Mapping and Positioning for a Prototype Lunar Rover

Eric Krotkov and Martial Hebert

Robotics Institute. Carnegie Mellon University, Pittsburgh, PA 15213

Abstract

In this paper, we describe practical, effective approaches to outdoor mapping and positioning, and present results from systems implemented for a prototype lunar rover. For mapping, we have developed a binocular head and mounted it on a motion-averaging mast. This head provides images to a normalized correlation matcher, that intelligently selects what part of the image to process (saving time), and subsamples the images (again saving time) without subsampling disparities (which would reduce accuracy). The mapping system has operated successfully during long-duration field exercises, processing streams of thousands of images. The positioning system employs encoders, inclinometers, a compass, and a turn-rate sensor to maintain the position and orientation of the rover as it traverses. The system succeeds in the face of significant sensor noise by virtue of sensor modelling, plus extensive filtering and data screening.

1 Introduction

Carnegie Mellon University has undertaken a research, development, and demonstration program to enable a robotic lunar mission. The two-year mission scenario is to traverse 1,000 kilometers, revisiting historic sites, and returning continuous live video. During that traverse, control will be traded between the rover and users who will be provided with rich visual feedback. The resulting experience is intended to attract mass participation and evoke strong public interest in lunar exploration [11]. To enable such a lunar mission, we are developing and testing key mobile robot navigation technologies, including mapping and positioning.

The current terrestrial platform is the Ratler, a four-wheeled vehicle developed by Sandia National Laboratory. This skid-steered vehicle features an articulated chassis in which the body is divided into two halves, with two wheels on each side. Control of the Ratler may be directed from a local pendant, a remote command station, or on-board processors.

The goal of the work reported here is to develop and demonstrate lunar-relevant mapping and position estimation capabilities. Various elements of these capabilities have been developed in theory and reported elsewhere. The principal contributions of the work reported here are in (1) the advanced development of a number of existing but disparate approaches into practical algorithms that are effective in natural terrain; and (2) the real-world demonstration and evaluation of these algorithms in field trials processing thousands of image pairs and millions of state sensor readings.

In the next section we review related work. In the following sections, we describe the mapping and position estimation approaches that have been implemented, and present quantitative experimental results. In the final section we summarize the results, and look toward the future.

2 Related Work

The problems of mapping and positioning for planetary rovers have attracted significant attention around the world.

A majority of the efforts to navigate mobile robots in natural terrain has employed laser rangefinders [3][7][8][12] or proximity sensors [2] rather than stereo. There have been notable exceptions. Matthies [13] developed a near-real-time system using Datacube hardware, and demonstrated 100 m traverses with a planetary rover prototype. Faugeras et al. [1][4] developed a real-time system using Digital Signal Processors. Ross [16] developed a trinocular stereo system for the Dante walking robot.

Each of these stereo systems has its virtues, and each exhibits great promise. What has not been reported in the literature is an approach that achieves good performance without either special-purpose hardware or stringent requirements for alignment.

Virtually all efforts to navigate mobile robots face the position estimation problem in one form or another. Some choose to ignore it, others choose to water it down, and others attack it with Kalman filters and exotic sensors. Our approach applies aggressive filtering to readings from a suite of simple sensors.

3 Mapping

The mapping system consists of a stereo module that derives terrain information from binocular images. The hardware consists of two CCD cameras, with auto-iris 8 mm lenses, mounted on a motion-averaging mast, a video link,

IEEE International Conference
on Robotics and Automation
0-7803-1965-6/95 $4.00 ©1995 IEEE

and off-board frame grabbers and processors. The mapping software takes as input a stereo pair and outputs arrays of the three coordinates X, Y, and Z of the image pixels.

The sets of points measured by stereo are accumulated over time into a terrain map centered at the vehicle which is used by the navigation system to drive the vehicle. Since all the data is accumulated in the terrain map, this system does not deal with dynamic scenes which would require explicit identification of moving objects.

