

# Stereo Driving and Position Estimation for Autonomous Planetary Rovers \*

E. Krotkov   M. Hebert   M. Buffa   F. Cozman   Luc Robert'  
The Robotics Institute,  
Carnegie Mellon University,  
5000 Forbes Avenue,  
Pittsburgh PA 15213

## Abstract

In this paper we present two new approaches to planetary rover perception. One approach concerns stereo driving without 3-D reconstruction. This approach begins with weakly calibrated stereo images, and evaluates the traversability of terrain using shape indicators such as relative slope and relative elevation. The approach then evaluates candidate paths based on the traversability analysis and generates the best path.

The second approach involves estimating vehicle position by observing the Sun. At a given time, a measurement of the Sun's altitude constrains the observer to lie on a circle on the terrestrial surface called the circle of equal altitude. We determine the position of the observer by intersecting circles of equal altitude identified at different times.

We are validating experimentally both approaches in unstructured, outdoor environments with several wheeled rovers. Future efforts will transfer the developed technology into Lunar Rover demonstration and flight programs.

## 1 Introduction

Planetary exploration, by its nature, requires operation in highly unstructured environments. The lack of structure on the surfaces of Mars or the Moon challenges established machine perception techniques because the irregular terrain found on planetary surfaces does not satisfy standard constraints on shape, e.g., symmetry, or surface properties, e.g., smoothness, nor does it permit the standard tricks of controlled lighting or fixtured objects.

---

\*This research is supported in part by NASA under Grant NAGW-1175 and by ARPA under contracts DACA76-89-C-0014 and DAAE07-90-C-R059. Robert and M. Buffa were supported under post-doctoral fellowships from INRIA, France. F. Cozman is supported under a scholarship from CNPQ, Brazil.

†Current address: INRIA, 2004 Route des Lucioles, 06902 Sophia Antipolis Cedex France.

Over the past five years, we have addressed this challenge by developing and validating laser rangefinder systems for two outdoor rovers:

1. Ambler, a six-legged robot intended to undertake Mars Rover Sample Return class missions. The implemented system enabled the Ambler to successfully walk kilometers over rugged, natural terrain [8].
2. HMMWV, a four-wheeled all-terrain vehicle intended for cross-country missions. The implemented system enabled the vehicle to drive continuously at 10 km/hr over rough, barren terrain [2].

These two systems are compared in [5].

Over the past year, we have shifted our focus from Mars to the Moon, as well as from missions driven exclusively by science objectives to missions that jointly satisfy entertainment and scientific purposes. Given these changes, we have undertaken a transition from walkers to rolling robots, and from range to video sensing.

Despite all these changes, perception remains as one of the critical technologies enabling mobile robots to operate autonomously for extended periods of time in rugged, natural, unstructured environments. Thus, we continue to investigate new approaches to the old problems of avoiding obstacles and fixing position, all to enable robotic exploration of planetary surfaces.

In this paper, we present new approaches and results in stereo driving and in position estimation. Both approaches are entirely passive; they do not require scanning or transmission of electromagnetic radiation into the environment. In addition, both are consistent with our aim for maximal rover self-reliance; they diminish requirements for close human supervision. We are validating experimentally these approaches on rovers—including the Ratler and the HMMWV (Figure 1)—and in outdoor trials. Future efforts will transfer the

developed technology into Lunar Rover demonstration and flight programs.



Figure 1: Rover testbeds: Ratler (left) and HMMWV (right)

In Section 2, we present an innovative stereo driving approach that bypasses traditional three-dimensional reconstruction. In Section 3, we develop position estimation approach that computes the latitude and longitude of a camera based on its observations of the Sun or other celestial bodies. In Section 4, we summarize the contributions, and discuss future directions.

## 2 Stereo Driving

Conventional stereo systems for navigation use three basic steps. First, matches between pixels in the left and right images are computed. Second, the matches are converted to points in 3-D space. Finally, the 3-D points are arranged into a map which is used for evaluating possible vehicle paths and for selecting the safest path.

This conventional approach has been demonstrated successfully in a limited number of systems [9, 14]. However, this approach suffers from several limitations that compromise its use in full-fledged navigation systems, especially in areas that cannot be easily accessed, e.g., as in planetary exploration scenarios. In particular, one major limitation is the need for precise camera calibration.

