Terrain Mapping for a Walking Planetary Rover

Eric Krotkov, *Member, IEEE,* and Regis Hoffman

***Abstract*—We present a terrain mapping system for walking robots that constructs quantitative models of surface geometry. The accuracy of the constructed maps enables safe, power-efficient 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. To quantify performance in terms of accuracy, timing, and memory utilization, we conducted extensive tests in natural, rugged terrain, producing hundreds of millions of map points. The results indicate that the mapping system 1) is one of the few that can handle extremely rugged terrain, and 2) exhibits a high degree of real-world robustness due to its aggressive detection of image-based errors and in its compensation for time-varying errors.**

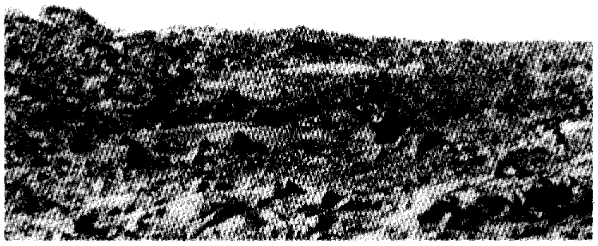


Fig. 1. Mars terrain viewed by the Viking lander.

I. Introduction

IN THIS PAPER we present 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: (1) safe, and (2) 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 one meter in size [19].

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 craters. 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 extraterrestrial 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 colliding with obstacles.

The terrain found on planetary surfaces is rugged and irregular (Fig. 1), 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.

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 appears to be 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 III 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 VII we present a method of compensating for time-varying elevation errors. We conclude the paper with a summary and critical discussion of the approach.

II. Related Research

Researchers have advanced a spectrum of terrain mapping and obstacle detection concepts for outdoor robotics operations. In this section we review perception-based systems that have been implemented and tested in realistic outdoor settings. We restrict our remarks to land systems, excluding perception systems developed for underwater or air vehicles, and we concentrate on perception of surface geometry for the sake of locomotion rather than for the sake of position estimation. We begin with outdoor perception techniques for planetary rovers, then address unmanned ground vehicles, and then consider other outdoor mobile robots.

Sustained effort in the planetary rover domain at NASA's Jet Propulsion Laboratory [27], [28] has developed a family

Manuscript received June 7, 1993; revised March 29, 1994. This research was supported in part by NASA under Grant NAGW-1175.

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA.

IEEE Log Number 9405741.

of outdoor perception techniques based on stereo mapping and, currently under development, structured light sensing. Matthies *et al.* [20] describe a stereo system enabling Robby, a six-wheeled vehicle, to achieve up to 100 m of autonomous travel in a cross-country test area in a desert arroyo. The vehicle acquires images from two cameras with a 25 cm baseline and processes them while stationary (start/stop mode). The system computes the height of scene points from the disparity images, and detects obstacles by thresholding the heights. These results establish a new level of practicality of stereo vision for outdoor navigation.

Researchers at MIT [1], [5], [6] have developed a number of small mobile robots with multiple sensors, including a rangefinder (up to 3 m) and a CCD camera for route following, a whisker (up to 25 cm) for local obstacle detection, and proximity sensors (up to 5 cm) for foot placement. When properly "wired up" by the subsumption architecture, these sensors enable capable walking behaviors with minimal or no computation. Although performance on rugged, natural terrain remains to be demonstrated, to achieve locomotion with little computation is a remarkable accomplishment. However, observation of implemented behaviors reveals frequent contact with the terrain, not only at the feet, but at the legs and body as well. The robots appear to stumble and bump into obstacles. This leads us to believe that the behaviors are power inefficient, because each terrain contact transfers energy. Further, the contact accelerates wear and tear on the vehicle, thus shortening its life.

Fua *et al.* [11] have developed a trinocular stereo system for the French Mars Rover project [4]. Their approach begins by rectifying the images to obtain parallel epipolar lines. Then the approach correlates the rectified images, comparing for each point of image 1 the grey levels in square windows along its corresponding epipolar line in image 2. To check the validity of the correlation, the algorithm applies the same process from image 2 to image 1, validating the result only if the two disparity values are similar. The approach has produced encouraging initial results using three cameras in an L-shaped configuration with 40 cm baselines, with simulated terrain under different outdoor lighting conditions.

In the unmanned ground vehicle domain, the eight-wheeled Autonomous Land Vehicle (ALV) project [26] represents a fpioneering effort. Daily *et al.* [7] present an operational perception system for cross-country ALV traversal that transforms range data from a laser rangefinder into a Cartesian elevation map. A reflexive planning system uses this map to detect and avoid obstacles on a hillside containing steep slopes, rock outcrops, vegetation, and ravines.

The Navlab [25] projects have produced a series of integrated systems with vision, planning, and control of four-wheeled vehicles. Researchers have developed and demonstrated stereo vision, range data processing, object recognition, and neural network methods in the systems, enabling autonomous driving for as long as 21 mi at speeds up to 55 mph, cross-country navigation for over an hour continuously, parallel parking, and suburban navigation missions.

A number of noteworthy outdoor mobile robot systems fall outside the planetary rover and unmanned ground vehicle domains. Some of these systems achieve a significant level of autonomy. For example, the VaMoRs driving system [8] operates on roadways with lane markings, which is an environment relatively rich in structure compared to the surface of the Moon or Mars.

Other systems have the potential for significant autonomy, but have not yet realized it. For example, Iagolnitzer *et al.* [16] describe a system based on a laser rangefinder that constructs a local terrain map as a "horizontal bitmap" in a fixed reference frame. They demonstrate results in modeling a log pile, and discuss plans for testing the system on a tracked robot.

Still other systems have the ingredients for autonomous operation, but no requirement for autonomy. One example is the Adaptive Suspension Vechile project [24], which utilized a laser rangefinder, but more as an operator aid, not for autonomous robot control. Another example is the U.S. Department of Energy operating a wide variety of outdoor vehicles in hostile nuclear environments. Many of these vehicles capture rich video and other data, but for teleoperation, not for autonomous robot control [10].

