Inproc. Int'l Conf. on Advanced Robotics
Tokyo, Japan, November, 1993

Results in Terrain Mapping for a Walking Planetary Rover

Eric Krotkov and Regis Hoffman

Robotics Institute, School of Computer Science
Carnegie Mellon University
Pittsburgh, PA. 15213 U.S.A.

Abstract - We present results from a terrain mapping system for walking robots that constructs quantitative models of surface geometry. The accuracy of the constructed maps enables safe, powercfficient locomotion over the natural, rugged terrain found on planetary surfaces. The mapping system acquires range images with a laser rangefinder, preprocesses and stores the images, and constructs elevation *maps* from them *at* arbitrary resolutions, in arbitrary reference frames. The system is one of the few that can handle extremely rugged terrain, and is more robust than any comparable system in its aggressive detection of image-based errors and in its compensation for time-varying errors. Extensive tests in natural, rugged terrain producing hundreds of millions of map points indicate that the mapping system (i) exceeds established performance standards in accuracy, timing, and memory utilization, and (ii) exhibits a high degree of real-world robustness.

1. Introduction

In this paper we present results from a terrain mapping system for the Ambler, an autonomous walking robot configured to traverse boulder-strewn surfaces like those on Mars. The objective of our work in perception for planetary rovers is to develop and demonstrate approaches to terrain mapping that enable locomotion that is (i) safe, and (ii) power-efficient over rugged, natural terrain. To quantify the term "rugged," we designed the Ambler mechanism to meet Mars mission requirements to climb 30 degree slopes with frequent surface features (e.g., ditches, boulders, and steps, existing simultaneously) of up to 1 meter in size.

Safe locomotion is essential to the success of any exploration mission. The rover must be able to detect and avoid hazards in its environment. such as cliffs, or else end the mission by succumbing to them. In addition, to minimize mechanical wear and to preserve its health, the rover must avoid colliding with and stumbling over obstacles.

Power-efficient locomotion is a less obvious requirement, but it too is essential. Power is at a premium for any extraterrestial mission, and planetary rovers face severely limited power budgets. Since power is such a precious commodity, and because every contact with the terrain transfers energy from the robot to the environment, the rover can ill afford to waste energy by bumping into obstacles.

The terrain found on planetary surfaces is rugged and irregular, characteristics that pose significant challenges for autonomous mapping. First, rugged terrain violates the constraints on shape (e.g., symmetry) and surface properties (e.g., smoothness) that established machine perception techniques exploit. Second, irregular terrain resists the geometric modeling required by model-based vision approaches. Finally, the natural environment does *not* provide the controlled lighting, fixtured objects, or other simplifications of the laboratory.

Researchers have advanced a spectrum of terrain mapping and obstacle detection concepts for outdoor robotic operations. The literature reports a variety of different sensors, control regimes, and algorithms for different environments [1][3][4][5][10][12]. We find no mapping system suitably dense and accurate for autonomous and powercfficient legged locomotion through terrain as rough and rugged as that found on Mars (cf. Section 1).

To meet the needs of safe and power-efficient locomotion over rugged, natural terrain, we have developed a mapping system that efficiently builds quantitative models of terrain geometry, making explicit the spatial layout of the environment. The rover successfully uses these models to select footholds that avoid hazards, and to plan motions that conserve power by minimizing terrain contact. The mapping system is one of the few that can handle extremely rugged terrain, and is more robust than any comparable system in its aggressive detection of image-based errors and in its compensation for time-varying errors.

The plan of the paper is as follows. In the next section we review related research. In Section 2 we describe the Ambler walking robot, and in the next sections we discuss sensing and calibration, the terrain mapping system, and experimental results. In Section 6 we present a method of compensating for time-varying elevation errors. We conclude the paper with a *summary* and critical discussion of the approach.