3.1 Image Acquisition

We mounted the cameras on a mast to satisfy a number of imaging requirements. Here we describe three of the more important requirements.

1) *Lookahead distance*: The cameras must look far enough ahead to allow the robot enough time to stop or maneuver around an obstacle. The stopping distance is the sum of the distance traveled before braking and the distance traveled while braking:

$$d_{stop} = t_{stop} v + v^2 (2\mu g)^{-1}$$

where t_{stop} is the braking time, v is the velocity of travel, μ is the coefficient of sliding friction, and g is gravitational acceleration. We estimate the braking time to be about 2 sec by summing the times required to acquire stereo imagery, compute stereo disparities from the images, and detect an obstacle from the stereo disparities. Assuming that the vehicle travels at the maximum velocity of 0.7 m/s, and a coefficient of sliding friction of 0.25, the stopping distance is 1.5 m from the front wheels. This constrains the camera height to be at least 66 cm, the height at which the line of sight just grazes the top of the wheels.

2) *Width of field of view*: The cameras must see at least 3 vehicle widths (about 400 cm), at all distances beyond the lookahead distance, so that the Ratler can maneuver around obstacles one vehicle width in size.

3) *Resolution*: An obstacle 20 cm tall must subtend at least 6 pixels in order to be reliably detected.

Given these requirements, we identified the key variables to be camera height, camera baseline, width of field of view, and tilt angle. We performed trade-offs on these variables. The analysis is complicated due to conflicting requirements. For example, raising the cameras increases the width of field of view (good), but decreases the resolution (bad). After extensive simulation and experimentation, we converged on a camera height of 1.5 m, a baseline of 0.9 m, and a tilt angle of 25 deg down from horizontal.

To maximize image stability as the rover traverses surface irregularities, we designed and built a motion-smoothing four-bar linkage that averages the pitch of the two Ratler bodies. This linkage has proven to be extremely valuable in providing reasonably overlapped images while traversing extreme terrain.

3.2 Stereo Matching

We denote by x and y the axis of coordinates of the image plane, y being vertical. We assume in the stereo matching that the epipolar lines are the scanlines of the images so that, given a pixel (x_r,y_r) in the right image, we need to search for the best matching pixel (x_l,y_l) in the left image such that $x_l = x_r + d(x_r,y_r)$ and $y_l = y_r$, where $d(x_r,y_r)$ is the disparity at (x_r,y_r). In order to ensure that the epipolar lines are correctly aligned with the scanlines, we use a rectification procedure developed by Robert [14]. The rectification is applied to the input images and all the algorithms described below are applied to the rectified images.

The best disparity $d(x,y)$ is computed by finding the maximum over d of the normalized correlation $C(x,y,d)$ [5]:

$$\frac{\sum_{(x,y)\in W} I_l((x+d),y) I_r(x,y) - \frac{\left(\sum_W I_l(x+d,y)\right)\left(\sum_W I_r(x,y)\right)}{n^2}}{n}}{\sigma_l \sigma_r}$$

In this expression, W is the window $[x-w_x,x+w_x][y-w_y,y+w_y]$, $\sigma(x,y)$ is the standard deviation of the intensity values in W and n is the number of pixels in the window, $n = (2w_x+1)(2w_y+1)$. Since all the computations are referenced to the right image, we will simplify the notations by dropping the index r and by denoting the pixel position simply by (x,y). We will denote by $C(x,y)$ the correlation value for the best disparity $d(x,y)$.

In order to have disparity resolution better than the image resolution we use a parabolic interpolation that uses the correlation values of the two closest disparities.

We chose to use the normalized correlation criterion over the sum of squared differences (SSD) for two reasons. First, the normalized correlation enables us to avoid the use of a LOG filter to remove photometric differences between the images. Second, the normalized correlation $C(x,y)$ provides a natural measure of confidence of the disparity value at pixel (x,y). Moreover, the increase in computation time compared to the SSD criterion is minimal.