In summary, the literature reports a spectrum of mapping approaches using a variety of different sensors, control regimes, and algorithms for different environments. We find no mapping system suitably dense and accurate for autonomous and power-efficient locomotion through terrain as rough and rugged as that found on Mars (cf. Section I).

III. AMBLER

The Ambler is a walking robot (Fig. 2) 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], [23].

The Ambler has six legs, arranged in two stacks on central shafts. Each leg consists of a rotational link and an extensional link that move in the horizontal plane, and an orthogonal vertical link. A six-axis force/torque sensor mounted on the base of each vertical link measures the forces acting on the feet. An arched body connects the shafts two central shafts and supports four enclosures housing electronics and computing. The height ranges from 4.1 to 6.0 m, and the width varies between 4.5 and 7.1 m. The mass of the mechanism and all equipment (including power generation and storage) is about 2500 kg.

The Ambler integrated walking system (Fig. 3) consists of a number of distributed modules, each with a specific functionality: the Task Controller [22], which coordinates the distributed robot system; the Real-Time Controller, which implements motion control; the perception modules, which acquire and store images, and construct terrain elevation maps; the planning modules, which plan footfalls, leg trajectories, and gaits; and the graphical user interface. The distributed processes communicate by passing messages through the Task Controller.

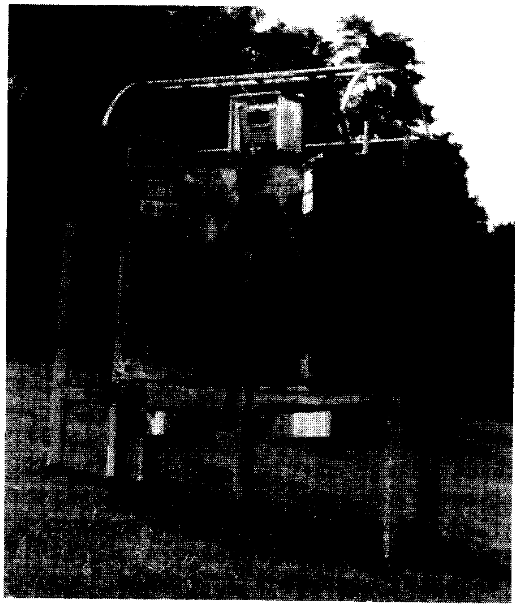


Fig. 2. The Ambler.

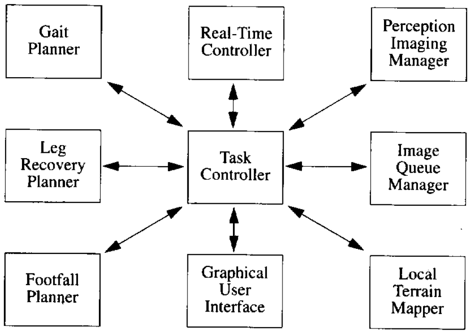


Fig. 3. Modules in the Ambler integrated walking system.

IV. Sensing and Calibration

The primary sensor is a scanning laser rangefinder, which measures both reflectance and range (Fig. 4). We use a laser scanner 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 II) 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 256 x 256 pixel images at a rate of 2 Hz. The sensor is an amplitude-modulated 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 degrees in azimuth and 60 degrees in elevation. The scanner, mounted on top of the Ambler, looks down directly in front of the robot. Readers interested in further information on the sensor will find it elsewhere [13].

Fig. 4. Range (left) and reflectance (right) images. The scene consists of a sand base with meter-tall boulders (foreground) and a wooden ramp.

We have developed an automatic calibration procedure that identifies the transformation between the Ambler-centered reference frame and the sensor-centered frame [17]. 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.

The overall problem is to identify the rigid transformation relating a vehicle-centered reference frame to a sensor-centered reference frame. The origin of the scanner frame S is attached to the scanner and lies somewhere near it. The origin of the body frame B is attached to the walking robot.

We attach a number T of targets to the legs. Then, we move the legs to a number L of different stations. At each, we identify the position r_B of each target in the body frame (by reading joint positions and using known kinematics), and we identify the position r_S of each target in the scanner frame (by image analysis). After acquiring a sufficient number of pairs of measurements, we seek the rotation R and translation t that refer a vector in S to B:

$$r_{Bi} = Rr_{Si} + t, \text{ for } 1 \leq i \leq LT$$

where t is the translation vector relating the two origins, and R as a 3×3 rotation matrix. In practice, it is unlikely that R and t exist that satisfy the above equation, because measurements are not exact and may be contaminated by noise. Instead, we seek R and t that best satisfy the equation in the least-squares sense: Find R and t minimizing the sum of squares of errors

$$E = \sum_{i=1}^{LT} ||e_i||^2$$

where the error of the ith pair of measurements is $e_i = r_{Bi} - Rr_{Si} - t$.

This problem is closely related to a number of other problems that arise in photogrammetry and computer vision. Given the pairs of measurements, the problem is equivalent to the absolute orientation problem in photogrammetry, and to the exterior orientation part of the camera calibration problem in computer vision.

For solution, we implement the technique in [9] (related to [15]), which is an exact closed-form solution that uses

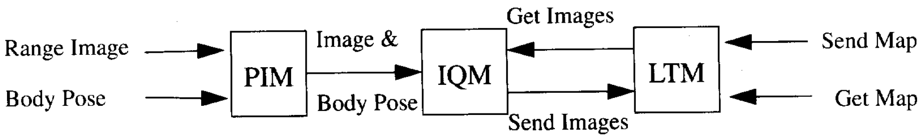


Fig. 5. Mapping system modules.

unit quaternions to represent the rotation. The solution for the desired quaternion is the eigenvector associated with the smallest eigenvalue of a symmetric matrix whose elements are combinations of sums of products of corresponding coordinates from r_S and r_B.

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.

V. 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 (Fig. 5). 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 I. This configuration is modular, extends easily to more and different sensors, decouples imaging and mapping, and manages large quantities of data.

A. 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 body move completes, the Perception Imaging Manager (PIM) receives a command to initiate image acquisition. This is consistent with stop/start walking, but could function equally well for continuous walking.

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 frame (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.

B. 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 nonnatural 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 (Fig. 6). Readers interested in the detailed processing steps will find them elsewhere [13], [14]. 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.