2. Ambler

The Ambler is a walking robot (Figure 1) designed to satisfy constraints characteristic of exploration missions to planetary surfaces. The key constraints are to traverse extreme terrain, to minimize power consumption, and to provide a stable platform for imaging, scientific, and sampling equipment. Readers interested in the Ambler configuration will find detailed discussions elsewhere [2][11].

3. Sensing and Calibration

The primary sensor is a scanning laser rangefinder, which measures both reflectance and range. We use a laser scanner

Figure 1 The Ambler

because it directly recovers the environment's three-dimensional structure. supplying 3-D data more rapidly and reliably than passive vision techniques such as binocular stereo and structure-from-motion. Recent advances (cf. Section 2) make stereo far more feasible for real-time rover applications than it was in 1987, when this effort began.

The particular sensor, manufactured by Perceptron, Inc., acquires data in 256x256 pixel images at a rate of 2 Hz. The sensor is an amplitude-modulate continuous wave device that measures range from the phase difference between transmitted and reflected signals. It digitizes to 12 bits over approximately 40 m, providing a range resolution of approximately 1 cm, and a range precision of 10-15 cm. The measurements cover 60 deg in azimuth and 60 deg in elevation. The scanner, mounted on top of the Ambler, looks down directly in front of the robot.

We have developed an automatic calibration procedure that identifies the transformation between the Ambler-centered reference frame and the sensor-centered frame [8]. The procedure moves the leg to various positions within the scanner field of view, and processes the reflectance image to locate the leg. The procedure uses the leg positions in the images and the leg positions in the body frame to compute the transformation that minimizes the distance between pairs of corresponding points referred to the Ambler-centered frame.

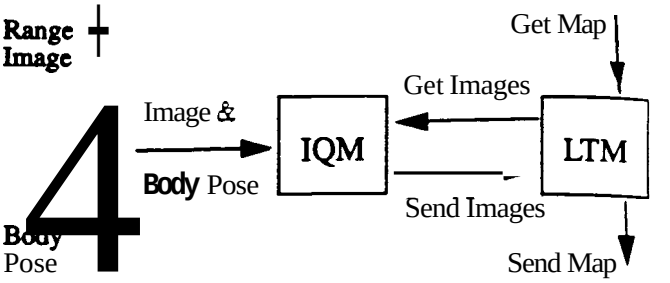

Figure 2 Mapping system modules

From experiments with thousands of points, we conclude that the accuracy of the computed transformation is 2-7 cm with a precision no lower than 2-5 cm. These results have proven to be Satisfactory for constructing terrain maps and using them to select footholds during our rough terrain walking experiments.

4. Mapping System

The purpose of the mapping system is to construct models of the local terrain from range images. The system consists of three major modules that acquire images, preprocess and store images, and construct terrain elevation maps (Figure 2). Approximately ten other modules perform data analysis and display tasks, each executing concurrently. Many different system configurations, including this one, suffice to meet the perception needs identified in Section 1. This configuration is modular, extends easily to more and different sensors, decouples imaging and mapping, and manages large quantities of data.

4.1. Image Acquisition

Typically, new information appears in the field of view only after the body moves. We have designed the walking system so that when a move completes, the Perception Imaging Manager (PIM) receives a command to initiate image acquisition.

The PIM acquires frames of range and reflectance data, selecting either 1) real images from the rangefinder, 2) virtual images stored previously, or 3) synthetic images computed by a 3-D graphics simulator. Options 2 and 3 have been valuable for debugging and development. The PIM tags each image pair with robot state information (the body pose and six leg positions), the transformation relating the scanner frame and world coordinate frames (composed from transformations derived from the calibration procedure and the Ambler's dead-reckoned pose), and the parameters that convert a raw range measurement into units of meters. Then, the PIM transmits the tagged images to the Image Queue Manager.

4.2. Image Preprocessing and Storage

The Image Queue Manager (IQM) preprocesses the images, and then stores them. Preprocessing is required to compensate for both known and unexpected problems.