Area-based stereo is a well-established technique but it is known to produce a potentially large number of false matches due to lack of texture, occlusions, and repetitive patterns. It is especially important to be able to filter out those false matches in the context of navigation applications because even a few erroneous points out of hundreds of images may have catastrophic consequences.

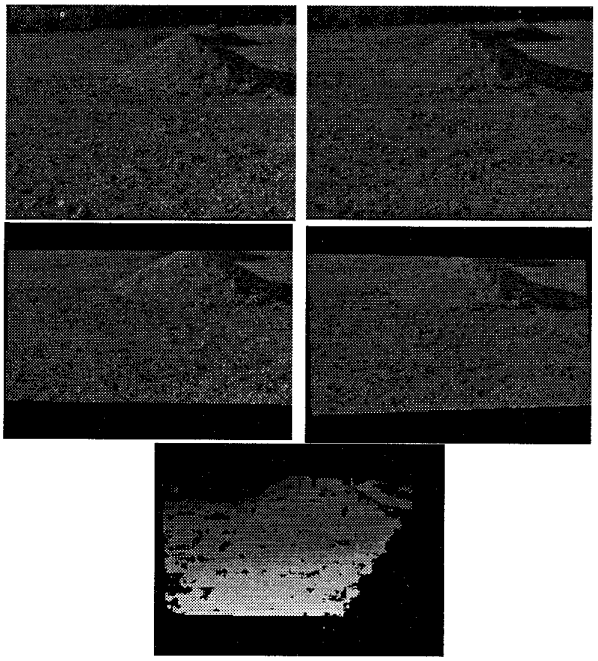

Figure 1: Typical input stereo pair (top); Rectified images (middle); Disparity map (bottom)

In order to achieve the level of reliability required for navigation, we use four types of filtering. The first two types use thresholds $\sigma_{min}(x,y)$ and $C_{min}(x,y)$ on the standard deviation of the distribution of intensity in the neighborhood of (x,y) and on the best correlation at (x,y), respectively. These classical filters eliminate the low-textured areas and part of the occluded areas.

The third filter is designed to eliminate ambiguous matches. It uses a threshold M on the relative difference between the global maximum of correlation and the second best peak in the correlation peak, $C'(x,y)$. Specifically, a point is rejected if $(C(x,y) - C'(x,y))/C(x,y)$ is lower than M. This test is effective in discarding pixels at occlusion boundaries and ambiguous matches due to repetitive patterns. The last filter is a median filter on the disparity map.

Figure 1 shows a typical stereo pair from our test site, the corresponding rectified pair, and the disparity map. The disparity map is computed from 2m to 14m at full resolution although we will show a more efficient use of the stereo matcher below. Except for the resolution, the parameters used for this result and for all the navigation experiments are as follows: 640 columns, 480 rows, $w_x = 25$, $w_y = 17$, $\sigma_{min} = 2$, $C_{min} = 0.5$, and $M = 10$ percent.

After the stereo matching, $d(x,y)$ is converted to a 3-D point $P = (X(x,y),Y(x,y),Z(x,y))$ by using the projective transformations P_l and P_r between the two images and a coordinate system referenced to the vehicle. P_l and P_r are computed by combining the rectification matrices with the calibration matrices computed using a standard calibration procedure [15]. This coordinate system in which the points are expressed is set up so that the Z axis is up, the Y axis is the direction of travel of the vehicle, and the origin is at the base of the mast supporting the cameras. In the remainder of the paper, Cartesian coordinates are expressed with respect to this vehicle-based coordinate system.

We predicted the Cartesian coordinate errors based on a one pixel error in disparity using the current camera configuration. For targets from 0 to 20 meters, the Y error increases quadratically to almost 60 cm at 20 m, while the errors in X and Z do not exceed 6cm. This suggests that our stereo gives a precision level that is comparable with the precision of laser range finders used for navigation.