The image queue facilitates access to ordered sequences of images. Sequences are needed because a single forward-looking sensor cannot possibly image obstacles either below or behind the vehicle. A single image does not, in general, contain enough information to accomplish tasks such as planning the trajectory of a recovering leg.

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 (256 K per pair, plus variable-size auxiliary structures such as edge images), and many images may be required (with typical queries for 1 m^2 areas, we observe experimentally that no more than five images are required for a straight-line trajectory on flat terrain, and no more than 25 images 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. Further, this has no effect on map accuracy, because the IQM continues to supply all imagery relevant to the given region of interest.

C. 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.

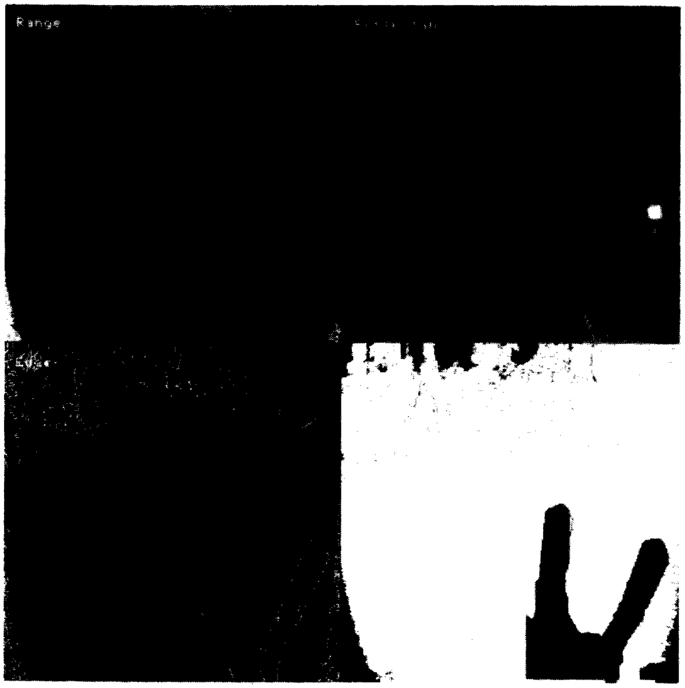


Fig. 6. Preprocessed range image. Range image truncated to 8 bits (upper left). Reflectance image truncated to 8 bits (upper right). Edges detected by Canny operator (lower left). Binary image with white pixels indicating valid range data (lower right).

External modules request elevation maps, specifying (1) a polygonal region, (2) a resolution, and (3) 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:

1) *Unknown:* for those cells outside the polygon or outside the rangefinder field of view or with no valid image pixels.
2) *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.
3) *Known:* for all other cells. In this case, the LTM computes and stores the terrain elevation with the Locus Method [18], an efficient algorithm for transforming and interpolating range data from the sensor frame into Cartesian coordinates.

Finally, the LTM sends the map to the requesting module. Fig. 7 illustrates an elevation map of the scene shown in Fig. 4. This map contains many points labelled *Known*, and few points labelled *Occluded or Unknown.*

1) Elevation Map Construction: The rangefinder measures the coordinates of scene points in a spherical polar reference frame, in which ρ is the measured range, and ϕ and θ are the vertical and horizontal scanning angles of the beam direction corresponding to row and column position in the image. The Cartesian coordinates of a point measured in spherical polar coordinates are

$$x = \rho \sin\theta, y = \rho\cos\phi\cos\theta, z = \rho\sin\phi\cos\theta$$

Applying these relations to the measurements in a range image yields an elevation map. However, this map is non-uniform in Cartesian space, because the coordinate transformation is nonlinear. Further, the map grows less dense and less accurate with increasing distance from the sensor.

One could circumvent the former difficulty by using a map structure that is not a regularly spaced grid, such as a Delaunay triangulation. However, this is not practical because of the complex algorithms required to access data points and their neighborhoods. Another approach is to interpolate between data points to build a dense elevation map on a grid, either by approximating the surface between data points (e.g., as a bicubic surface), or by globally fitting a surface under some smoothness assumptions (e.g., regularization). However, both of these approaches have significant limitations: they make assumptions on the local shape of the terrain that may not be valid in the case of rough terrain; and they depend on the resolution and position of the grid (i.e., they cannot compute an estimate of the elevation at an (x, y) position that is not a grid point without resampling the grid).

We have developed an alternative, the locus algorithm, that uses a model of the sensor to interpolate at arbitrary resolution without making any assumptions on the terrain shape other than the continuity of the surface. Historically, the locus algorithm was conceived by Kweon and Kanade [18].

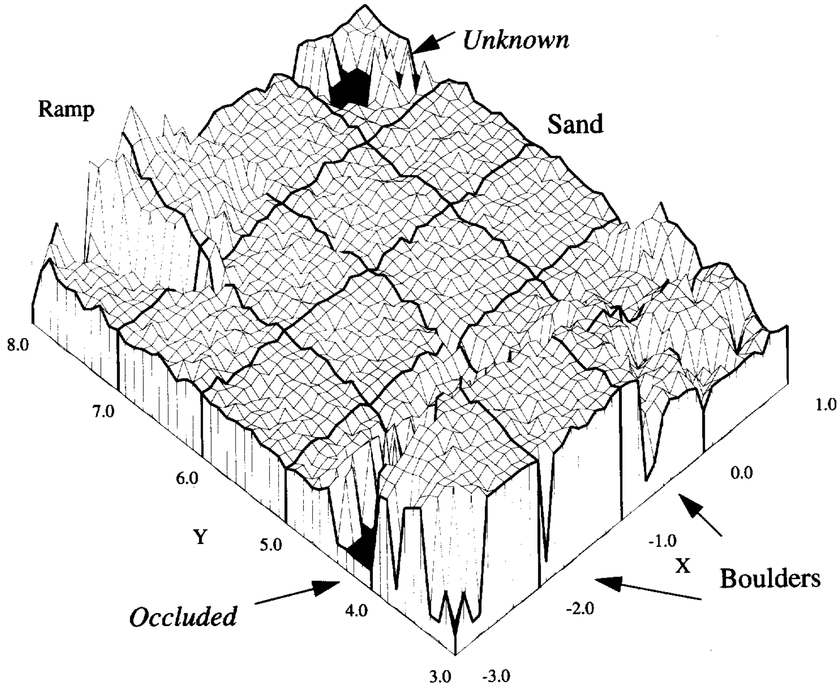


Fig. 7. Elevation map computed from range images of the scene in Fig. 4. The units are in meters.