We set out to develop an alternative more general and robust approach by eliminating the conversion to Cartesian coordinates and the building of a dense elevation map. More precisely, our approach is to initially compute some of the parameters of the stereo system by matching natural features in the environment. These parameters are used at run-time for computing stereo matches between the two images and for deriving geometric cues such as the elevation of the terrain at the matched pixels. The parameters are also used for projecting candidate paths into the images. The candidate paths are evaluated by comparing them with the distribution of elevation values in the image. An optimal vehicle path is selected based on the result

of the evaluation. The image processing and the path evaluation are fast enough that the vehicle path can be updated at a rate high enough to permit continuous motion at moderate speed.

The remainder of this section is organized as follows; in Section 2.1, we describe our algorithm for stereo matching; in Section 2.2, we describe our approach to computing relative elevations from stereo matches. In Section 2.3, we describe how the stereo system is integrated into a complete driving module, and, particular, how paths can be evaluated based on the result of stereo matching; finally, in Section 2.4, we discuss the performance of the system based on experiments on the HMMWV testbed. We also discuss our current work and future plans, including moving the system to the Ratler.

### 2.1 Stereo Matching

Many conventional systems are limited in that, in order to facilitate the initial search for matches between left and right images, their cameras either need to be aligned so that the epipolar lines are the scanlines of the images or they need to be precisely calibrated with respect to a common external calibration target in order to derive the epipolar geometry.

Recent work in the area of weak calibration shows that the epipolar geometry can be derived by matching features between the two images without the use of an external calibration target [3, 4]. In practice, we use the program described in [17] in order to compute the epipolar geometry.

Moreover, the epipolar geometry can be used for computing a 2-D rectification transformation which brings the images into a standard configuration, irrespective of the physical configuration of the cameras. Once the rectification matrices are computed, rectifying an image is fast because the mapping between pixel positions in the original image and the corresponding positions in the rectified image is stored in a look-up table. The rectification of a full 480x640 image takes 0.2s on a Sparc-10 workstation.

Once both images are rectified, the best match to a pixel in the left image is found by evaluating a normalized correlation criterion at the pixels in the right image along the corresponding scanline. The search is limited to a pre-set disparity interval  $[d_{min}, d_{max}]$ .

The stereo algorithm is implemented as a random access function which takes as input a pixel position  $(x_r, y_r)$  and a disparity interval  $[d_{min}, d_{max}]$ , and outputs the corresponding disparity. We chose this implementation over the conventional raster scan implementation because we need to compute only a small

number of matches compared to the total number of pixels in the image, **as** we will show in Section 2.3. Furthermore, **as** we will show in Section 2.2, the anticipated range of disparity  $[d_{min}, d_{max}]$  can be adjusted dynamically at each pixel.

It is well known that any stereo matching algorithm is prone to false matches due to lack of texture, repetitive texture, and occlusions. In our case, false matches are of great consequence because they directly impact the behavior of the entire driving system. To reduce instances of false matches, we implemented three levels of mismatch detection. First, pixels at which the variation of intensity in the neighborhood,  $\sigma$ , is insufficient are not matched. This test eliminates points on uniform surfaces such **as** roads. Second, matches with an insufficient correlation value  $\rho$  are not matched. This test eliminates points in occluded parts, for example. A third test finds the local extrema of the correlation curve  $\rho(d)$  at each pixel and discards the pixel if there exists a local maximum  $d_i$  such that  $\rho_{max} - \rho(d_i) < \delta\rho$ , where  $\rho_{max}$  is the global maximum value of correlation. The current values used in the three tests are:  $\sigma_{min} = 2$ ,  $\rho_{min} = 0.5$ , and  $\delta\rho = 0.05$ .

## 2.2 Computing Relative Elevation from Stereo Matches

As previously mentioned, a limitation of the conventional approach to stereo driving is that it relies on precise metric calibration with respect to an external calibration target in order to convert matches to 3-D points. From a practical standpoint, this is a serious limitation in scenarios in which the sensing hardware cannot be physically accessed, such **as** in the case of planetary exploration. In particular, this limitation implies that the vision system must remain perfectly calibrated over the course of an entire mission. From a philosophical point of view, navigation should not require the precise knowledge of the 3-D position of points in the scene: What is important is how much a point deviates from the reference ground plane, not its exact position.

