

# Obstacle Avoidance and Safeguarding for a Lunar Rover

Reid Simmons, Lars Henriksen, Lonnie Chrisman and Greg Whelan  
School of Computer Science/Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213

## Abstract

We are developing techniques for safeguarding the remote operation of lunar rovers. This paper presents two complementary techniques: One, based on stereo vision, evaluates the traversability of paths the rover could follow, and produces preferences for steering directions. The other, based on laser proximity sensing, looks for hazards immediately in front of the rover, commanding an emergency stop if any are detected. The stereo-based technique provides reliable obstacle avoidance, but operates fairly slowly, while the laser-based technique operates faster and is more conservative in its evaluations. The stereo-based obstacle avoidance planner has been used to drive a rover over ten kilometers in outdoor, natural terrain. The laser proximity system has been tested, and is currently being integrated with the rest of the rover system.

## 1. Introduction

We are investigating techniques to help humans operate rovers on the Moon. Our work belongs to a larger Lunar Rover Initiative, which aims to conduct a lunar mission, sponsored by private ventures, dedicated to returning to the Moon before the turn of the Millennium [3]. The mission would have rovers navigate hundreds, if not thousands, of kilometers over several years, visiting sites of geological and historical interest. The research reported here involves techniques for planning rover motions that are (locally) safe and easily traversable.

Our motivation starts with the observation that teleoperation of mobile robots is often fatiguing and disorienting for operators. This is especially true for remote Lunar driving, in which the environment is foreign, and operators would have to contend with up to a five second communications time delay. While an alternative scenario is to have the rover drive itself autonomously, the ability to teleoperate the vehicle constitutes much of the appeal of the mission. Furthermore, the current state-of-the-art is not reliable enough to enable the robot to make correct decisions in every conceivable situation.

Our philosophical approach is to combine the relative strengths of the human operator and the rover to produce

reliable, goal-driven navigation. The idea is to 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. The idea, which we call "safeguarded teleoperation," [9] is to let the humans guide the rover, but have software running on-board that safeguards the vehicle by preventing dangerous movements, or biases the vehicle's actions towards more easily traversable areas of the terrain.

Our implementation approach is systemic and layered. By *systemic*, we mean that we are building a complete, integrated robot system — from real-time control to user interface. By *layered*, we mean that higher-level system functionality is built on top of lower layers. For example, the lowest level — the real-time control — accepts commands in the form of steering angle and velocity. Local obstacle avoidance uses this layer to autonomously traverse terrain. The safeguarded teleoperation combines the local obstacle avoidance and user interface layers to produce safe, reliable navigation. Advantages of the layered approach are increased flexibility (the system can be commanded at any layer) and increased reliability (if designed correctly, the reliabilities of each layer complement one another).

Central to our approach is on-board software to sense and react to terrain features. The work reported here consists of two techniques: one for local obstacle avoidance based on stereo vision, and one for hazard detection based on a laser proximity rangefinder.

The rationale for using two techniques based on different sensing modalities is that they effectively complement one another. Stereo-based vision provides a relatively wide, medium-range view of the terrain (three to seven meters in front of the vehicle), but is rather slow (less than 1 Hz) and has only fair resolution (5-10 cms). The local obstacle avoidance planner that uses the stereo data is rather sophisticated, and is used to make decisions about where the rover should, and should not, be heading. The laser-based proximity sensor provides high-resolution (under a centimeter) data at a fast rate (minimum of 4 Hz), but in a relatively narrow band close to the front of the vehicle (100-150 cms). The hazard detection software uses simple, conservative heuristics to determine if a variety of hazards are present, and acts to stop the vehicle in an emergency.

Although we presume that the stereo system will keep the rover out of most hazardous situations, the laser system acts as a short-range backup. The combination increases the overall reliability of the navigation system, and increasing our confidence that the rover will not accidentally drive over a cliff or become stuck.