3.3 Window Selection

In the previous section we briefly described the stereo matching and filtering techniques that we selected. In this and the following section, we describe in detail the improvements that we added to the basic stereo algorithms in order to make them usable in a practical navigation application requiring computational efficiency, robustness, and precision.

In order to apply correlation-based stereo to a practical navigation system, we are faced with a difficult challenge. We need to process the images at a speed high enough to sustain continuous motion of the vehicle while retaining maximum precision on the disparity estimates. Two ways of addressing this challenge have been proposed in the past. First, the image can be processed at a coarser resolution [13][16]. In this case, the processing time can be reduced arbitrarily by decreasing the resolution of the images until it matches the needs of the application. However, the quality of the resulting maps is degraded because the resolution of the disparity values decreases with the resolution of the image.

A second approach is to use special-purpose hardware in order to perform the correlations. This solution has led to several "real-time" stereo systems using DSP [4] or Datacube systems [13]. However, for our application, considerations of cost, power, and availability limit us to conventional computing. Moreover, even though the stereo processing itself is fast in this approach, it produces massive amount of data, the processing of which introduces unnecessary overhead. Although some of the shortcomings of those solutions are specific to our applications, we believe that they do apply to some extend to any practical vision system for which cost, speed, and precision are all issues of equal importance.

A better way of approaching the problem is based on the observation that an autonomous driving system needs to process a small subset of the image, as long as that subset is selected judiciously as function of the speed of the vehicle and of the geometry of the camera system. Intuitively, the vision system needs to process only the subimage that adds information to the map built from all the previous images. As we will see, the size of the subimage is quite small, assuming that the system is in steady state and that the speed is approximately constant. A complete theory supporting this observation was developed by Kelly [10].

We exploit this property in the following way: the planner computes the interval $I_Y = [Y_{min}, Y_{max}]$ of distance from the vehicle within which it needs data in order to expand its map. I_Y is computed from the speed of the vehicle and from the anticipated delay in getting the result from stereo. The algorithm for computing I_Y was introduced by Kelly [9].

After receiving a new value of I_Y from the planning module, the stereo module computes the bounds of a subimage, (y_{min}, y_{max}), and the corresponding bounds in disparity (d_{min}, d_{max}). This computation requires some assumptions about the geometry of the environment because I_Y constrains only the region of interest in the (X,Y) plane but not along the Z axis. Specifically, we assume maximum and minimum heights of the objects in the scene, Z_{min} and Z_{max}. That is not to say that objects of larger size cannot be detected but that only the parts of the objects between Z_{min} and Z_{max} are computed, which is sufficient for navigation purposes.

The bounds are computed by first finding the y coordinates of the pixels in the right image such that x=0 or $x = dim_x$ and $Y = Y_{\{min,max\}}$ and $Z = Z_{\{min,max\}}$. These points are at the boundary of the region of interest. The y coordinates can clearly be computed by solving for y and X at each of the boundary points. The minimum and maximum values of the y coordinates are the vertical bounds of the subimage. Once their y and X coordinates are computed, the boundary points are also projected in the left image and the min and max differences between the x coordinates of their projections in the left and right images are d_{min} and d_{max}.

Figure 2 shows the stereo computation time as a function of $\Delta Y = Y_{max} - Y_{min}$ for Y_{min} = 5m. The computation times are normalized with respect to the largest value. The reduction in computation time compounds two effects: first, the size of the subimage is reduced, thus decreasing the number of pixels processed; second, the disparity interval is reduced and, correspondingly, the number of steps in the correlation search.

Figure 3 shows the average of the min and max value of the requested image rows, the disparity, and the corresponding Y coordinates. These values were computed by averaging

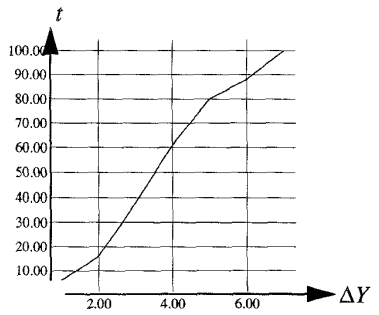

Figure 2: Stereo computation time as function of the width in Y of the desired interval for Ymin = 5m

	min	max	Δ
y (pixels)	130	233	103
d (pixels)	44	118	74
Y (meters)	4.2	7.5	3.3

Figure 3: Average values of disparity range, subimage size and distance interval recorded during typical run

the values in the planning requests recorded during a 150m (310 images) run of the system with the vehicle moving at 0.15m/s.

