Presented at the NASA Conference on Space Telerobotics, Pasadena, California, January 31-February 2, 1989.

First Results in Terrain Mapping for a Roving Planetary Explorer

E. Krotkov, C. Caillas, M. Hebert, I. S. Kweon, T. Kanade

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract[1]

To perform planetary exploration without human supervision, a complete autonomous rover must be able to model its environment while exploring its surroundings. We present a new algorithm to construct a geometric terrain representation from a single range image. The form of the representation is an elevation map that includes uncertainty, unknown areas, and local features. By virtue of working in spherical-polar space, the algorithm is independent of the desired map resolution and the orientation of the sensor, unlike other algorithms that work in Cartesian space. We also describe new methods to evaluate regions of the constructed elevation maps to support legged locomotion over rough terrain.

1 Introduction

We are prototyping a legged vehicle called the Ambler (fig. 1) for an exploratory mission on another planet, conceivably Mars, where it is to traverse uncharted areas and collect material samples. Planetary exploration poses significant challenges for rovers: unprecedented levels of autonomy and reliability due to communication delays that limit conventional Earth-based teleoperation; and traversal of rugged, irregular terrain for which existing mechanisms and perception techniques are inadequate.

Papers that describe the background of our work include a comprehensive account of the Ambler configuration [1] and an overview of the integrated research program [4]. The aim of this paper is to describe first results from the Ambler perception system.

The Ambler perception system must build and maintain representations of the terrain and discrete objects—*terrain maps* that are appropriate for a wide variety of tasks, each with different requirements. For example, locomotion and sampling require detailed, local representations, while navigation and mission planning demand broad, global descriptions. In this paper, we do not address the full scope of the Perception system; we focus only on building maps based on the observations of a single sensor, and using those maps to support locomotion.

This paper addresses sensing in section 2, and presents a new technique for constructing elevation maps in section 3. It describes methods for analyzing map geometry for locomotion

[1]This research was sponsored by NASA under Contract NAGW-1175. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agency.

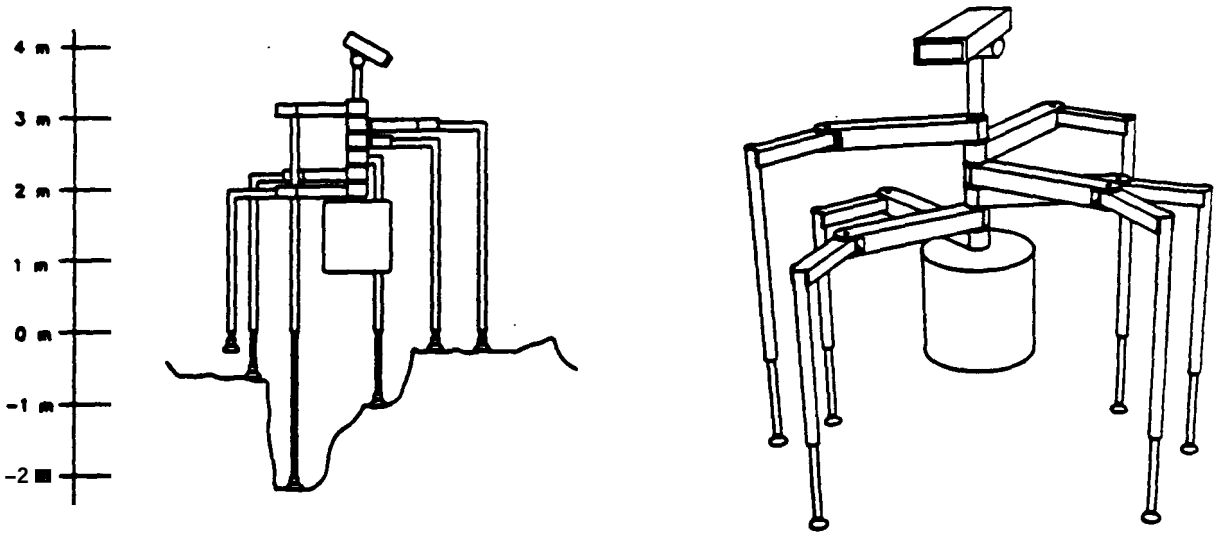

Figure 1: The Ambler drawn at shown scale (left) and another scale (right)

in section 4, and documents experimental methods and results in section 5. It concludes by discussing limitiations and future work.