Based on these observations, we developed an approach in which a relative measure of height with respect to a ground plane is computed from the matches without requiring the knowledge of the full set of camera parameters. This height is *relative* in the sense that it is **a** multiple of the true height by a unknown scale factor. We now describe the construction of the relative height in detail. The geometry described below **has** been used in earlier work in which a point is classified **as** belonging to one of two halfspaces based on its projections in two uncalibrated images [11].

Let us consider first a flat ground plane observed by two cameras. We assume that the only known information about the geometry of the cameras is the epipolar geometry. Let  $\mathbf{M}$  be a generic point on the plane and  $m_l$  and  $m_r$  be its projections in the left and right images, respectively. We represent points in the image plane by 2-D projective coordinates,  $m_i = [\mathbf{u} \ \mathbf{v} \ \mathbf{w}]^t$ , where the usual Cartesian image coordinates are  $\mathbf{x} = \mathbf{u}/\mathbf{w}$  and  $\mathbf{y} = \mathbf{v}/\mathbf{w}$ . It can be easily shown that for any point  $\mathbf{M}$ , the projections are related by a linear projective transformation, or homography,  $\mathbf{H}: m_r = \mathbf{H}m_l$ . In this relation, the symbol  $=$  means that the two sides are equal in the projective sense, i.e., that their coordinates are proportional. Intuitively,  $\mathbf{H}$  maps a pixel from the right image to its location in the left image assuming that the corresponding 3-D scene point lies on the plane (Figure 2(a)).

The homography  $\mathbf{H}$  is a 3x3 matrix defined up to a scale factor.  $\mathbf{H}$  can be easily estimated from real images in the following way. First,  $n$  features  $m_l^i$  are selected in the left image and the corresponding pixels  $m_r^i$  in the right image are computed using the algorithm of Section 2.1. Then, the parameters of  $\mathbf{H}$  are computed by solving for the least-squares criterion:  $\sum_i \|\mathbf{H}m_l^i - m_r^i\|^2$ . The features used in the computation of  $\mathbf{H}$  may be anywhere in the image. Moreover, computing  $\mathbf{H}$  does not require any information on the actual 3-D positions of the scene points used **as** features.

We now show that  $\mathbf{H}$  is all we need to compute a relative elevation map. Consider a world point  $\mathbf{M}$  not necessarily on the ground plane and its projections  $m_l$  and  $m_r$ . Let us assume that we also have defined once and for all a "reference" point  $\mathbf{M}'$  described by its projections  $m_l^*$  and  $m_r^*$ .  $\mathbf{M}'$  may be anywhere in space so long **as** it is not on the reference plane. Of course the point  $\mathbf{M}'$  is not known; only its projections in the images are known. Let us consider now the image points  $p_r = \mathbf{H}m_l$  and  $p_r^* = \mathbf{H}m_l^*$ . Point  $p_r$  (resp.  $p_r^*$ ) is the point at which  $\mathbf{M}$  (resp.  $\mathbf{M}'$ ) would be projected in the right image if it were on the ground plane. Finally, consider the point  $q_r$ , intersection in the right image of the two segments  $m_r m_r^*$  and  $p_r p_r^*$ . Since  $m_r m_r^*$  is the projection of the segment  $\mathbf{M} \mathbf{M}'$  and  $p_r p_r^*$  is the projection of a line segment contained in the plane, the intersection point  $q$  must be the projection of the intersection **of**  $\mathbf{M} \mathbf{M}'$  with the reference plane (Figure 2(b)).

The previous reasoning shows that we now have a way to compute the intersection of the line joining a point and a reference point with a reference plane without computing the actual 3-D position of the point. Now, let  $\mathbf{M}'$  become the point at infinity in a given

direction. The intersection  $q$  becomes the image of the projection of  $M$  onto the reference plane in the direction given by  $M^*$  (Figure 2(c)). Because  $q$  is now the projection of  $M$ , the distance in the image  $m_r q$  is directly related to the height of  $M$  with respect to the reference plane. In practice, we use another reference point  $P^{unit}$  which we declare to be at height one from the reference plane. If  $q_r^{unit}$  is the image of the projection of  $P^{unit}$  on the reference plane, then the height is defined as:  $h = qm_r / q_r^{unit} m_r^{unit}$  (Figure 2(d)).

In affine geometry, this definition of height is exact in the sense that  $h$  is proportional to the distance between  $M$  and the reference plane. In the projective case, an additional reference plane is required in order to define a concept of projective height. However, the affine approximation is accurate enough for the purpose of navigation because the heights are computed at relatively long range from the camera and over a relatively shallow depth of field.