Our contribution is one of correct implementation, analysis, and optimization. For completeness, in the remainder of this section we describe the locus algorithm, and an extension for detecting range shadows.

2) Locus Algorithm: The problem of finding the elevation z of a point (x, y) is equivalent to computing the intersection of the surface observed by the sensor with the vertical line passing through (x, y). The basic idea of the locus algorithm is to convert the latter formulation into a problem in image space, specifically, spherical-polar space rather than row-column space. A vertical line (we present the case of the vertical line to simplify exposition) is a locus (curve) in image space, whose equation as a function of ϕ is derived by inverting the spherical/Cartesian coordinate transformation, assuming x and y constant:

$$\rho = \rho_l(\phi) = \sqrt{\frac{y^2}{\cos\phi} + x^2}, \quad \theta = \theta_l(\phi) = \text{atan}\frac{x\cos\phi}{y}.$$

Similarly, the range image can be viewed as a surface $\rho = I(\phi, \theta)$ in ϕ-θ space. The problem then is to find the intersection, if it exists, between a curve parameterized by ϕ and a discrete surface. Since the surface is known only from a sample of data, the intersection cannot be computed analytically. Instead, we must search along the curve for the intersection point. Let $\Theta_l(\phi)$ be the image column closest to $\theta_l(\phi)$, and let $\Delta(\phi_j) = \rho_1(\phi_j) - I(\phi_j, \Theta_l(\phi_j))$. The search proceeds in two stages.

1) Locate the two scanlines of the range image, ϕ_1 and ϕ_2, between which the intersection must be located, i.e., such that sgn $\Delta(\phi_1)$ differs from sgn $\Delta(\phi_2)$.
2) Apply a binary search between ϕ_1 and ϕ_2. The search stops when $|\phi_n - \phi_{n+1}| < \varepsilon$ (i.e., the resolution of the elevation is controlled by the parameter ε).

Since there are no pixels between ϕ_1 and ϕ_2, we perform Lagrangian interpolation for $\phi_1 < \phi < \phi_2$, using as control points the four pixels that surround the intersection point. The result is a value ϕ that is mapped to ρ and θ, and then mapped to an elevation value. Repeating this for vertical lines at every desired (x, y) point yields a dense elevation map of the desired resolution, as required.

3) Range Shadows: Objects in the environment may cast range shadows (cause occlusions). It is important to identify the occluded regions, because if the mapping system applies the locus algorithm there directly, then the surface would be smoothly interpolated, possibly incorrectly. In turn, this could lead the rover to plan a path through that region, expecting it to be traversable when in fact it is unknown.

We incorporate the detection of shadow regions into the locus algorithm, again working in image space. We observe that a range shadow corresponds to an occluding edge in the image. An (x, y) location in the map is in a shadow area if its locus intersects the image at a pixel that lies on such an edge. We implement this idea by first detecting edges in the range image by using a variation of the GNC algorithm [3]. Then, when we apply the locus algorithm and observe that the locus of a given location intersects the image at an edge pixel, we mark that location as lying in a range shadow.

4) Combining Elevation Maps: We have so far addressed the problem of building a representation of the environment from sensor data collected at one fixed location. However, over the course of a mission, the rover must deal with a stream of images. Processing multiple views yields at least two benefits. First, by identifying the transformation between viewing positions the perception system can independently estimate the vehicle displacement. Second, merging maps into a composite can (1) increase the resolution of the parts of individual elevation maps originally measured at a distance from the vehicle, and (2) add information about previously occluded areas.

Kweon and Kanade [18] developed a two-stage approach, using feature matching and ionic matching together. The result is an estimate T of the transformation between the two views. Once their matching algorithms identify the transformation corresponding to the displacement between two viewing positions, they apply it pairwise to the sequence of images, producing a single, composite elevation map as follows. They simply add non-overlapping points to the composite map. For overlapping points, we take the maximum likelihood estimate

$$z = \frac{\sigma_2^2 z_1 + \sigma_1^2 z_2}{\sigma_1^2 + \sigma_2^2}$$

where σ_1 and σ_2 are the standard deviations of the uncertainty distributions on the two elevation estimates.

This method is effective, but too slow for practical, real time application while walking. To achieve real time performance, we developed another map merging approach that uses the dead reckoned[1] positions of viewpoints, rather than matching, to estimate the transformation between them. This method uses the most recent image information first to compute elevation, searching back through the queue of images until all elevation values have been computed.

One benefit of combining maps with the most recent images first is the capability of handling moving obstacles (which are expected to be rare in planetary environments). The mapping system will effectively forget an object appearing in older images and not appearing in new images.

Another virtue of the "most recent first" policy is that, when building maps in the current body reference frame (the most common selection for the Ambler walking system), the map merging method does not suffer from the well-known accumulation of error in dead-reckoned estimates. More exactly, it does not suffer from error accumulation over the length of the mission, it suffers only to the (significantly smaller) extent that error accumulates in the few backward chaining steps. A deluxe system for combining elevation maps would periodically perform map matching and use the result to update dead reckoned location estimates. This was not implemented here.

D. System Operation

Fig. 8 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 the 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 Fig. 8, the PIM is free to initiate the image acquisition and storage process.

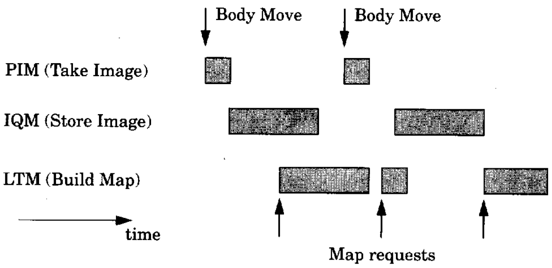


Fig. 8. Concurrent operation of mapping system.