2 Active Range Sensing

The Ambler Perception system will use multiple sensing modalities, both imaging and non-imaging. Hen we concentrate on active range sensors, which measure the distance to an object in the environment by observing the reflection of a reference *signal* (sonar, laser, radar, etc.) from the object. Active sensors offer *two* chief advantages: they provide range data without the numerous computations required by passive techniques such as stereo vision; and they are largely insensitive to illumination conditions, thus simplifying the image analysis problem?

We use a scanning laser range finder, developed by ERIM, that measures the phase difference between an amplitude-modulated laser beam and its reflection from a point in the scene [7]. We measure the coordinates of the point in a non standard 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 have been derived [3] as

$$x = \rho \sin\theta \quad , \quad y = \rho \cos\phi \cos\theta \quad , \quad z = \rho \sin\phi \cos\theta \quad . \tag{1}$$

[2]This is especially important for images of outdoor scenes in which illumination can be neither controlled nor predicted.

3 Elevation Map Construction

Applying eq. 1 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 non-linear. 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 [5]. 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 which may not be valid in the case of rough terrain; and they depend heavily 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 propose 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.

31 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[3] (fig. 2). A vertical line[4] is a locus (curve) in image space, whose equation as a function of ϕ is derived by inverting eq. 1, assuming x and y constant:

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

Similarly, the range image *can* be viewed as a surface $\rho = I(\phi, \theta)$ in ϕ, θ, ρ 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 $\hat{\theta}_l(\phi)$ be the image column closest to $\theta_l(\phi)$, and let $\Delta(\phi_j) \equiv \rho_l(\phi_j) - I(\phi_j, \hat{\theta}_l(\phi_j))$. The search proceeds in two *stages*. First, we locate the two scanlines of the range image, ϕ_1 and ϕ_2, between which the intersection must be located, i.e., *such* that $\text{sgn}\,\Delta(\phi_1) \neq \text{sgn}\,\Delta(\phi_2)$. Second, we apply a binary search between ϕ_1 and ϕ_2. The search stops when $|\phi_n - \phi_{n+1}| < c$ (i.e., the resolution of the elevation is controlled by the parameter c). Third, since then are no pixels between ϕ_1 and ϕ_2,

[3] Specifically, spherical-polar space rather than row-column space.

[4] We have generalized the locus algorithm from the *case* of a vertical line to the case of a general line in space [3], which allows us to build maps using my reference plane, not just the xy plane. We present *the* case of the vertical line to simplify exposition.

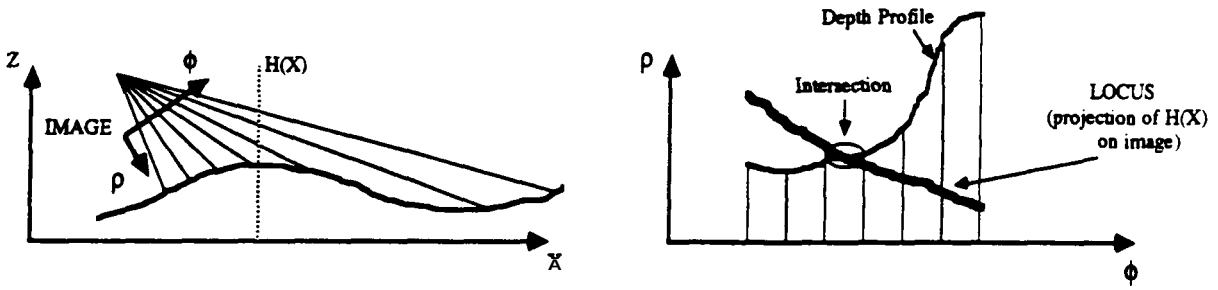

Figure 2 Imaging geometry (left) and one-dimensional locus (right)

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 by eq. 2, and then mapped to an elevation value by eq. 1. Repeating this for vertical lines at every desired (x, y) point yields a dense elevation map of the desired resolution, as required.

3.2 Range Shadows

Objects in the environment may cast range shadows (cause occlusions). It is important to identify the occluded regions, because if we apply the locus algorithm then 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.

One could *detect empty* regions in the elevation map given by eq. 1, without interpolation. This does not work, because the *size* of the shadow regions may be *on* the order of the average distance between data points.[5]

Another approach is to 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. As in fig. 3, 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 the GNC algorithm [2]. 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.