Certain defects in the range measurements occur within known image regions. These defects include variations in range values caused by differences in the optical coating on the exit window, and artificially low range values in the bottom corners of the images caused by the exit aperture. Other defects in the range measurements occur at unknown image locations. These defects include artificially small range values for materials that poorly reflect the laser energy. and for depth discontinuities at the right-hand side (from the sensor's point of view) of objects. Still other effects (not defects, properly speaking) can disqualify range measurements. Such effects include viewing objects lying at distances beyond the sensor ambiguity Interval, and viewing non-natural objects, such as the robot's legs

An aggressive preprocessing stage identifies pixels corrupted by the known defects and the known but unpredictable effects, and marks them as invalid. Readers interested in the detailed processing steps will find them elsewhere [6][7]. In addition. the preprocessing stage applies the Canny operator to detect range discontinuities, which the mapping algorithms will use to identify range shadows and occluded regions.

When queried for an image sequence, the IQM could return all images. However, this requires substantial data transfer: the size of the images is large (256K per pair, plus variable-size auxiliary structures such as edge images), and many images may be required (with typical queries for I m^2 areas, 5 are required for a straight-line trajectory on flat terrain, and 25 are required for a point turn over obstacles). Instead of returning all images, the IQM returns some subimages. viz., those that intersect a given region of interest polygon. In typical operation, this reduces data transfer by 75-90 percent.

4.3. Terrain Mapping

The Local Terrain Mapper (LTM) constructs elevation maps, which serve as the primary terrain representation. They are well-suited for representing natural, rugged terrain, and can be accessed simply by specifying the boundary of a region Of Interest.

External modules request elevation maps. specifying (i) a polygonal region, (ii) a resolution, and (iii) an arbitrary reference frame to be used. Typical reference frames are the current body frame, a future body frame (to allow advance planning), and the world frame. To construct the requested maps, the LTM first queries the IQM for a sequence of relevant subimages. Then the LTM uses the subimages, starting with the most recent, to assign one of three labels to each elevation point:

- *unknown*, for those cells outside the polygon or outside the rangefinder field of view or with no valid image pixels.
- occluded, for those cells occluded by other objects. In this case, the LTM uses knowledge of the viewing geometry to compute and store an upper bound on the elevation.
- *known*, for all other cells. In this case, the LTM computes and stores the terrain elevation with the Locus Method [9], an efficient algorithm for transforming and interpolating range data from the Sensor frame into Cartesian coordinates.

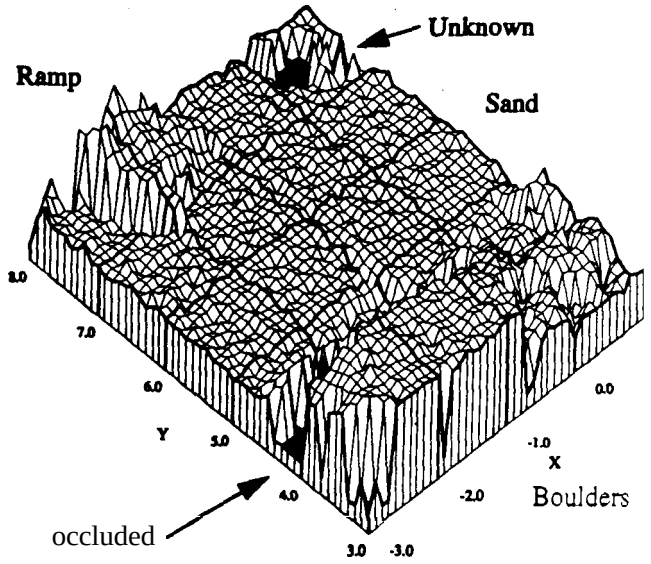

Figure 3 Elevation *map* computed from range images

Finally, the LTM sends the map to the requesting module. Figure 5 illustrates an elevation map containing many points labelled *known*, and few points labelled *occluded* or *unknown*.