After an initial adjustment phase, the values in the planning requests remain within 10% of the average values shown in Figure 3. These values confirm that only a small fraction of the image needs to be processed once the navigation system is in a steady state.

3.4 Partial Subsampling

Further reduction of the computation time may be achieved by observing that it is not necessary to process the data at full resolution. More precisely, the resolution on the ground of the data points obtained after transformation is too high compared to what is actually needed for evaluating terrain traversability. For example, the distance in X between consecutive points on a scanline at full resolution at 10m is on the order of 5cm, whereas a 25cm grid is typically sufficient for evaluating navigability. This suggests that it is advantageous to subsample the image in order to process fewer pixels while retaining enough data density for navigability evaluation.

This idea has been used successfully in navigation systems using laser range finders [5][9]. In the case of stereo, we have to be more cautious because simple subsampling will automatically degrade the resolution of the disparity. In other words, we want less data but not at the cost of less accurate data.

We modified the correlation algorithm to get the best compromise between speed and precision by using a partial subsampling in which the disparity is evaluated on a subsampled set of pixels but in which the disparity at a given pixel is searched by using the maximum resolution. We call this approach partial subsampling.

Let δx and δy be the sampling steps in columns and rows respectively, i.e., the disparity is computed only at the pixels (x_r, y_r) such that $x_r = k\delta x$ and $y_r = k\delta y$. We assume that the window sizes w_x and w_y are multiples of δ_x and δ_y respectively. The window W over which the sums are taken in $C(x_r,y_r,d)$ is now defined as:

$$W = \{ (x, y) \in W_f \mid (x = x_r + k\delta x), (y = y_r + k\delta y) \}$$

where W_f is the full resolution window:

$$W_f = [x_r - w_x, x_r + w_x] \times [y_r - w_y, y_r + w_y]$$

In the expression of $C(x_r,y_r,d)$, only the sum of products has to be recomputed for each disparity since the mean and standard deviation at each pixel do not depend on d and therefore can be computed only once. For each disparity d in $[d_{min},d_{max}]$, the products $I_l(x_r+d,y_r)I_r(x_r,y_r)$ are computed for all the subsampled values of (x_r,y_r). Let us denote the sum of products at (x_r,y_r) over W_f by $S(x_r,y_r)$, and the sum of products over column x_r only by $S_y(x_r,y_r)$:

$$S(x_r, y_r) = \sum_{(x,y) \in W} I_r(x, y)\, I_l(x + d, y)$$

$$S_y(x_r, y_r) = \sum_{y \in [y_r - w_y, y_r + w_y]} I_r(x_r, y)\, I_l(x_r + d, y)$$

Then $S(x_r+\delta x,y_r)$ is computed recursively from $S(x_r,y_r)$, $S_y(x_r-w_x,y_r)$, and $S_y(x_r+w_x+\delta x,y_r)$ as

$$S(x_r + \delta x, y_r) = S(x_r, y_r) + S_y(x_r + w_x + \delta x, y_r) - S_y(x_r - w_x, y_r)$$

This shows that, for a given disparity d, the map $C(x_r,y_r,d)$ can be computed recursively at the reduced resolutions $(\delta x,\delta y)$, thus reducing the computation required by $\delta x.\delta y$. At the same time, the resolution in disparity is maintained by computing the $C(x_r,y_r,d)$ for all the values of d, without subsampling it. Therefore, the partial subsampling, that is, subsampling the image without subsampling the disparity, does achieve our goal of more efficient stereo matching without loss of precision.