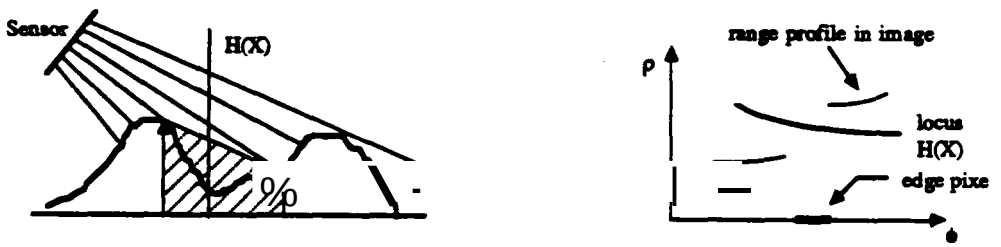

Figure 3 Shadowed area (left) and range discontinuity (right)

[5] This is especially true for distant regions in which the distribution of data points is sparse.

3.3 Uncertainty

We have developed a probabilistic model of the uncertainty on the sensor measurements, according to which the measured range errors are normally distributed with standard deviation proportional to the square of measured range ([3], p. 7). The range measurement uncertainty is oriented along the direction of measurement (fig. 4).

To identify the uncertainty on the elevation value at each grid point (x, y), as part of the locus algorithm we transform the uncertainty on a sensor measurement so that it is oriented along the z axis ([3], pp. 25-27). This conversion is non trivial, since the the range uncertainty is distributed across a region in the elevation map. According to this model, the distribution of elevation errors is approximately normal, with standard deviation proportional to the product of measured range and elevation.

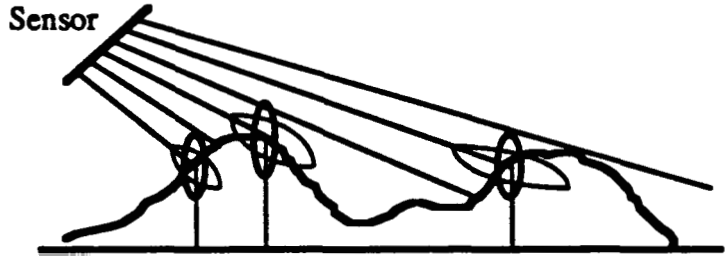

Figure 4: One-dimensional uncertainty distributions on sensor and map

4 Footfall Evaluation

A perceptual task unique to legged locomotion is to evaluate terrain regions as footfall locations (foot placements). This is essential for locomotion over the rugged terrain that could be encountered on the surface of other planets such as Mars. In this section, we describe several methods to evaluate elevation map regions as footfall locations. These methods operate on the geometric structure of the surface described by the elevation map, fur now ignoring important material properties of the soil such as load-bearing strength, compliance, and coefficient of friction. While incomplete, these methods are considerably more sophisticated than others reported in the literature, which require operator interaction [6].

An Ambler foot is modeled by a flat disk 30 cm in diameter. The problem is to find the "best" foot-shaped subregion B in a given region R of a given elevation map.[6] We have developed five solutions, corresponding to different measures of "best" and present them in increasing order of sophistication.

Max-min Find B that minimizes the difference $z_{max} - z_{min}$ of extremal elevations, as illustrated in fig. 5a. There are cases where this method prefers a flat surface punctuated by a single spike rather than an undulating surface (cf. figs. 5a and 5b). This is obviously undesirable.

[6] The region R is computed elsewhere based on the current heading and gait.

Planar fit Find **B** that best fits a plane, subject to the constraint that the plane normal is approximately parallel *to* the leg. This method suffers *the* same deficiency as above.

Support area Find **B** that *minimizes* the depth of penetration d_{opt} into the soil (fig. 5c) required *to* achieve the minimum necessary support area A_{min} (contact area between foot and terrain, or the number of map points within the circumference of the *disk* that *are* above the plane of the foot). This method is superior *to* the previous two *to* the extent that it better accounts for the shape of the terrain. However, there *are cases* that it fails to distinguish, e.g., two sinusoidal surfaces with the same *frequency* but different amplitudes. This method should, but does not, select the surface with smaller amplitude variations.