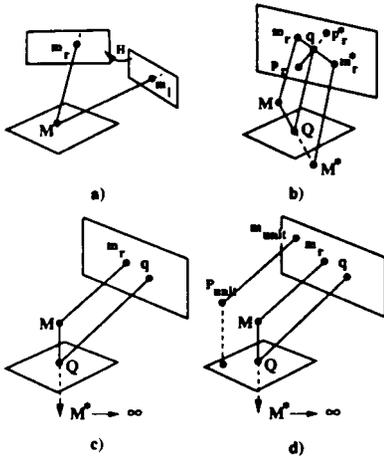


Figure 2: Computation of relative height: (a) Relation between the projections of a point in the reference ground plane in the images; (b) Computing the intersection of a line segment with the ground plane; (c) Computing the projection of a point on the ground plane; (d) Computing the height relative to "unit" point.

The relative height is also used for limiting the search in the stereo matching. More precisely, we define an interval  $[h_{min}, h_{max}]$  of heights which we anticipate in a typical terrain. This interval is converted at each pixel to a disparity range  $[d_{min}, d_{max}]$ . This is an effective way of limiting the search by searching only for disparities that are physically meaningful at each pixel.

In addition to the relative elevation, a measure of slope relative to the ground plane can also be com-

puted under minimal knowledge of camera geometry. We do not describe the slope evaluation algorithm here because it is not integrated in the current system. We refer the reader to [12] and [1] for a detailed presentation.

## 2.3 Navigation

The navigation part of the stereo driving system is based on an existing CMU Distributed Architecture for Mobile Navigation (DAMN), which provides a framework within which individual components can be combined into a coherent navigation system [6].

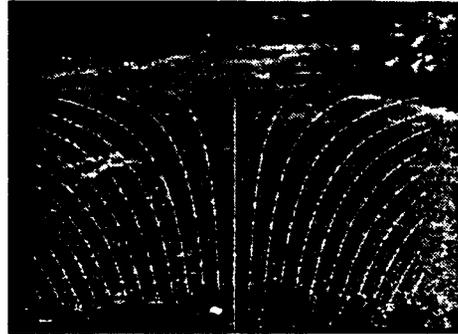


Figure 3: Discrete set of arcs projected in the image.

In DAMN, modules communicate using a common set of arcs of constant radii [13, 2]. Each module describes its current view of where the vehicle should steer next in terms of votes for a set of arcs. The votes from multiple behaviors are combined by an arbiter which decides on the best command to send to the vehicle controller.

Figure 3 shows the set of arcs used in the current system projected in one of the images. The parameters necessary for projecting the arcs in the image can be computed by a procedure similar to the one used in Section 2.2. The difference is that this procedure uses features tracked over several frames as the vehicle is moving instead of features detected in static images.

The obstacle avoidance module we developed periodically produces recommendations for the best steering directions based on stereo measurements. In other words, this module computes which arcs are safe without attempting to steer the vehicle in a preferred direction. The final steering commands is selected by an arbiter module which is responsible for the arbitration between the different modules involved in a given mission.

The recommendations are encoded in the form of an array of votes  $v_i, i = 1, \dots, N$  for a fixed set of possible arcs of radii  $R_i, i = 1, \dots, N$ . The votes are encoded as

continuous values between -1 and +1. The semantics of the votes are as follows: If  $v_i = -1$ , the behavior has determined that the arc of radius  $R_i$  should not be executed; if  $v_i = +1$ , the behavior has determined that the arc of radius  $R_i$  is an optimal arc to follow; if  $-1 < v_i < +1$ , the behavior indicates a level of preference for this arc proportional to  $v_i$ .

In order to compute these votes, we measure the height and the position on the ground of a few hundred points in the field of view using the algorithms of Section 2.2. Since we know the transformation between the ground plane and the image plane, we select these points on a grid in the image so that they cover a surface on the ground that fits the navigation requirements. The example in Figure 4 shows the grid of measurement points in a typical example and Figure 5 shows their projections on the ground. These points cover an area on the ground ranging between 12 and 32 meters from the robot.