Both techniques have been implemented and are being tested on a prototype wheeled rover. In particular, in one experiment the stereo-based local obstacle avoidance system was used to drive the rover safely over ten kilometers in outdoor, natural terrain. We are currently integrating the laser-based hazard detection subsystem, and will test the complete system by traveling autonomously over greater distances and rougher terrain.



Figure 1. The Ratler Rover

## 2. The Rover And Its Navigation System

While we await the completion of our new lunar rover [1], we are using a vehicle designed and built by Sandia National Laboratories [5] as a testbed to develop the remote driving techniques needed for a lunar mission. 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 1). 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 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, and an Accuity laser proximity rangefinder.

Figure 2 presents a block diagram of the overall navigation software system. The real-time controller handles servoing of the motors, collecting and processing of the internal sensor signals (encoders, compass, inclinometers), and dead-reckoning calculations [4]. It runs on a 286 and a 486 CPU board, connected by an STD bus, and communicates with the rest of the system via serial link. The laser subsystem (Section 4) also runs on-board, on another 486 board.

The controller module (Figure 2) transforms higher-level commands (steering angle and velocity) into the lower level commands (individual wheel velocities) used by the real-time controller, and transforms raw sensor signals into more familiar units (radians and meters). The stereo and obstacle avoidance planner modules work together, taking pairs of images and producing recommendations on which paths are best for the rover to traverse. The arbiter module combines information from the planner and user interface subsystems to select paths that satisfy both user preferences and vehicle safety [7, 8]. Each module is a separate process, running concurrently, and communicating with one another via Ethernet, using the message passing protocol of the Task Control Architecture [10]. Currently, these modules run off-board, on two Sparc 10 workstations, but we are in the process of porting them (except for the user interface) to run on-board, on two Pentium processors running Linux.

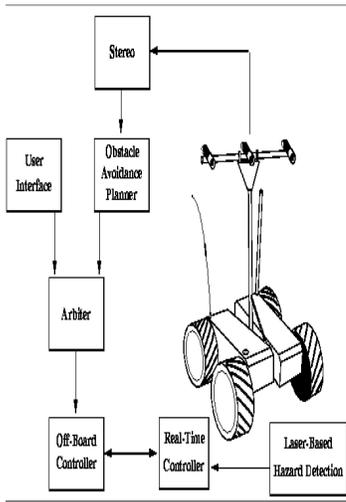


Figure 2. Navigation System Block Diagram

### 3. Stereo-Based Obstacle Avoidance

#### 3.1 Stereo Vision

The local obstacle avoidance planner uses stereo-based terrain elevation data to determine safe paths for the rover to travel. The stereo module takes its input from black-and-white CCD cameras, mounted on a motion-averaging mast (Figure 1). The camera images are first rectified to ensure that the scan lines of the image are the epipolar lines [6]. The best disparity match within a given window is then computed using a normalized correlation. Disparity resolution is increased by interpolating the correlation values of the two closest disparities. Various heuristics are employed to minimize outlier values (caused by false stereo matches), for example, by eliminating low-textured areas using lower bounds on the acceptable correlation values and variance in pixel intensity [4, 8].

The output of the stereo subsystem are sets of  $(x, y, z)$  triples, given in the camera coordinate frame, along with the pose of the robot at the time the images were acquired. Using the pose information, the obstacle avoidance planner transforms the  $(x, y, z)$  values into world coordinates to form a (non-uniformly distributed) terrain elevation map.

To make the stereo computation tractable, the planner requests only a small segment of the stereo image (about 2%), at reduced resolution (every fifth row and column). Experiments show that this is sufficient to reliably detect features on the order of 20 cm high.

#### 3.2 The Ranger Planning Algorithm

Our first local obstacle avoidance planner was an adaptation of a planner, called Ranger, that was developed at CMU for ARPA's Unmanned Ground Vehicle (UGV) program for cross-country navigation [2]. This planner enabled the rover to travel up to a kilometer in mild terrain [4, 8, 9].