Free volume Find **B** that minimizes the free (unoccupied) volume between the foot and *soil*, $V \equiv Nz_{max} - \sum_{i=1}^{N} z_i$, as shown in fig. 5b. This method correctly discriminates the two sinusoidal surfaces described above. However, it does not take into account the distribution of "holes" in the surface or the consequences of applying force to (stepping *on)* the surface.

Equilibrium Find **B** that minimizes *V* and *E*, where $E \equiv \sqrt{m_x^2 + m_y^2}$ is the first moment of the mass distribution about the center of the foot, and $m_x = \sum_{i=1}^{N} x_i(z_{max} - z_i)$. The second condition ensures the footfall of greatest "equilibrium" (balance) with respect *to* holes in the surface, as suggested by fig. 5d. The idea is that as the foot contacts *sandy* soil, the sand fills the holes with a minimum of foot penetration into the soil, and as the foot contacts rocky soil, it exerts the minimum lateral forces *on* potentially unstable materials.

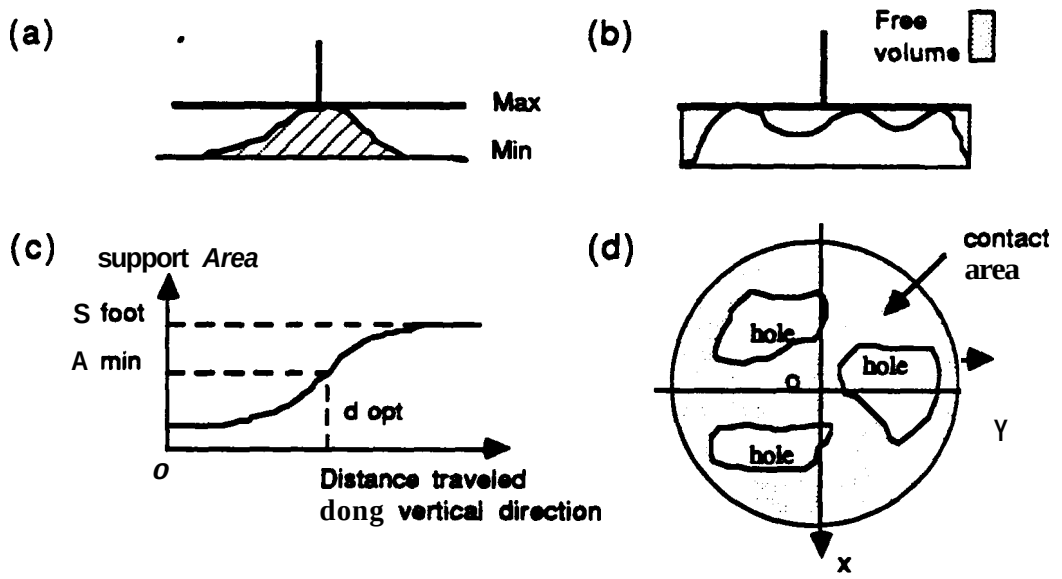

Figure 5: Footfall evaluation criteria

5 Experiments

In this section we summarize our initial experiments and results.

First, we evaluated the locus algorithm on synthesized range images with additive Gaussian noise by comparing its performance to that of Cartesian space interpolation algorithms (cf. section 3). The results show that the locus algorithm is more stable with respect to surface orientation and noise level than the others ([3], p. 25). We conclude that this is due to performing the interpolation in image space instead of first applying eq. 1 to the data points.

Then, we tested the locus algorithm on a variety of real range images. The left half of fig. 6 shows the result of applying it to a range image of uneven terrain found at a construction site. The figure shows the original range image and displays the elevation map as an isoplot surface at 10 cm resolution. The right half of fig. 6 shows an overhead view of a different elevation map, when the grey levels indicate the following: white is shadow, black is unknown, grey is proportional to elevation uncertainty. Note that more *distant* points are more uncertain, as expected.