We conducted a series of experiments in order to verify that the precision is not affected by the subsampling. In those experiments, we placed boxes at distances ranging from 2m to 8m from the center of the vehicle by increments of 1m. The front faces of the boxes are parallel to the X axis. We then measured the RMS difference, E_Y, between the measured Y values on the boxes and the true value for different values of δx and δy.

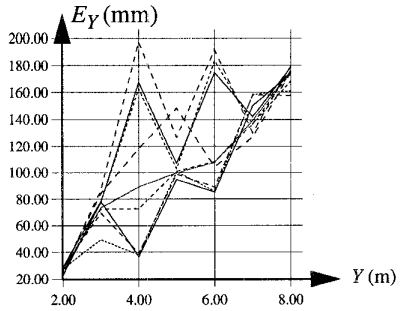

Figure 4: Error in measured distance as a function of distance for different levels of subsampling

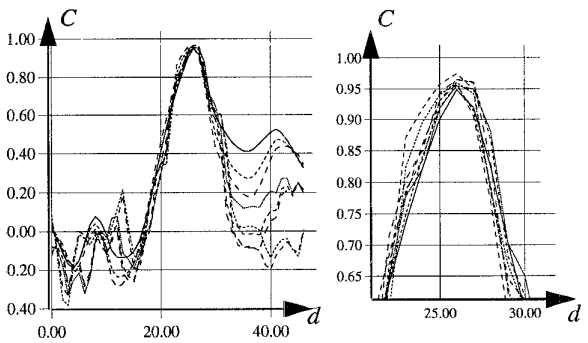

Figure 5: Correlation curves at one pixel for different levels of subsampling $\delta x = \{1,3,5\}$ and $\delta y = \{1,2,4\}$; the left plot is the complete correlation curve; the right plot is a magnified subset of the complete curve around the maximum

Figure 4 shows the plots of E_Y as a function of Y for $\delta x = \{1,3,5\}$ and $\delta y = \{1,2,4\}$ and for Y ranging from 1m to 8m by increments of 1m. These plots show that the error in Y is independent of δx and δy. They also show that the error in Y computed from experimental data is consistent with the theoretical error derived in Section 3.2. For example, the theoretical error at 8m is 10cm, while the actual error is 15cm.

Figure 5 shows the correlation values at one pixel plotted as a function of disparity for nine different combinations of $(\delta x,\delta y)$ for $\delta x = \{1,3,5\}$ and $\delta y = \{1,2,4\}$. The right part of Figure 5 shows the same curves in the vicinity of the maximum correlation. This figure shows that the best disparity is the same at all resolutions and that there is little variation of the correlation values across resolution levels. In contrast, for $\delta x = 5$, the disparity computed using a conventional subsampling technique can be in error by as much as 5 pixels.

In practice we use fixed subsampling factors of $\delta x = 5$ and $\delta y = 4$. With these values, the stereo matching takes 0.7s on average on a Sparc10 workstation using the parameters of Figure 3.

3.5 Performance

The combination of selective windowing and partial subsampling allows us to achieve both the computational speed and the precision required for continuous motion at low speeds using a general-purpose workstation (with no special-purpose hardware).

The longest stereo run took place over 6 hours of intermittent operation (interrupted by rain and battery recharges) as Ratler traversed 1,078 m over the rough terrain of a slag heap. During this trial, the stereo module processed at least 3,000 image pairs, and computed at least 1.5 million three-dimensional points (based on the average values reported in Figure 3).

It is difficult to quantify the accuracy of the maps over the course of kilometer-long runs, because we do not know ground truth. Whenever we performed spot checks of the computed coordinates, they were correct within the precision of our evaluation of ground truth using a tape measure and a large-scale protractor.

The only failures we observed were due to transient effects caused by disconnection of video cables, and by abrupt lighting changes that overwhelm the auto-iris lenses.

4 Positioning

We have formulated and implemented a suite of algorithms for maintaining an estimate of the robot's position and orientation in a fixed, external reference frame.