The Ranger algorithm works by analyzing the paths the vehicle would traverse along the terrain for a number of different steering angles, and choosing the one that evaluates as the safest. It merges individual stereo-produced elevation maps to create a 25 cm resolution grid map up to seven meters in front of the rover. Map merging is necessary because the limited fields of view of the cameras do not allow a single image to view sufficient terrain.

Ranger then projects the rover's state (position, roll, pitch, yaw) as it travels along a path. The projection is based on a desired steering angle, the vehicle dynamics, and the underlying terrain. The vehicle's current pose, its dynamics and the steering angle are used to determine the position and yaw of the vehicle at the next time step. The height of the terrain under the wheels is then used to determine the roll and pitch of the vehicle at that point.

Once a projection of the vehicle along a path has been computed, the vehicle state at each point in time is evaluated. Four criteria are used to determine the "goodness" of a path: roll, pitches of the left and right body segments of Ratler, and number of known terrain points the vehicle crosses along the path. If any of the criteria exceed a given threshold of safety (such as excessive roll or pitch), the whole path is given a very low evaluation. Otherwise, the criteria are normalized to the range [0..1] and are combined using a linear weighted function. This determines the overall merit of choosing that steering angle for the rover. These evaluations are then combined with the user's preferences to determine the overall best command, which is then sent to the rover to be executed. The cycle time for this process is about 1-2 seconds, with the stereo computations taking up about 75% of the time.

#### 3.3 The Morphin Planning Algorithm

While the Ranger algorithm has worked well for high-speed navigation of Humvees, it is not entirely well-suited to the much smaller, and slower, lunar rover. As is often the case in robotics, the problems are mainly attributable to an abundance of noise, particularly in the stereo-produced terrain maps and the dead-reckoning. The main effect of the noise is to make it difficult to cleanly merge terrain maps acquired from separate images, which is required by the Ranger algorithm since it uses only a small percentage of each image. Map merging often produced artifacts in the map, such as crevasses and ridges, which the rover would refuse to cross. This is less of an issue with the Humvees, since they can cross much taller obstacles. We tried several merging techniques in an attempt to minimize the artifacts, but none was robust enough to yield consistent driving results.

Another effect of noisy terrain data is that, because of the rover's relatively short wheelbase, small deviations in perceived terrain elevation under the wheels produced relatively large changes in estimated roll and pitch. For example, a 20 cm "spike" in the terrain map (not uncommon) produces a 13 degree change in pitch, given a 90 cm wheelbase. Thus, it is often difficult to distinguish noise from steep bumps. This same problem makes it difficult to reliably determine whether high-centering might occur, since the clearance of the rover is not much more than the noise in the map. Finally, the Ranger algorithm presumes that the rover can track the path exactly, and does not account for uncertainty in the execution of commands or for uncertainty in the vehicle dynamics models used to project paths.

To address these problems, we modified parts of the Ranger algorithm, creating an algorithm called Morphin (a "power" Ranger). In contrast to the *path-based* approach of Ranger, Morphin is *area-based*: it analyzes patches of terrain to determine the traversability of each patch, and evaluates the traversability of a path by determining the set of patches it travels through. As such, it is more akin to the terrain navigation planner of [11].

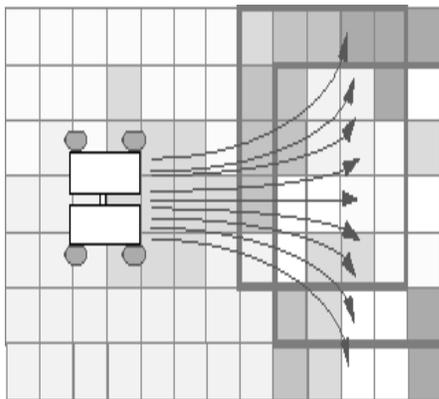


Figure 3. Local Obstacle Avoidance Planner

Each local terrain map, produced from a single stereo pair, is analyzed independently. The terrain is divided into overlapping patches, each 125 cm on a side, with patches offset by 25 cm from one another. Thus, each terrain point contributes to 25 patches (Figure 3).