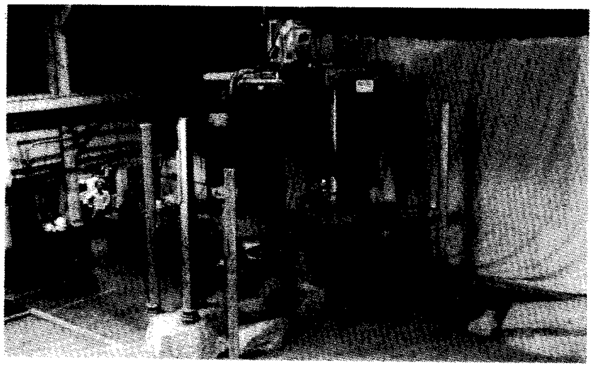


Fig. 9. Indoor obstacle course. Ambler traversing indoor obstacle course over sandy terrain with meter-tall boulders (under legs and body), ditches (the center leg on the far stack is standing in one), and ramp (lower right).

VI. Results

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

A. Maps Constructed in Field Trials

We conducted 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 environments 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-degree wooden ramp, and various other objects (Fig. 9). The courses typically include rolling, sandy terrain with several boulders 1 m tall, ten or so boulders 0.5 m tall, a ditch, and

[1] Dead reckoning integrates elemental motions of the vehicle (here, changes in leg joint angles) over time to estimate the current position and orientation of the vehicle in a fixed, external reference frame. Our approach appears in [21].

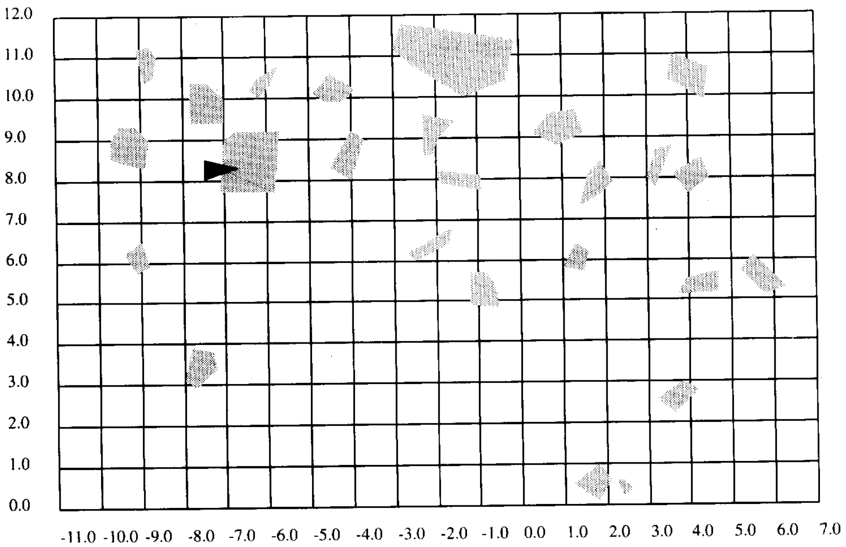


Fig. 10. Top view of figure-eight circuit. The thin line indicates the course followed by the center of the Ambler's body for one complete circuit. Boulders are drawn grey. The black triangle indicates a particular body pose (randomly selected). The units are in meters. Other trials typically result in different courses.

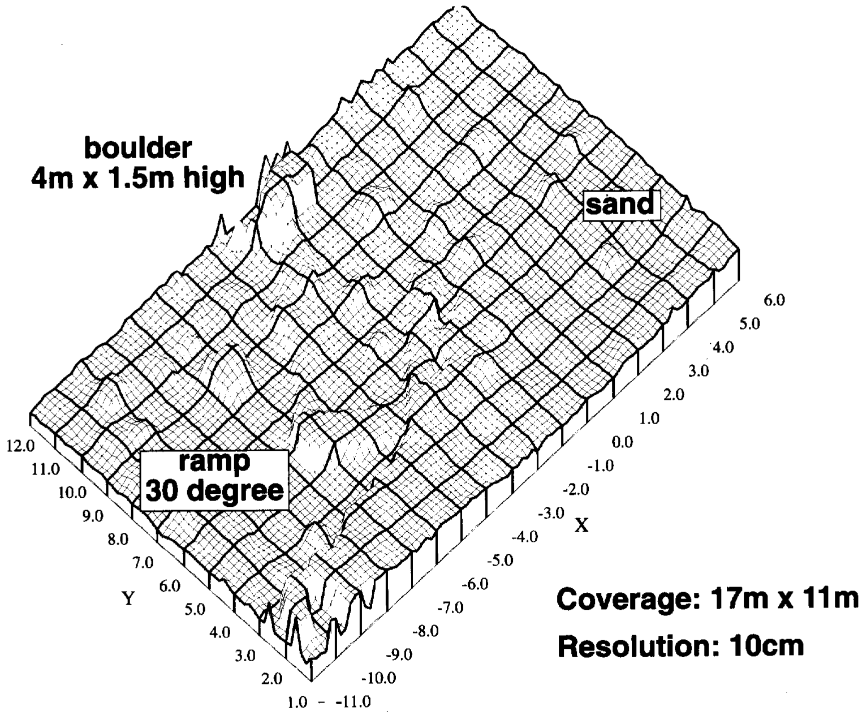


Fig. 11. Map constructed of indoor obstacle course. The mapping system created this 11 × 17 m map from 168 range images. The resolution is 20 cm, and the elevation range is 3 m.

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 m following a figure-eight pattern, each circuit of which covers about 35 m and 550 degrees of turn (Fig. 10). From the images acquired during this trial, the mapping system constructed a composite map of the environment (Fig. 11). Comparing Fig. 10 to Fig. 11, the constructed composite map clearly captures the key environmental features.