4.4. System Operation

Figure 4 illustrates the concurrent execution of the three main modules. It shows that when the Ambler completes a body move, the PIM acquires an *image*, then sends it to the IQM, which preprocesses and stores it. When the LTM receives a *map* request, it queries the IQM for a list of relevant subimages. and uses the list to compute the map. If the body moves while the LTM is active, as shown in Figure 4, the PIM is free to initiate the image acquisition and storage process.

5. Results

We have tested the mapping system extensively in real-world experiments over a three-year period. In this section, we present the results, beginning with qualitative examples, and then presenting quantitative performance statistics.

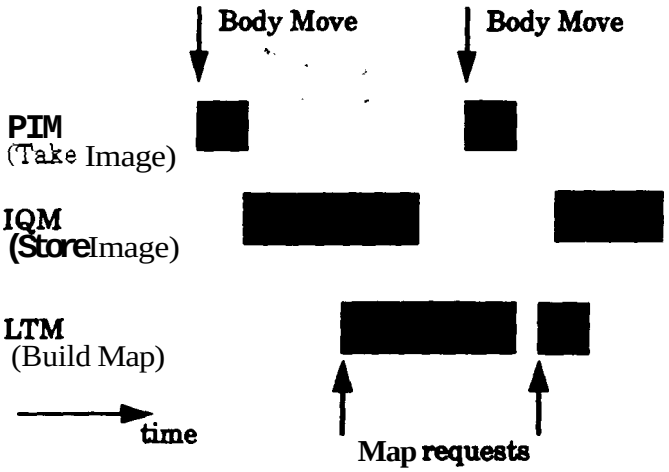

Figure 4 Concurrent operation of mapping system

5.1. Maps Constructed in Field Trials

We conductrd field trials in three different environments: an indoor obstacle course. an outdoor field, and a parking lot obstacle course. In this section, we describe the different &vironments and their physical challenges, and present maps constructed from range images acquired during the trials. However, we note that these maps are for human inspection; the Ambler planning modules use smaller scale versions of the maps shown.

For indoor trials, the Ambler operated on obstacle courses fashioned from 40 tons of sand, 20 tons of boulders, a 30 deg wooden ramp, and various other objects. The courses typically include rolling, sandy terrain with several boulders 1 m tall, ten or so boulders 0.5 m tall, a ditch, and a ramp. The largest of the boulders is 1.5 m tall, 4 m long, and 2 m wide.

In one indoor trial, the Ambler took 397 steps and traveled about 107 meters following a figure-eight pattern, each circuit of which covers about 35 m and 550 deg of turn. From the images acquired during this trial, the mapping system constructed a composite map of the environment (Figure 5) that clearly captures the key environmental features.

For one set of outdoor trials, the Ambler operated on rolling, grassy terrain (Figure 1). The site is a field cleared in a hilly, wooded area at an industrial park. Although it does not contain obstacles like those encountered in the indoor trials, the site poses its own challenges: steeper slopes and si&-slopes, soft ground, 30 degree F temperature variations, lighting conditions varying from bright sunshine to partly cloudy to dusk In the longest of these trials, the Ambler took 1219 steps, traveling 527 m horizontally and 25 m vertically. It followed a meandering course that included first climbing a hill and descending from it, then roughly following an iso-elevation contour for 250 m, executing a point turn of roughly π radians, and following an iso-elevation contour back to the starting region.

From the images acquired during this trial, the mapping system produced a comprehensive site map. Comparing the computed map to the site topography, we find excellent agreement