To determine traversability, a plane is fit to each patch using least-squared error. To avoid redundant computation, statistics (e.g., sum of X, sum of XY) are collected for smaller 25 cm squared patches and then aggregated to determine the plane parameters for each 125 cm squared patch. The plane parameters are used in determining the vehicle roll and pitch (see below), and the residual to the plane fit is used to estimate the roughness of the terrain. Two "roughness" measures are computed: one based on the residual in fitting the plane to the whole patch, and one based on the residual of each small (25 cm squared patch).

The former indicates the roughness of the overall area, while the latter indicates if the patch is bumpy/spiky. Finally, two factors are used to assess the confidence in the evaluations: the number of stereo-generated terrain points in a patch and the spatial distribution of these points (based on an entropy-like measure), which is used to ensure that the points are representative of the patch as a whole.

Morphin then projects the path of the rover over the terrain patches. Unlike Ranger, which uses a discrete numerical simulation to project paths, Morphin uses closed form solutions to calculate the intersections between arcs of a circle and the terrain patches. Morphin then sums the traversability metrics of the intersecting terrain patches, weighted by the length of the intersection between the arc and terrain patch. For each patch, Morphin determines roll, pitch, roughness, and confidence in the data. The pitch of the vehicle is easily calculated as the slope of the line along the plane in the direction of the current heading (yaw). A similar calculation yields the vehicle roll. The roughness and confidence measures are calculated as described above. If there are overlapping patches from different images, only the one associated with the most recently acquired image is used (given the dead-reckoning uncertainty of the rover, we find this to be much more effective than trying to combine overlapping evaluations in some way). Then, as with Ranger, the criteria are combined using a linearly weighted function.

While the path projection approach of Ranger (numerical simulation) produces higher fidelity paths (since dynamics and the effects of moving on uneven terrain can be taken into account), Morphin's geometrical approach is much more efficient, and is adequate for the task since the rover's dead-reckoning is not accurate enough to warrant a high fidelity approach. In fact, we are extending Morphin to explicitly deal with the uncertainty in the rover's heading: for each nominal steering angle, we project a number of paths (currently five) that differ slightly in the steering angle. The evaluation for each of these paths is weighted by the probability of the rover following that path (under an assumption of Gaussian distribution from the nominal steering angle).

### 3.4 Performance

To evaluate the strengths and weaknesses of the stereo-based approach, we performed extensive field trials. The test site (Figure 4) consists of soil, crumbled asphalt, loose gravel, scree, and some grassy vegetation. Obstacles to rover passage include soil mounds, depressions, cliffs at the river bank, building walls, metal pipes, cement blocks, railroad ties, trees, and bushes.



Figure 4. Terrain For Rover Navigation Experiments

In one particular experiment, the rover traveled more than 10 km over a three-day period. During the experiment, the rover operated autonomously over 98% of the time, successfully avoiding discrete obstacles, while averaging a speed of 15 cm/sec. This is an order of magnitude farther than we were able to traverse with the Ranger algorithm, and needed about one-third the amount of teleoperated control. This experiment demonstrated the superiority of the Morphin algorithm for our rover. Morphin addresses the problem of noisy data by aggregating independent data points into an overall statistic, thus dramatically lessening the impact of any single point. While this aggregation can sometimes cause the rover to behave more conservatively than would otherwise be warranted, in our application it is better to be too conservative than to allow the rover to head into danger.

## 4. Laser-Based Hazard Detection

While the stereo-based planner is fairly reliable, there are several hazards that it has trouble detecting. The major weakness is that the stereo vision often cannot detect depressions/craters, reporting them as unknown areas. In addition, the limited resolution of the stereo, combined with the large look-ahead distance (three to seven meters) means that small obstacles (on the order of 10-20 cms) may be overlooked. These can cause problems if the rover tries to straddle them, which can cause high-centering (hitting the bottom of the vehicle).