A ridge appears in the map of Fig. 11, running vertically down the center. This is an artificial feature caused directly by misregistration of maps and indirectly by inaccuracies in dead reckoned positions. The existence of such spurious, hallucinated ridges surved to artificially restrict the apparent

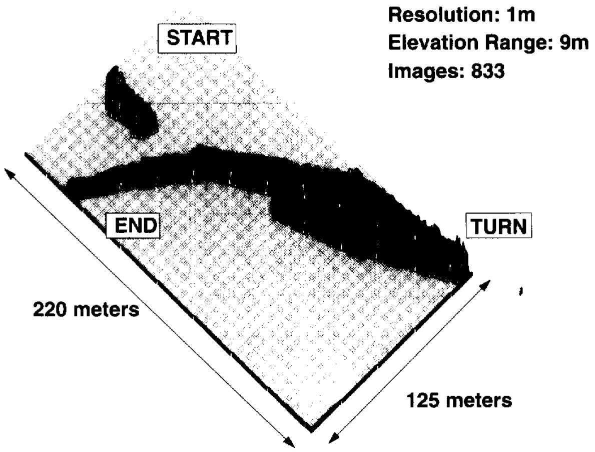


Fig. 12. Map constructed of outdoor field. The mapping system created this 220 x 125 m map from 833 images. The resolution is 1 m, and the elevation range is 9 m. During the trial the system could not store some images because a disk was full, causing the gap in the middle of the map.

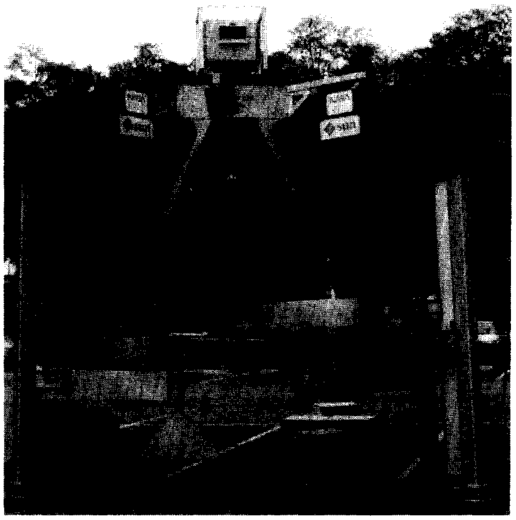


Fig. 13. Ambler walking 100 m on an outdoor obstacle course.

space of reachable foot placements [14]. After fixing the dead reckoning system problem that caused the discontinuity in estimated positions, we never observed another ridge.

For one set of outdoor trials, the Ambler operated on rolling, grassy terrain (Fig. 2). The site is a field 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 side-slopes, soft ground, 30 °F temperature variations, lighting conditions varying from bright sunshine to partly cloudy to dark. In the longest of these trials, the Ambler took 1219 steps, travelling 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 isoelevation contour for 250 m, executing a point turn of roughly π radians, and following an isoelevation contour back to the starting region.

From the images acquired during this trial, the mapping system produced a comprehensive site map (Fig. 12). The resolution of the computed map is 10 cm; the lower resolution of the map shown is an unfortunate consequence of limited printer capability. Comparing the computed map to the site topography, we find excellent agreement.

For another set of outdoor trials, the Ambler operated in a parking lost strewn with wooden obstacles such as boxes, pyramids, and ramps (Fig. 13). In one of these trials, the Ambler took 100 steps along a gently curving arc, travelling about 25 m over a variety of obstacles. In another of these trials, the Ambler walked in the parking lot at night, without lights. The laser rangefinder does not require ambient illumination, unlike ordinary cameras. In fact, we observed the range images to be noisier during the day, because the signal-to-noise ratio is higher without ambient illumination.

B. Performance Statistics

Table I records mapping system statistics gathered from several representative walking trials. The distance term represents the distance travelled 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.

TABLE I
Mapping System Statistics In Four Separate Trials

Trial	Distance (m)	Images acquired	Subimages transferred	Maps built	Map point
Outdoor	46	151	8974	626	375 003
Indoor	68	185	10 015	640	267 104
Indoor	107	397	20 286	1306	597 078
Outdoor	527	1219	101 129	4716	2 578 102

The distances traversed indicate the effectiveness of the system in accurately capturing the terrain geometry. The large volume of data processed, both images and maps, indicates the robustness of the system. The volume of data processed is widely used as an indicator of reliability for systems such as automated teller machines and airline reservation systems. In this case, the system is robust in the sense of continuing to operate effectively despite nonsensical map requests, sensor malfunctions, operating system errors related to memory management, and other exceptional conditions.

Over these trials, we find that taking a single step requires between 20 and 30 s of mapping system processing, which involves examining between 5 and 28 range subimages to compute elevation values for some 1700 points, on average. Further analyzing the timing, we find that for Sparc2 workstations, image acquisition takes about 1 s, image processing and storage requires about 10 s, and construction of elevation maps takes about 5 ms 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

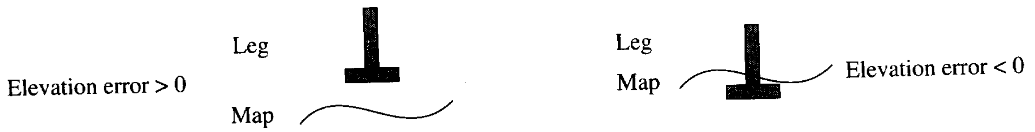


Fig. 14. Elevation error.

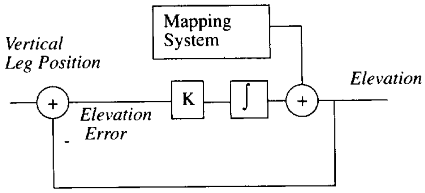


Fig. 15. Closed loop control of elevation values.

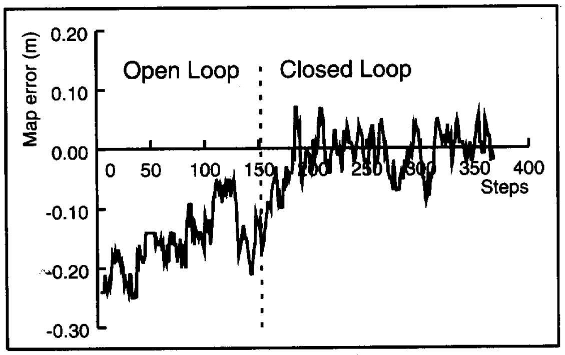


Fig. 16. Elevation error during open-loop and closed-loop control.