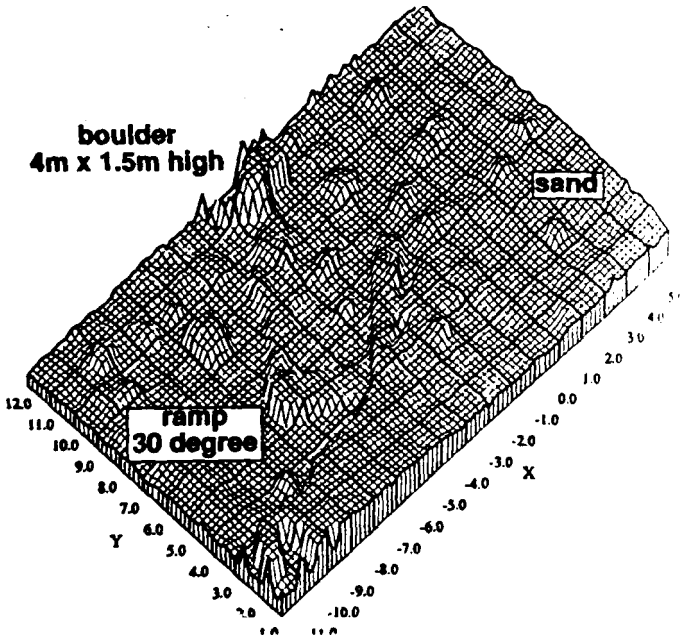

Figure 5 Map constructed of indoor obstacle course

5.2 Performance Statistics

Table 1 records mapping system statistics gathered from several representative walking trials. The distance term represents the distance traveled by the robot during the trial. The results vary because the terrain and software modules vary between trials. For example, in the 46 m run, a module planned body recovery motions, computing more map points than in the 68 m run, during which that module did not execute.

The distances traversed indicate the effectiveness of the system in accurately capturing the terrain geometry. The large vol-

D	I	S	Maps	Points
46	151	8,974	626	375,003
68	185	10,015	640	267,104
107	397	20,286	1,306	597,078
527	1219	101,129	4,716	2,578,102

D is the distance travelled (meters). I is the number of images acquired. S is the number of subimages transferred. Maps is the number of maps built. Points is the number of computed elevation values.

Table 1. Mapping system statistics

ume of data processed, both images and maps, indicates the robustness of the system.

Over these trials, we find that taking a single step requires between 20 and 30 sec of mapping system processing, which involves examining between 5 and 28 range subimages in order to compute elevation values for some 1700 points, on average. Further analyzing the timing, we find that for Sparc 2 workstations, image acquisition takes about 1 sec, image processing and storage requires about 10 sec, and construction of elevation maps takes about 5 msec per point.

One measure of map accuracy is the difference between the computed elevation and the elevation determined by encoders on the Ambler's legs. According to this measure, we find that the map accuracy varies from step to step and from trial to trial, but never strays far from the scanner precision of 10-15 cm, which is sufficient for the Ambler's 30 cm in diameter feet.

As the Ambler walks, images and maps consume more and more memory. The mapping system handles this with memory management routines that detect memory saturation, and then reclaim storage for new use and delete less recent data. In one typical trial, after acquiring 100 images, memory usage levels peaked at 1.6 Mb for PIM, 2.8 Mb for LTM, and 30.5 Mb for IQM. After this, the mapping system maintained nearly constant memory utilization, thus demonstrating the effectiveness of memory management.

6. Elevation Error Compensation

During the walking trials, we find that the elevation map errors can vary substantially. We have identified three principal causes of these variations:

1. Change of temperature. Range values returned by the rangefinder change 1 cm per degree F (we reported a 10 cm per degree drift [6], but sensor enhancements have reduced this). As the temperature changes, so do the range values, which in turn change the computed elevations.

2. Change of terrain type. Range values returned by the rangefinder vary with the reflectance properties of sensed objects. For an oversimplified example, darker objects appear to the Sensor to be farther away than lighter objects. As the terrain type changes. say by walking from grass into soil. the reflectance properties change, causing the range values to change, causing the computed elevations to change.

3. Change of terrain slope. Terrain perpendicular to the incident Sensor laser beam will return a stronger (and hence less noisy) signal than terrain at a smaller angle of incidence. As the slope changes, so does the sensed range, in turn causing the elevations to change.