To detect such hazards, we have developed a hazard detection technique that uses a high-resolution, laser proximity sensor. The requirements for this subsystem are that it must be very robust in detecting hazards and have very good response time. These requirements have driven the design and implementation of the laser-based safeguarding system.

### 4.1 System configuration

The sensor, an Acuity 3000-LIR laser ranger, sends a beam towards a rotating mirror projecting a plane of infrared laser light at a 45 degree angle to the ground. It is able to image the ground with a resolution of under a centimeter in all three dimensions at a range of about 100-150 cm in front of the rover. The effective field of view is limited by the effective angle of incidence and is, in practice, about 90 degrees producing a 4 m long laser line on the ground in

front of the rover. The scanner can produce data at various rates, depending on the number of samples per scan and the required precision. In the runtime configuration, a scan is available every 25-50 msec.

An on-board computer collects the range and angle readings and tests them for validity. The data is then linearized and transformed to obtain an array of (x, y, z) triples of the terrain with respect to the rover's local coordinate frame (i.e., this transformation does not adjust for the angular inclination of the vehicle). The resulting laser data are processed, looking for evidence of depressions/drop-offs and obstacles that might lead to the vehicle being stuck when attempting to drive over them. When such hazards are detected, the subsystem issues an emergency stop command to the vehicle and notifies the local obstacle avoidance planner (Morphin) of the hazard, so that it can incorporate that information into its planning.

As a baseline configuration the position information available on the rover is *not* incorporated into the detection of the hazards. This frees the laser safeguarding system from dependence on the controller module maintaining accurate dead-reckoning information, and hence makes it less dependent on sensor failures (encoders, compass). Even if all other navigation systems should fail the rover can still be controlled safely in teleoperated mode by the laser safeguarding. Besides being robust, this configuration involves less processing, which leads to a faster update rate. While incorporating pose information is an option, and might possibly produce better hazard detection, doing so is non-trivial since the desired cycle and reaction times of the laser subsystem are considerably smaller than the inherent time constants of the inclinometers. Our approach instead aims at identifying statistics of the scans that are insensitive to sudden (and unknown) inclinations. By using these, we have found this baseline configuration to be sufficient for all but the most extreme rover configurations.

### 4.2 Data acquisition

The first step in processing a laser scan is to determine the integrity of the laser system and to perform self-diagnostics, if necessary. The next step is to remove invalid data and to determine if the spatial density of the remaining data is high enough to reliably calculate the hazard metrics.

These calculations use a number of laser ranger sensor signals: absolute encoder, incremental encoder, range, temperature, data out of range, buffer overflow, intensity of reflected laser light, and ambient light. First the motor subsystem is checked through a test of correct motion of the mirror. This is done using three measures:

- Is zero pulse captured? (absolute encoder)
- Full cycle loaded? +/- 45 degrees in front of the vehicle captured.

- Is motor spinning? (incremental encoder)

The zero pulse of the absolute encoder synchronizes the angles captured by the incremental encoder. If this pulse is missed, the absolute orientation of the sweep is unknown and the data is of no value. Both the capture of the synchronizing zero pulse and a successful acquisition of a full cycle depends on the speed of the mirror. If the mirror is spinning too fast, the zero pulse may be missed and, if spinning too slowly, a full range may not be available within the number of samples recorded. As the mirror system has relatively slow dynamics, the system is designed so that the zero pulse or full angle measures have to fire a number of times before the spinning of the motor is tested. This avoids erroneous fault detection during start-up and temporary disturbances.