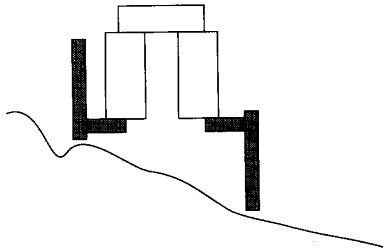


Fig. 17. Slope traversal. The leg shown on the left is near the lower limit of vertical travel. The leg shown on the right is near the upper limit of vertical travel.

the map accuracy varies from step to step and from trial to trial, but is almost always within the scanner precision of 10–15 cm. This accuracy is sufficient for the Ambler, whose feet are 30 cm in diameter. Another measure of map accuracy is the sensor calibration residual, which ranges from 2–7 cm over different trials (cf. Section IV). The most straightforward measure of map accuracy would be the mean distance between the computed map and the ground truth. However, we were unable to devise a reliable and economical method to measure dense ground truth over even a small patch of terrain, and so we cannot report this statistic.

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.

VII. 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 [13], 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 computed elevations to change.

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 error variations in the computed maps.

To characterize this elevation error 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 error is negative, the foot is below the sensed terrain (Fig. 14).

To adjust the elevation values, we implement a control loop (Fig. 15) that uses the elevation error as an error signal to increase the accuracy of the elevation maps. The control law

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

where i indicates the step number, includes a proportional term, with gain k typically set to 0.1, and an integral term, to reduce the steady-state error. To adjust the elevation values,

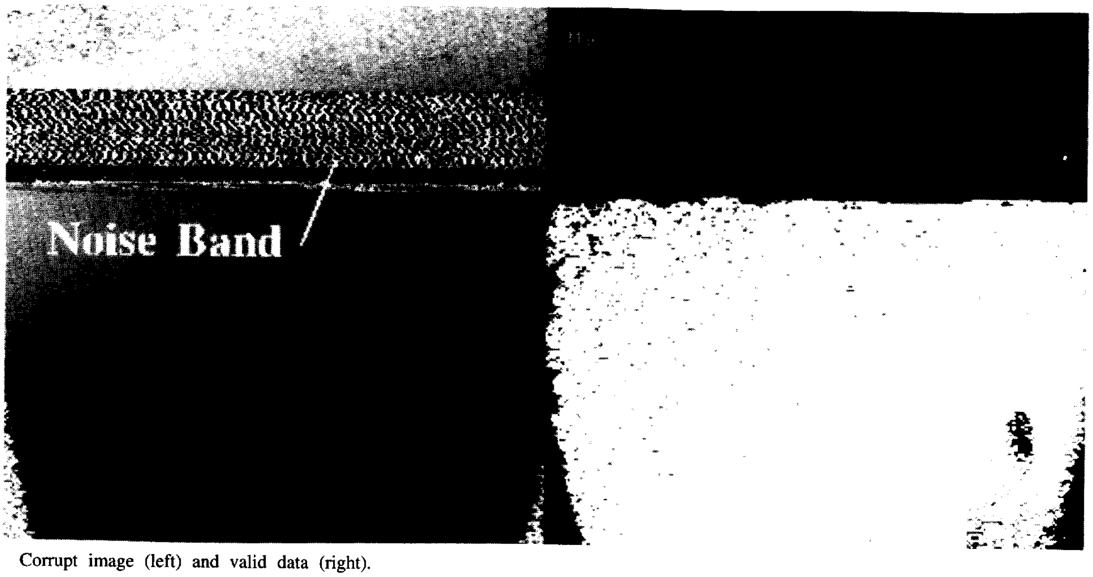


Fig. 18. Corrupt image (left) and valid data (right).

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 (cf. Section IV).

Fig. 16 plots the elevation error for 371 steps taken over an 8-h 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 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. In other words, the controller has eliminated the measurements bias, but not the measurement noise.

There are two consequences of the closed loop operation, both demonstrated during the 500 m trial. First, mobility is greater on slopes. In Fig. 17, the leg on the left stack is near a lower vertical limit (above which the leg cannot travel), and the leg on the right stack is near an upper vertical limit (below which the leg cannot travel). If the elevation error is too negative, the robot incorrectly "thinks" that it cannot raise the leg on the left stack to clear the terrain. If the elevation error is too positive, the robot incorrectly "thinks" that it cannot lower the leg on the right stack to contact the terrain. In either case, the robot finds it impossible to advance. 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. Under closed loop control, the leg stays closer to the ground than in the open loop case, thus saving both time and power.

VIII. 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.

We consider the robustness of the system to be one its key contributions. To illustrate, consider an incident from the outdoor walking trial. Fig. 18 illustrates an image acquired outdoors that is corrupted by a horizontal band of high-contrast salt-and-pepper noise. In three prior years of working with the rangefinder, we never observed such an effect. We never anticipated that kind of problem, and never explicitly modeled or solved a related problem. Indeed, the researchers conducting the walking tests did not observe the problem occur (they were playing Frisbee as the robot walked autonomously and discovered it only afterwards) and did nothing to intervene. The aggressive image preprocessing routines were not crippled by this sensor malfunction. They detected disconnected pixels and invalidated the offending data, thus preventing the introduction of spurious data into the maps. This is the kind of robust behavior required by planetary rovers.

Another key issue addressed by the mapping system is calibration. Recently, researchers have questioned the need for calibration, and techniques to avoid it have gained favor. In the case of the Ambler, it is likely that we too could survive without calibration; because the machine is big, rugged, and heavy, many objects that are obstacles before the Ambler steps on them are planar afterwards. But for missions to distant, rugged regions like planetary surfaces, Antarctica, and the ocean floor, survival is not enough. Power-efficient locomotion is essential. With a calibrated sensor, the Ambler selects where to step, and thus prevents spending significant fractions of the total power budget on stumbling rather than productive advance. This ability to select footholds is central to the fundamental advantages of high mobility and power efficiency that walkers enjoy over rolling and crawling machines. Achieving those footholds requires calibration.

Our future work will continue to address the theme of robust three-dimensional sensing. We will develop stereo 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 hardware faults such as memory parity errors and operating system errors.

Acknowledgment

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.