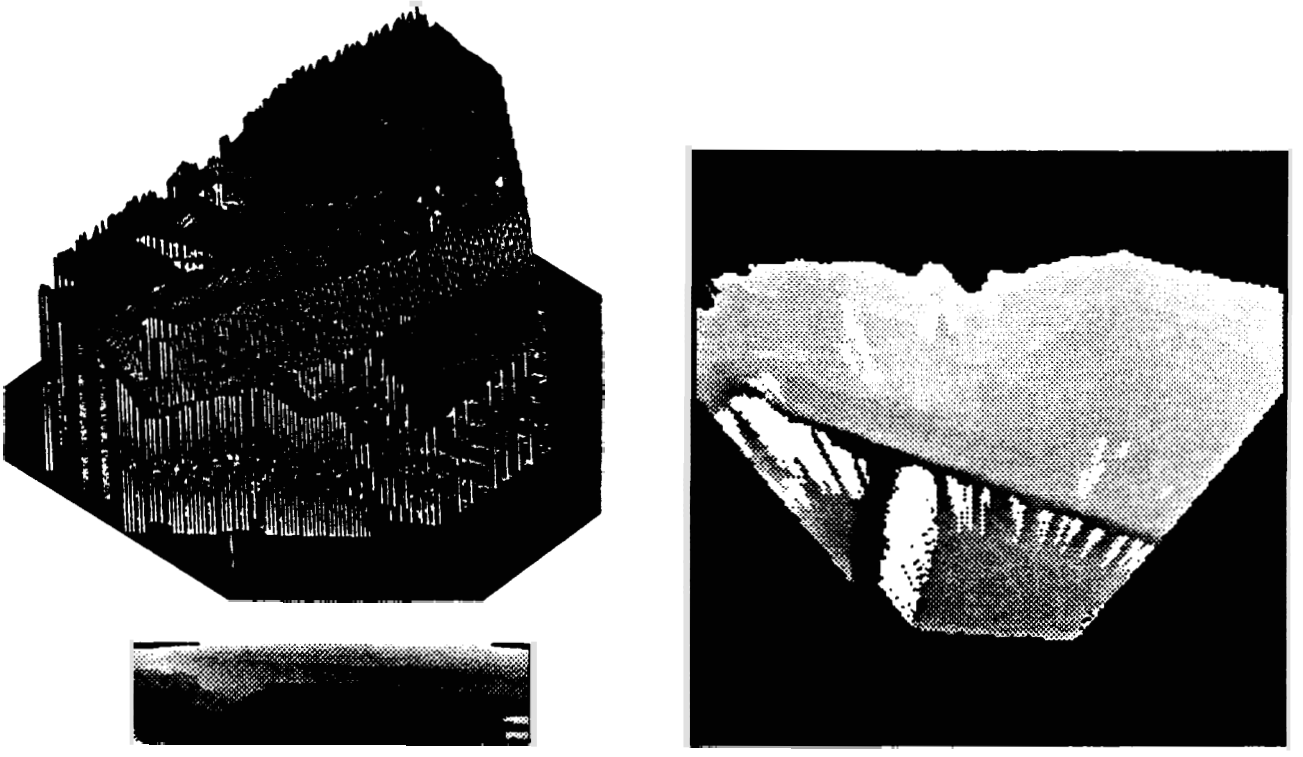

Figure 6 Elevation map (left), shadow regions and uncertainty (right)

Finally, we tested a partially integrated Ambler *system at* an experimental testbed (fig. 7): a single leg with a fully operational controller; the range finder mounted above the leg; and a 25 m^2 "sandbox" of terrain to be traversed. The perception *system* communicates with other modules through queries, which typically are requests for the elevation map at a given resolution within a polygonal region.

The lower left panel of fig. 8 shows the polygonal region referenced by queries for elevation, uncertainty, and footfall location, and the other panels show the perception system's replies