In addition to determining whether the mirror is spinning correctly, a check is made of the motor temperature and whether there are internal errors (e.g., buffer overflow) on the SCSI interface board (which indicates that samples have been lost). Finally the system assesses whether the density of reliable data is sufficient. A common problem is that the laser beam hits a terrain point which does not reflect enough light to make an accurate range estimation. This can be due to the angle of incidence, non-diffuse reflection, or a low reflectance of the object being measured (dark surface). This results in an unreliable datum, which can confound subsequent processing. A dependable way to detect zones of unreliable data is high variance between adjacent range readings.

All checks, except for the variance in range estimates, are very fast as their input are direct sensor signals, which are more or less dedicated for integrity analysis purposes. Only the high-variance test needs a non-trivial amount of computation to determine status. In any event, data acquisition is fast: including integrity checking and data testing, it can be done in about 180 msec (including 35 msec for the laser to generate range data).

When a problem occurs, corrective action is necessary. For some of the very low level problems, like mirror motion, appropriate actions can be directly associated with the problem. In the case of mirror motion problems, new scans are commanded to see if the problem was just a result of spurious unfavorable conditions. For other problems, such as high temperature, different actions can be taken involving other systems of the rover (like applying extra cooling, shutdown or seeking shade). Since other subsystems may also be affected by these kind of problems, in most cases the laser subsystem will just discard the data as invalid, and leave it to other systems to correct the problem.

### 4.3 Hazard detection metrics

Since the laser line hits the ground fairly close to the vehicle (100-150 cm), detection must be made quickly in order to react in time. For this reason, we have chosen to define simple heuristic metrics for each type of hazard that we want the laser to detect. These metrics are defined in terms of a single scan of the proximity sensor, so that no information needs to be saved between scans (increasing robustness and decreasing computation).

When designing the metrics two approaches were considered. One approach evaluates whether the elevation of the surface in front of the rover (represented in the rover’s local coordinate frame) exceeds the capability of the rover. While this approach is fairly general and computationally very simple, it has the problem that the apparent elevation of the terrain in front of the rover is a function of both the actual terrain height and the rover’s current inclination (e.g., if the front wheels of the rover are on small rocks, the elevation of the terrain one meter in front of the rover appears lower than it actually is). Thus, while true hazards will be detected reliably and quickly, there are situations where potential hazards will be detected erroneously, and the vehicle will be stopped unnecessarily.

The other approach involves identifying *signatures* of different landscape formations that are invariant to the motions that occur when driving over minor obstacles. For example, when obliquely approaching a downward slope, the range measurements will gradually increase starting at the point where the laser line intersects the beginning of the slope, forming an “elbow bend.” This characteristic shape is evident regardless of whether the front of the rover is elevated by a rock, and so is less likely to detect hazards erroneously. However, in the signature approach it is difficult to quantify the danger a profile constitutes to the vehicle. For example, when approaching a minor downwards slope from different angles, the shift in range varies and so the steepness of the slope cannot be known. Thus, it is difficult to quantify what constitutes a real hazard. In addition, in the signature approach much more processing has to be performed, as the number of possible landscape feature signatures is relatively large compared to the number of rover limitations (see Table 1).

Table 1. Hazards to be detected

Rover limitation	Landscape danger	Importance
Positive elevation (step)	Small, medium and large rocks Step in landscape (broken rock surface) Boulders	Less important. Stereo is reliable

Negative elevation (ditch)	Ditch Craters Step in landscape (broken rock surface)	Important. Stereo has poor performance here.
Stuck on belly	Objects on cross slopes	Equally good

Another problem with the interpretation approach is that the set of features may not cover all possible landscapes encountered. Hence, safe operation would not be guaranteed. To ensure safety (at the cost of sometimes stopping erroneously), it was decided to employ the direct method based on the capabilities of the rover. Three hazard types are considered:

- Maximum traversable step (curb-like, head on)
- Maximum traversable ditch (curb-like, head on)
- Belly clearance

As the metrics are defined in terms of a single scan, no information is available about the transition from the surface currently under the rover to the scanned surface at the laser line. The transition must therefore be treated as a worst case, which is a step-like transition at the laser line. Also, since the laser subsystem does not know the current vehicle steer angle, to be safe it must analyze the complete laser line. For the step and ditch metrics, this translates into defining a simple upper and lower threshold (respectively) directly on the 3D elevation profile (Figure 5). The thresholded data is spatially filtered to prevent spurious signals from firing the metric. A median filtering is used, which is quite fast since it operates in the binary domain.

