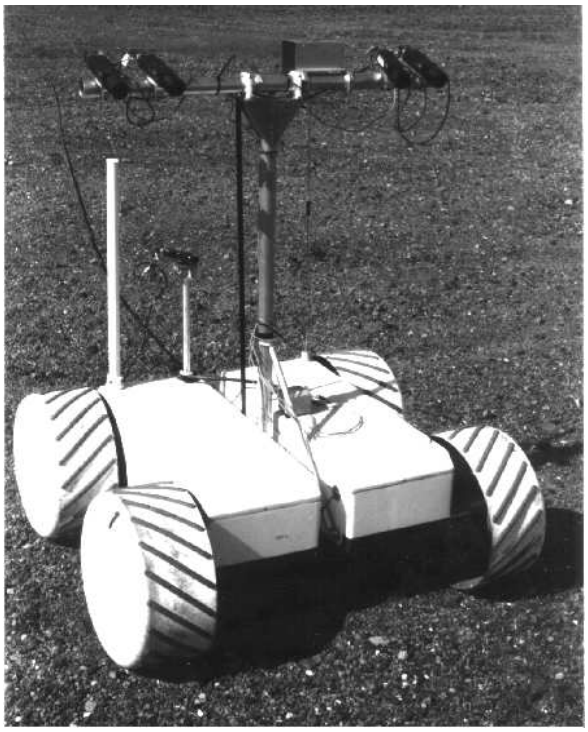

Figure 2: The Ratler Rover

can either veto, or slightly alter, the driving command if it would lead to a dangerous situation.

While other efforts have taken similar approaches to navigation for wheeled planetary rovers [1, 2, 3, 4, 12], including the use of obstacle avoidance using stereo vision, our work is distinguished by its emphasis on long-distance traversal, mixed mode driving, and use of efficient stereo vision using only general-purpose processors. We are currently working to demonstrate remote, safeguarded teleoperation of up to 10 km, and to quantitatively demonstrate the advantages of safeguarding for time-delayed teleoperation.

The next section describes the rover that is currently being used for our experiments. We then describe the software system developed to drive the rover, and our experimental results. Finally, we address work that is still needed and present our conclusions.

The Ratler

We are currently using a vehicle designed and built by Sandia National Laboratories [8] to test the navigation concepts and algorithms that we are developing. The Ratler (Robotic All-Terrain Lunar Exploration Rover) is a battery-powered, four-wheeled, skid-steered vehicle, about 1.2 meters long and wide, with 50 cm diameter wheels (Figure 2). The Ratler is articulated, with a passive axle between the left and right body segments. This articulation enables all four wheels to maintain ground contact even

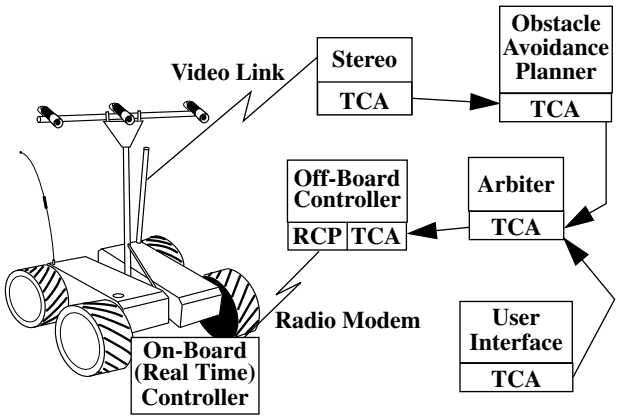

Figure 3: The Navigation System Architecture

when crossing uneven terrain, which increases the Ratler's ability to surmount terrain obstacles. The body and wheels are made of a composite material that provides a good strength-to-weight ratio.

Sensors on the Ratler include wheel encoders, turn-rate gyro, a compass, a roll inclinometer, and two pitch inclinometers (one for each body segment). There is a color camera for teleoperation, and we have added a camera mast and four black-and-white cameras for stereo vision (only two of which are currently being used). We have also recently added a laser proximity sensor (not pictured). On-board computation is provided by a 286 and a 486 CPU board, connected by an STD bus, which also contains A/D boards and digitizer boards for the stereo cameras.

The Navigation System

The rover navigation system consists of a number of distributed processes that communicate via message passing protocols (Figure 3). For ease of development and debugging, the system is currently divided into on-board and off-board components, although in the actual Lunar rover, all but the user interface component will be on board.