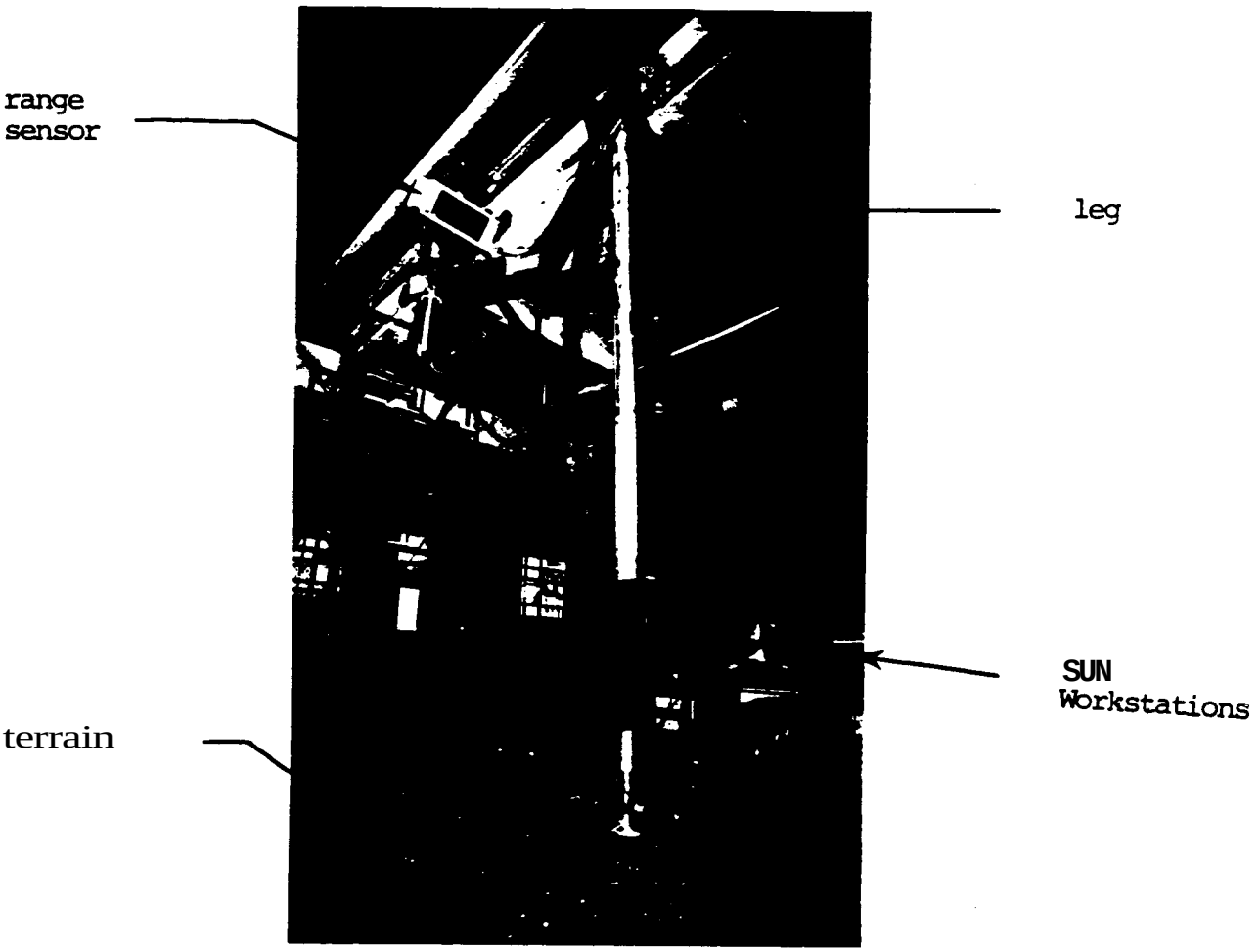

Figure 7: Single leg testbed

computed by the locus and equilibrium algorithms. We evaluate the selected footfall location by servoing the leg there, thus closing the loop between perception and action. Visual inspection of the servoed positions shows the selected locations to be reasonably accurate; quantitative error measurements are not yet available. Dozens of trials on different terrains suggest that the perception algorithms provide reliable and reasonably accurate descriptions of the terrain that suffice for moving the leg and executing footfalls.