The belly hazard metric first estimates the slope by linear regression and then equalizes the elevation profile accordingly, yielding a level elevation profile centered around zero elevation. Based on the minimum and maximum elevation in this compensated profile, the most favorable levels of a positive and a negative threshold is computed (difference between the two levels is the body clearance minus a margin). The compensated elevation profile is then tested for exceeding the elevation band defined by the two threshold levels and this output is filtered spatially as for the step and ditch metrics. Since this metric is more computationally expensive than the other two, it is processed last (and only if the other two do not fire).

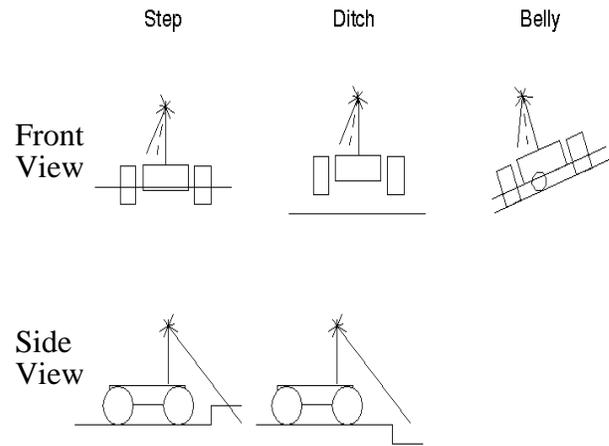


Figure 5. Physical metrics and their corresponding hazards

As an example, Figure 6 shows the interpretation of a typical scene. The elevation profile is inclined to the left, relatively flat, and shows a small mound at  $y=-1m$ . The two dashed lines indicate the step and ditch thresholds. For  $y>0.5$  the step metric has detected a hazard (denoted by “o”s). No belly hazard is detected.

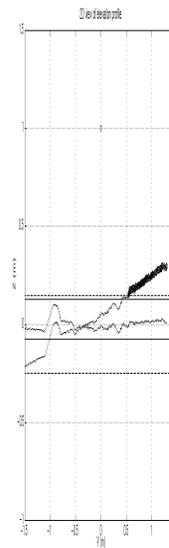


Figure 6. Hazard detection. The elevation profile is seen in the direction of travel from the vehicle. The laser is positioned at  $y=0, z=1$ .

#### 4.4 Performance

We have implemented the integrity checks and hazard detection metrics described above, and are currently running experiments to characterize their performance. Preliminary indications are that laser proximity safeguarding will be a very valuable supplement to the overall navigation system. In terms of missed hazards, the performance is excellent. Some false detections are encountered, mainly due to specular reflecting surfaces and small angles of incidence. This is, however, very dependent

on the scene used for testing and the pose of the laser scanner. Some of these problems can thus be overcome by placing the sensor in a more favorable location.

The cycle time is currently about 4 Hz on a 66Mhz 486 in the test configuration. This includes no effort for optimizing the algorithms, relatively dense sampling, and a high range precision, which requires more time by the laser (currently 15% of the total time). It is expected that the speed can be increased considerably without significant loss of detection reliability by streamlining code and reducing range precision to more realistic values.

An obvious extension would be to incorporate information about vehicle movements and the planned path. This would enable the metrics to evaluate each point on the elevation profile in accordance with the specific parts of the rover that will be passing that point. However, the add-on should be kept apart from the base-line system so that there always is a working backup if the required information from the rover should become unavailable.