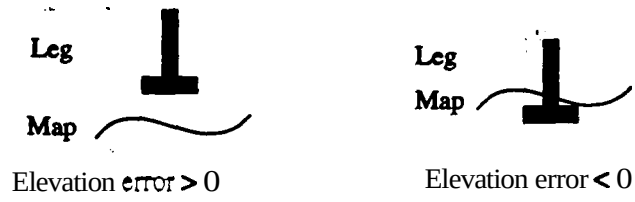

Figure 6 Elevation error

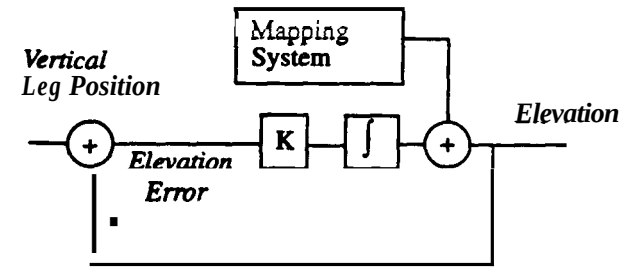

Figure 7 *Closed* loop control of elevation values

During the 500 m walking experiment, the Ambler encountered all three types of changes: temperature changes of 30 degrees F, transitions from wet to dry grass, and slopes from 0 to 15 degrees. It is therefore not surprising to observe significant elevation variations in the computed maps.

To characterize this elevation *map* variation, we use the position of the robot legs when in ground contact as a measure of ground truth. Each time the robot takes a step, we compute the elevation error as the difference between the vertical leg position and the elevation value stored in the map. If the elevation error is positive, the foot is above the sensed terrain, and if the *map* error is negative, the foot is below the sensed terrain (Figure 6).

To adjust the elevation values, we implement a control loop (Figure 7) that uses the elevation error as an error signal to increase the accuracy of the elevation maps. The control law

$$\delta_z = \sum_i k\,(z_{i,\,leg} - z_{i,\,map})$$

where i indicates the step number, includes a proportional tam, with gain k typically set to 0.1, and an integral term, to reduce the steady-state error. To adjust the elevation values, the system adds the vertical displacement δ_z determined by the control law to the vertical component t_z of the translation term in the transformation matrix determined by the calibration procedure.

Figure 8 plots the elevation error for 371 steps taken over an 8 hour period, during which the robot travelled 172 m. During the open loop portion of the experiment, the elevation values drift up and then down. For open loop operation. we intentionally aim for a negative elevation error so that when the planning modules add safety margin offsets, the commanded leg moves

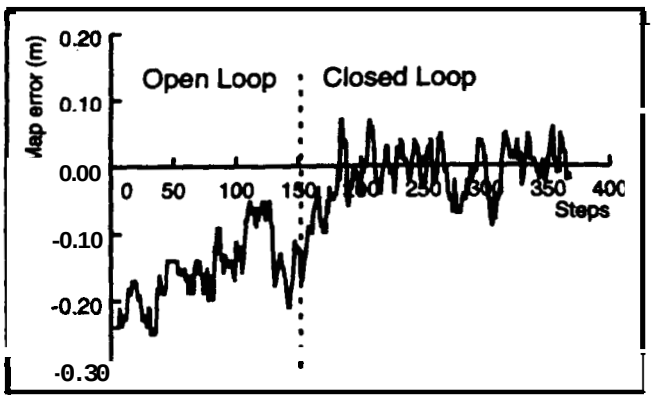

Figure 8 Elevation error for open and closed loop control

will terminate slightly above the terrain, thus preventing unexpected ground contact. When closed loop operation begins, the controller drives the mean elevation error to near zero. Note that the 10-15 cm variations due to sensor imprecision do not disappear.

There are two consequences of the closed loop operation, both demonstrated during the 500 m trial. First, decreasing the map error to near zero increases mobility, by permitting a greater range of terrain slopes to be successfully traversed. Second, leg recovery (lifting a leg, moving it between the two stacks, and planting it on the ground) is more power efficient, because the rover need not lift the leg artificially high above the ground.