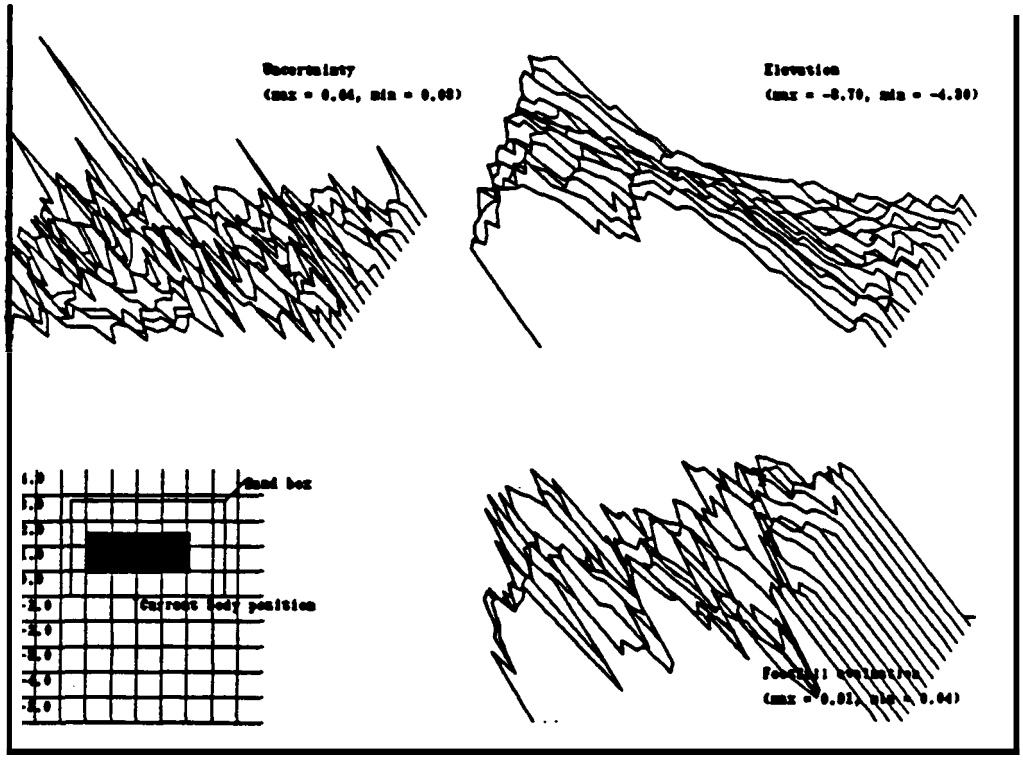

Figure 8: Perception system replies to map and footfall location queries

6 Discussion

In this paper we presented techniques to build maps based *on* the observations of a single range sensor, and to use those maps to *support* locomotion: a new algorithm to build elevation maps *at* arbitrary resolution, including elevation uncertainty and unknown areas; and new methods for geometrically evaluating areas of the constructed elevation map as footfall locations.

Preliminary experiments demonstrate that an integrated system *can* build and use maps to *select* footfall locations. This illustrates the advantages of working in image space rather than in Cartesian space.

While the first results are encouraging, further work is required both in map building and map analysis. For the former, we must complete an *automatic* calibration procedure to more

accurately relate sensor and vehicle coordinate systems. For the latter, we must investigate more sophisticated footfall evaluations that take into account not only the local geometry of the terrain, but also geometric uncertainty and material properties of the soil such as load-bearing strength, compliance, and coefficient of friction. Further, we must better integrate the algorithms into the Ambler system, and make more quantitative assessments of their performance.

The work reported in this paper addresses a small fraction of the problems faced in developing a complete perception *system* for the Ambler. The scope of future research includes two broad categories: navigation and sampling. For the former, we aim to increase map coverage by processing multiple views from multiple sensors, to determine vehicle position by landmark triangulation, and to compute vehicle displacement by matching elevation maps. For the latter, we intend to use surface topography to identify promising sample sites, and to build models of discrete objects both to select particular samples and to guide sample acquisition.

References

[1] J. Bares and W. Whittaker. Configuration of an Autonomous Robot for Mars Exploration. In *Proc. World* Robotics *Conference*, Society of Mechanical Engineers, To appear, May 1989.

[2] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, Massachusetts, 1987.

[3] M. Hebert, T. Kanade, and I. Kweon. *3-D Vision Techniques for Autonomous Vehicles*. Technical Report CMU-RI-TR-88-12, The Robotics Institute, Carnegie Mellon University, 1988.

[4] E. Krotkov, J. Bares, M. Hebert, T. Kanade, T. Mitchell, R Simmons, and W. Whittaker. An Autonomous Rover for Exploring Mars. *IEEE Computer*, To appear, June 1989.

[5] D. J. Orser and M. Roche. The Extraction of Topographic Features in Support of Autonomous Underwater Vehicle Navigation. In *Proc. Fifth International Symposium on Unmanned Untethered Submersible Technology*, Merrimack, New Hampshire, June 1987.

[6] F. Ozguner, S. J. Tsai, and R. B. McGhee. *An* Approach to the Use of Terrain-Preview Information in Rough-Terrain Locomotion by a Hexapod Walking Machine. *International Journal of Robotics Research*, 3(2):134–146, Summer 1984.

[7] D. Zuk, F. Pont, R. Franklin, and V. Larrowe. *A System* for *Autonomous Land Navigation*. Technical Report IR-85-540, Environmental Research Institute of Michigan, Ann Arbor, Michigan, 1985.