## 5. Conclusions

This paper has presented an integrated approach to safeguarded navigation for lunar rovers. The key idea is to combine multiple techniques, using different sensing modalities, to increase the reliability of the overall system. In particular, we presented two complementary techniques: stereo-based vision for obstacle avoidance and laser-based hazard detection.

The stereo-based approach allows the rover to actively change its steering angle to avoid obstacles in the mid-range (3-7 meters in front of the rover), but it is rather slow, and can miss certain types of hazards, such as depressions. The laser-based subsystem is much simpler and faster, analyzing high-resolution data immediately in front of the rover (100-150 cms), but can only command the robot to stop.

We anticipate that the combination of these two techniques should give very good performance over wide range of terrain. Having tested the stereo and obstacle avoidance planner extensively (in one experiment autonomously traversing 10 kilometers of natural terrain), we know the types of terrain where its performance is weak, and have designed the laser proximity system specifically to address those weaknesses.

We are currently working to integrate the laser-based hazard detection component into our complete navigation system, and to fine tune the various metrics. The next step is to demonstrate that the integrated system will enable the rover to navigate extremely reliably in rougher and more varied terrain.

## Acknowledgments

Many thanks to other members of the Lunar Rover Navigation team, in particular Eric Krotkov, Martial Hebert, Fabio Cozman, Richard Goodwin and Sven Koenig. This work has been partially supported by NASA, under grants NAGW-3863 and NAGW-1175.

## References

- [1] L. Katragadda, J. Murphy, W. Whittaker and D. Gump. Commercial lunar rover mission for science and edutainment, In *Proc. AIAA Forum on Advanced Developments in Space Robotics* (this volume), Madison, WI, August 1996.
- [2] A. Kelly. An intelligent predictive control approach to the high speed cross country autonomous navigation problem. Technical Report CMU-CS-TR-95-33, Carnegie Mellon University, 1995.
- [3] E. Krotkov, J. Bares, L. Katragadda, R. Simmons, and R. Whittaker. Lunar rover technology demonstrations with Dante and Ratler. In *Proc. Intl. Symp. Artificial Intelligence, Robotics, and Automation for Space*, Jet Propulsion Laboratory, Pasadena, California, October 1994.
- [4] E. Krotkov, M. Hebert, and R. Simmons. Stereo perception and dead reckoning for a prototype lunar rover. *Autonomous Robots*, 2(4):313-331, December 1995.
- [5] 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.
- [6] L. Robert, M. Buffa and M. Hebert. Weakly-calibrated stereo perception for rover navigation. In *Proc. Image Understanding Workshop*, 1994.
- [7] J. Rosenblatt and C. Thorpe. Combining multiple goals in a behavior-based architecture. In *Proc. Intl. Conf. Intelligent Robots and Systems (IROS)*, 136-141, Pittsburgh, PA, August 1995.
- [8] R. Simmons, E. Krotkov, L. Chrisman, F. Cozman, R. Goodwin, M. Hebert, L. Katragadda, S. Koenig, G. Krishnaswamy, Y. Shinoda, W. Whittaker, and P. Klarer. Experience with rover navigation for lunar-like terrains. In *Proc. Intl. Conf. Intelligent Robots and Systems (IROS)*, 441-446, Pittsburgh, PA, August 1995.
- [9] R. Simmons, E. Krotkov, L. Chrisman, F. Cozman, R. Goodwin, M. Hebert, G. Heredia, S. Koenig, P. Muir, Y.

Shinoda, and W. Whittaker. Mixed-mode control of navigation for a lunar rover. In *Proc. SSI/Princeton Space Manufacturing Conference*, Princeton, New Jersey, May 1995.

[10] R. Simmons. Structured control for autonomous robots. *IEEE Transactions on Robotics and Automation*, 10:1, Feb. 1994.

[11] A. Stentz. The NAVLAB system for mobile robot navigation. PhD thesis, Carnegie Mellon University, 1989.