4.1 Sensors

Ratler includes four types of state sensors:

1. Encoders on the 4 drive motors
2. A flux-gate compass
3. Inclinometers (3) measuring the pitch of the left body, the pitch of the right body, and vehicle roll
4. A turn-rate gyro

The outputs of the sensors are digitized to 10 bits, although historically, telemetry from Ratler has included only 7-bit values.

The raw data from each of the four types of sensor is biased and noisy and sometimes corrupt. Thus, we have found it necessary to develop filters for preprocessing the sensor data. As an example, consider the compass data.

In steady state, the compass signals are corrupted by random noise. We performed a spectral analysis of the data, and observed a cut-off frequency of 0.25 Hz. We implemented several low-pass filters, including Butterworth and Bessel filters. These were extremely effective in suppressing the noise, but also introduced a 2-3 cycle delay between the filtered value and the signal.

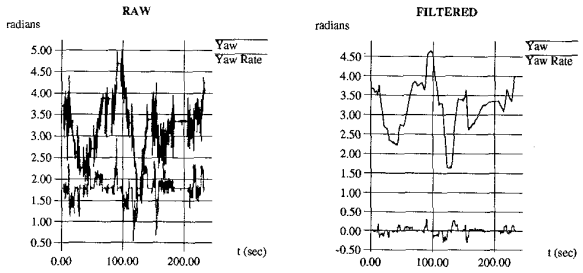

Figure 6: Raw data (left) and Kalman filtered data (right)

When the Ratler accelerates, the compass signals are corrupted not only by random noise, but also by transients caused by the dynamic effects. We implemented a Kalman filter:

$$\hat{x}_{n+1} = A\hat{x}_n + K(z - A\hat{x}_n)$$

with a sensor model in the 'A' matrix that heavily weights the previous compass reading if the turn-rate sensor indicates that the robot is accelerating. Essentially, this filter does not believe the compass when Ratler turns. Figure 6 illustrates the results of applying this filter to the compass (yaw) and turn-rate (yaw-rate) sensors. The improvement is dramatic.

4.2 Position Estimation

Given the previous state and the new state sensor readings, the position estimation algorithm computes the current state of the robot. Let θ represent the filtered heading, let ϕ represent the mean of the filtered readings from the left and right pitch inclinometers, and let Δ represent the mean of the differences between the current and previous encoder readings. Then the new position is given by

$$x_n + \Delta\cos\theta\cos\varphi \quad y_{n+1} = y_n + \Delta\sin\theta\cos\varphi \, z_{n+1} = z_n + \Delta\sin\varphi$$

Figure 7 tabulates the results of tests in which we repeatedly commanded Ratler motions, and logged computed and true positions and orientations. Trial A uses the earliest, simplest generation of the Ratler positioning system at an outdoor slag heap, with straight-line motion only. The positional accuracy is poor, as indicated by the average 17 percent error. The heading accuracy is excellent, and the positional and heading precisions are moderate. Trial B uses (again at the slag heap) an intermediate version of the developed positioning system, for which a number of low-level motion control constants were tuned. The results show that the accuracy in both position and heading is excellent. The precision of the heading estimate appears to have decreased; we believe this is an artifact of limitations in our ability to obtain the ground truth orientation measurements. Trial C uses the positioning system with filtered compass data on a flat, grassy field, and motion along

Trial	Mean Distance	Mean Orient.	StDev Distance	StDev Orient.
A	17.0	0.0	2.0	3.7
B	3.1	0.5	1.3	11.0
C	2.5	---	2.3	---

Figure 7: Statistics of position and orient. errors (percent)

curved arcs. The results show moderate accuracy improvements and moderate precision degradations with respect to Trial B.

5 Discussion

In this paper we have presented the mapping and positioning systems for a prototype lunar rover.