References

[1] C. Angle and R. Brooks, "Small planetary rovers," in *Proc. IEEE Int. Workshop Intelligent Robots and Syst.*, Tsuchiura, Japan, July 1990, pp. 383–388.

[2] J. Bares and W. Whittaker, "Configuration of autonomous walkers for extreme terrain," *Int. J. Robotics Res.*, to appear, 1993.

[3] A. Blake and A. Zisserman, *Visual Reconstruction.* Cambridge, MA: MIT Press, 1987.

[4] L. Boissier, B. Hotz, C. Proy, O. Faugeras, and P. Fua, "Autonomous planetary rover: On-board perception system concept and stereovision by correlation approach," in *Proc. IEEE Int. Conf. Robotics Automat.*, Nice, France, May 1992, pp. 181–186.

[5] R. Brooks and A. Flynn, "Fast, cheap, and out of control: A robot invasion of the solar system," *J. Brit. Interplanetary Soc.*, vol. 42, no. 10, pp. 478–485, 1989.

[6] R. Brooks, P. Maes, M. Mataric, and G. More, "Lunar base construction robots," in *Proc. IEEE Int. Workshop Intell. Robots Syst.*, Tsuchiura, Japan, July 1990, pp. 389–392.

[7] M. Daily, J. Harris, and K. Reiser, "An operational perception system for cross-country navigation," in *Proc. Image Understanding Workshop*, Cambridge, 1988.

[8] E. Dickmanns and B. Mysliwetz, "Recursive 3-D road and relative ego-state recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 14, no. 2, 1992.

[9] O. Faugeras and M. Hebert, "The representation, recognition and locating of 3D shapes from range data," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 27–52, 1986.

[10] R. Fogle, "The use of teleoperators in hostile environment applications," in *Proc. Int. Conf. Robotics Automat.*, Nice, France, April 1992, pp. 61–66.

[11] P. Fua, "A parallel stereo algorithm that produces dense depth maps and preserves image features," Tech. Rep. 1369, INRIA, Sophia-Antipolis, France, January 1991.

[12] E. Gat, "Robust low-computation sensor-driven control for task-directed navigation," in *Proc. IEEE Int. Conf. Robotics Automat.*, Sacramento, CA, May 1991, pp. 2484–2489.

[13] M. Hebert and E. Drotkov, "3-D measurements from imaging laser radars: How good are they?" *Int. J. Image and Vision Comput.*, vol. 10, no. 3, pp. 170–178, 1992.

[14] R. Hoffman and E. Krotkov, "Perception of rugged terrain for a walking robot: True confessions and new directions," in *Proc. IEEE Int. Workshop on Intell. Robots and Syst.*, Osaka, Japan, November 1991, pp. 1505–1511.

[15] B. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Optical Soc. Am.*, vol. A4 pp. 629–642, 1987.

[16] M. Iagolnitzer, F. Richard, J. Samson, and P. Tournassoud, "Locomotion of an all-terrain mobile robot," in *Proc. Int. Conf. Robotics Automat.*, Nice, France, April 1992, pp. 104–109.

[17] E. Krothov, "Laser rangefinder calibration for a walking robot," in *Proc. IEEE Int. Conf. Robotics Automat.*, Sacramento, CA, April 1991, pp. 2568–2573.

[18] I. S. Kweon and T. Kanade, "High-resolution terrain map from multiple sensor data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 278–292, 1992.

[19] Mars Study Team, "A preliminary study of Mars rover/sample return missions," Solar System Exploration Division, NASA Headquarters, January 1987.

[20] L. Matthies, "Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation," *Intl. J. Comput. Vision.*, vol. 8, no. 1, pp. 71–91, 1992.

[21] G. Roston and E. Krotkov, "Dead reckoning navigation for a six-legged walking robot," in *Proc. IEEE Int. Workshop on Intell. Robots Syst.*, Raleigh, NC, July 1992, pp. 607–612.

[22] R. Simmons, L.-J. Lin, and C. Fedor, "Autonomous task control for mobile robots," in *Proc. IEEE Int. Symp. Intell. Contr.*, Philadelphia, PA, September 1990, pp. 663–668.

[23] 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," *J. Applied Intell.*, vol. 2, pp. 163–180, Spring 1992.

[24] S. Song and K. Waldron, *Machines that Walk: The Adaptive Suspension Vehicle.* Cambridge, MA: MIT Press, 1989.

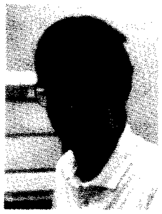
[25] C. Thorpe, Ed. *Vision and Navigation: The Carnegie Mellon Navlab.* New York: Kluwer, 1990.

[26] M. A. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra, "VITS—A vision system for autonomous land vehicle navigation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 3, 1988.

[27] B. Wilcox and D. Gennery, "A Mars rover for the 1990's," *J. Brit. Interplanetary Soc.*, vol. 40, pp. 484–488, 1987.

[28] B. Wilcox, L. Matthies, D. Gennery, B. Copper, T. Nguyen, T. Litwin, A. Mishkin, and H. Stone, "Robotic vehicles for planetary exploration," in *Proc. IEEE Int. Conf. Robotics Automat.*, Nice, France, May 1992, pp. 175–180.

Eric Krotkov (S'85–M'87) received the B.A. degree in 1982 from Haverford College and the Ph.D. degree in 1987 from the University of Pennsylvania, in computer and information science. Since 1988 he has been a Senior Research Scientist at Carnegie Mellon University. Prior to his appointment to Carnegie Mellon University he was a postdoctoral fellow at the Laboratoire d'Automatique et d'Analyse des Systemes in Toulouse, France. He has worked in several areas of machine perception and robotics: active computer vision, three-dimensional vision, mobile robot navigation and perception of material properties. He has authored one book and over fifty papers and technical reports in these areas.



Regis Hoffman received the B.S. degree in Computational Physics from Carnegie Mellon in 1981. He is a Project Supervisor in the Field Robotics Center at Carnegie Mellon University. His research interests focus on mobile robot perception and simulation, and he has authored over 15 papers in these areas. His current work includes perception for agricultural robots and computer modeling of hazardous waste sites.