7. Discussion

In this paper, we have presented a terrain mapping system for the Ambler suitable for robotic exploration of the natural, rugged terrain found on planetary surfaces. The mapping system acquires range images with a laser rangefinder, preprocesses and stores images, and constructs elevation maps from them at arbitrary resolutions, in user-specified reference frames. Extensive tests in natural, rugged terrain indicate that the mapping system exceeds established performance standards in accuracy, timing, and memory utilization, and exhibits a high degree of real-world robustness. In summary, the results show that the implemented mapping system constructs quantitative models of terrain geometry that enable safe, power-efficient locomotion in rugged terrain.

Our future work will continue to address the theme of robust three-dimensional sensing. We will develop stem techniques, including multi-baseline stereo with up to five cameras, for walking in rugged terrain. We aim for robustness and reliability in the presence of false solutions to the correspondence problem, imperfect registration of the cameras, and inaccurate knowledge of the calibration parameters. In addition, we will investigate techniques for detecting and recovering from hardwart faults such as memory parity errors and operating system mors.

Acknowledgments

Many members of the Planetary Rover project have contributed to the results reported here. In particular, we acknowledge the efforts of Kenichi Arakawa, Mike Blackwell, Martial Hebert, Takeo Kanade, In So Kweon, Reid Simmons, and Red Whittaker. This research was sponsored by NASA under grant NAGW-1175.

References

[1] 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.

[2] J. Bares and W. Whittaker. Configuration of Autonomous Walkers for Extreme Terrain. Intl. J. Robotics Research To appear. 1993.

[3] L. Boissier, B. Hotz, C. Roy. O. Faugeras, and P. Fua. Autonomous Planetary Rover: On-Board Perception System Concept and Stereovision by Correlation Approach. In Proc. IEEE Intl. Conf. Robotics and Automation pp. 181-186, Nice. France, May 1992.

[4] M. Daily. J. Harris, and K. Reiser. An Operational Perception System for Cross-Country Navigation. In Proc. Image Understanding Workshop Cambridge, 1988.

[5] P. Fua. A Parallel Stereo Algorithm that Produces Dense Depth Maps and Preserves Image Features. Technical Report 1369, INRIA, Sophia-Antipolis, France, January 1991.

[6] M. Hebert and E. Krotkov. 3-D Measurements From Imaging Laser Radars: How Good Are They? Intl. Journal of Image and Vision Computing 10(3):170-178, April 1992.

[7] R. Hoffman and E. Krotkov. Perception of Rugged Terrain for a Walking Robot: True Confessions and New M o n s . In Proc. IEEE Intl. Workshop on Intelligent Robots and Systems. pp. 1505--1511, Osaka, Japan. November 1991.

[8] E. Krotkov. Laser Rangefinder Calibration for a walking Robot. In Proc IEEE Intl. Conf. Robotics and Automation pp. 2568-2573, Sacramento. California April 1991.

[9] I.S. Kweon and T. Kanade. High-Resolution Terrain Map from Multiple Sensor Data. IEEE Trans. Pattern Analysis and Machine Intelligence 14(2):278-292, February 1992.

[10] L. Matthies. Stem Vision for Planetary Rovers: Stochastic Modeling to Near Real-Tim Implementation. Intl. J. Computer Vision 8(1):71-91. July 1992.

[11] R. Simmons. E. Krotkov, W. Whittaker, B. Albrecht, J. Bares, C. Fedor, R Hoffman, H. Pangels, and D. Wettergreen. Progress Towards Robotic Exploration of Extreme Terrain. Journal of Applied Intelligence 2:163-180, Spring 1992.

[12] C. Thorpe, editor. Vision and Navigation: the Carnegie Mellon Navlab Kluwer, 1990.

[13] B. Wilcox, L. Matthies, D. Gennery, B. Cooper, T. Nguyen, T. Litwin, A. Mishkin, and H. Stone. Robotic Vehicles for Planetary Exploration. In Proc. IEEE Intl. Conf. Robotics and Automation pp. 175-180, Nice. France, May 1992.