The mapping system consists of a camera head on a motion-averaging mast, which provides image pairs to a normalized correlation matcher. This matcher intelligently selects what part of the image to process (saving time), and subsamples the intensities (saving time) without subsampling disparities (which would reduce accuracy). The mapping system has operated successfully during long-duration field exercises, processing thousands of image pairs.

The positioning system employs encoders, inclinometers, a compass, and a turn-rate sensor to maintain the position and orientation of the rover as it traverses. The system succeeds in the face of significant sensor noise by virtue of sensor modelling, plus extensive filtering and data screening.

Although both the mapping and positioning systems use classical sensors and previously discovered algorithms, they have achieved unprecedented results, enabling long-duration (6 hours) and long-distance (1 km) outdoor traverses. The key contributions are in tailoring the general ideas to a specific robot performing a specific task, and in demonstrating practical and unprecedented performance.

Future work will continue to concentrate on robust, reliable operation in the face of occasionally abysmal sensor data. To enable better performance in avoiding obstacles, we will achieve a wider stereo field of view by replacing the binocular rig with a four-camera setup. To improve position estimation performance we will port all of the preprocessing and processing onto the on-board system. In addition, we will mount an Inertial Measurement Unit on Ratler, and derive position and attitude from the measured accelerations.

6 Acknowledgments

This research was partly sponsored by NASA, under grants NAGW-3863 and NAGW-1175. We gratefully acknowledge assistance from Ben Brown, Michel Buffa, Lonnie Chrisman, Fabio Cozman, Yasutake Fuke, Richard Goodwin, Guillermo Heredia, Paul Klarer, Luc Robert, Yoshikazu Shinoda, and Reid Simmons.

7 References

[1] C. Proy, B. Hotz, O. Faugeras, P. Gernesson, and M. Berthod. Onboard Vision System for a Mobile Planetary Exploration Robot. In Proc. Workshop on Computer Vision for Space Applications, pp. 2-12, Antibes, France, September 1993.

[2] C. Angle and R. Brooks. Small Planetary Rovers. In Proc. IEEE Intl. Workshop on Intelligent Robots and Systems, pp. 383-388, Tsuchiura, Japan, July 1990.

[3] R. Chatila, R. Alami, S. Lacroix, J. Perret, and C. Proust. Planet Exploration by Robots: From Mission Planning to Autonomous Navigation. In Proc. Intl. Conf. Advanced Robotics, pp. 91-96, Tokyo, Japan, November 1993.

[4] O. Faugeras et al. Real Time Correlation-Based Stereo: Algorithm, Implementations and Applications. INRIA Technical Report. 1993.

[5] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press. 1994.

[6] M. Hebert. Pixel-Based Range Data Processing for Autonomous Navigation. Proc. Int. Conf. on Robotics and Automation. San Diego. 1994.

[7] M. Hebert and E. Krotkov. 3-D Measurements from Imaging Laser Radars. Inter. J. Image and Vision Computing. 10(3). April 1992.

[8] M. Hebert and E. Krotkov. Local Perception for Mobile Robot Navigation in Natural Terrain: Two Approaches. In Proc. Workshop on Computer Vision for Space Applications, Antibes, France, September 1993.

[9] 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.

[10] A. Kelly. Adaptive Perception for Autonomous Vehicles. Tech. Report CMU-RI-TR-94-18. Robotics Institute, Carnegie Mellon University, 1994.

[11] 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.

[12] E. Krotkov and R. Hoffman. Terrain Mapping for a Walking Planetary Rover. IEEE Trans. Robotics and Automation, 10(6):728-739, December 1994.

[13] L. Matthies. Stereo Vision for Planetary Rovers: Stochastic Modeling to Near Real-Time Implementation. Intl. J. Computer VIsion, 8(1):71-91, July 1992.

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

[15] L. Robert. Camera Calibration Without Feature Extraction. Proc. Int. Conf. Pattern Recognition. 1994.

[16] B. Ross. A Practical Stereo Vision System. Proc. Computer Vision and Pattern Recognition. New York. 1993.