The on-board (real-time) controller handles servo control of the motors and sensor data acquisition. The on-board and off-board controllers communicate over a serial link using the RCP protocol developed at Sandia. The rest of the components communicate over the Ethernet via the Task Control Architecture (TCA). TCA is a general-purpose architecture for mobile robots that provides support for distributed communication over the Ethernet, task decomposition and sequencing, resource management, execution monitoring, and error recovery [11]. TCA connects processes, routes messages, and coordinates overall control and data flow.

The arbiter component is key to the implementation of mixed-mode navigation. The arbiter combines recommendations from the remote user and the rover's

Mixed-Mode Control of Navigation for a Lunar Rover

Reid Simmons, Eric Krotkov, Lonnie Chrisman, Fabio Cozman, Richard Goodwin, Martial Hebert, Guillermo Heredia, Sven Koenig, Pat Muir, Yoshikazu Shinoda, William Whittaker

The Robotics's Institute, Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

The question of how to navigate is critically important to the success of any lunar rover mission. While humans typically have good judgement about which routes to take, they often get fatigued and disoriented when teleoperating rovers with time delay. On the other hand, while autonomous systems can produce safe and reliable navigation, they tend to be myopic. We are investigating mixed-mode methods of control that combine the strengths of humans and rovers. The rover uses range maps produced by stereo vision and a detailed model of the vehicle to evaluate the traversability of various paths. The evaluations are combined with recommendations from a human operator to produce a commanded steering angle and speed that is both safe and responsive to the operator's objectives. We have implemented and are testing such a system, using a prototype lunar rover that operates in outdoor, natural terrain.

Introduction

The next visitors to the Moon may be robots. In one promising scenario, a pair of rovers would be landed on the Moon for a multi-year, 1000 kilometer traverse of historic sights, including Apollo 11, Surveyor 5, Ranger 8, Apollo 17 and Lunokhod 2 [5]. The robots would be driven by operators on Earth, based on panoramic stereo images from the rover's perspective (Figure 1).

Even in the best of circumstances, experimental evidence shows that teleoperation of robots is fatiguing and disorienting for operators. In addition, for remote Lunar driving, operators would be further hampered by up to a five second round-trip communications delay. Such factors imply that remote Lunar driving would likely either put the safety of the rover at risk, or would have to be done too slowly to accommodate the 1000 kilometer mission.

An alternative scenario is to have the rover drive itself, autonomously. This eliminates the factor of time delay, but would make the rover more complex by adding hardware and software. In addition, the current autonomous robot navigation algorithms may not be applicable in all situations, especially when the rover finds itself in tight or unusual situations.

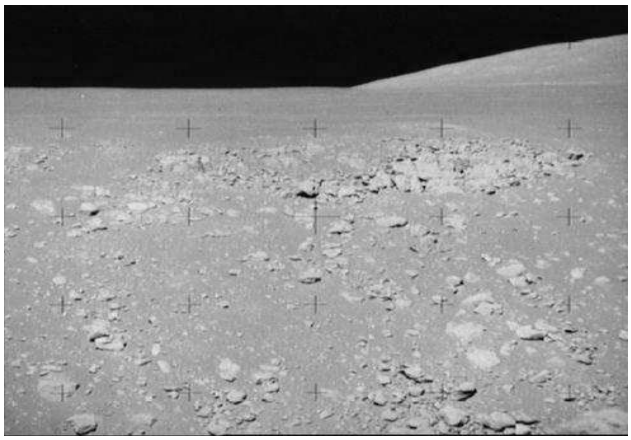

Figure 1: Typical Lunar Terrain

The question then is how to combine the relative strengths of the human operator and the rover to produce reliable, goal-driven navigation? How can we take advantage of the human's common sense and long-range planning capabilities, and the rover's ability to sense and react quickly and dependably?

We are investigating these issues in the context of a larger program to develop techniques that would be useful for planetary rovers and mobile robots, in general. In particular, we are investigating techniques for stereo vision, local obstacle avoidance, position estimation, and user interaction. The aim is to provide both the technologies and evaluations of their effectiveness, in order to enable mission planners to make informed cost/benefit tradeoffs in deciding how to control rovers.

The work reported here is in the area of mixed-mode operation, where a human operator and an autonomous system each provide "advice" on how to drive, with the recommendations arbitrated to produce the actual steering commands to the rover. By suitably combining the advice from the human and rover, the rover can operate under either pure teleoperation, autonomous operation (where the rover uses stereo vision and local obstacle avoidance planning to steer itself), or *safeguarded teleoperation*, where the human provides the primary input and the rover