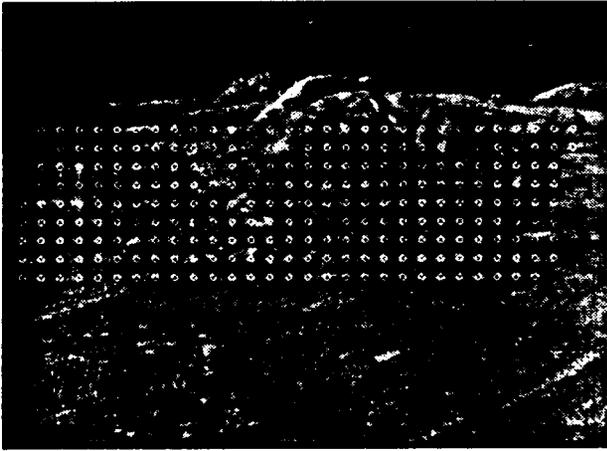


Figure 4: Grid of measurement points.

The obstacle avoidance behavior first computes a vote  $v_k^i = v(R_i, x_k, y_k)$  for all radii  $R_i$  and all points  $(x_k, y_k)$  (Figure 6). Intuitively,  $v_k^i$  is the vote that would be assigned to cell  $(x_k, y_k)$  if it were an obstacle. For reasons of space, we give only a qualitative description of the computation of  $v_k^i$ . The detail algorithm is described in [2] and [6].

The computation of  $v_k^i$  is based on three parameters:  $L_{max}$ , the distance along the arc after which an obstacle does not matter;  $L_{min}$ , the distance along the arc before which an obstacle causes the arc to be removed from consideration; and  $K$ , a coefficient which indicates how fast  $v_k^i$  increases as function of the lateral distance between the arc and the obstacle cell. Qualitatively,  $v_k^i$  decreases as the distance  $L_k^i$  of the cell along the arc decreases and is set to 1 (resp -1) if  $L_k^i$  if

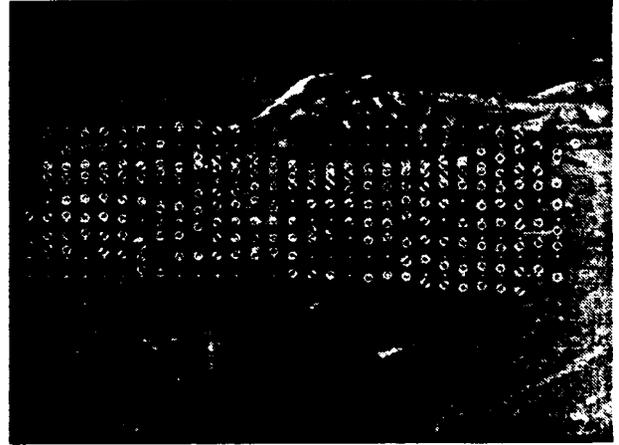


Figure 5: Measured points from Figure 4 projected on the ground.

greater (resp. lower) than  $L_{max}$  (resp.  $L_{min}$ ). Moreover, for cells that do not directly intersect the arc,  $v_k^i$  increases with the lateral distance  $D_k^i$  between the arc and the cell.

Given a measured point  $P_k$  at grid location  $(x_k, y_k)$ , and with relative height  $h_k$ , the vote for arc  $i$  is a function of  $h_k$  and is computed as:  $V_i^k(h_k) = \lambda(h_k)v_k^i + (1 - \lambda(h_k))$ . In this expression,  $\lambda(h) = e^{1-gh}$  is a function that is small when  $h$  is small, and that converges to 1 as  $h$  increases (Figure 7). With this definition,  $V_i^k(h_k)$  is small when the point is close to the ground plane and becomes close to  $v_k^i$ , the obstacle value, as the elevation increases.

The gain  $g$  controls the sensitivity of the system to variations in height. A large value for  $g$  causes the system to be sensitive to small variations in terrain elevation; a small  $g$  causes the system to respond only to large objects. Given now a set of measured points  $\{P_k\}$ , the final votes for each arc  $i$  are obtained by taking the minimum values of the  $V_i^k$  for each measured point.

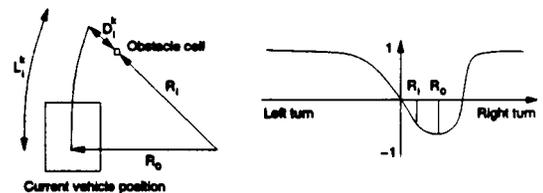


Figure 6: Influence of an obstacle cell

Using this definition of the votes, we can precompute in a grid all the  $v_k^i$  for all the arcs and for all the possible cells  $(x_k, y_k)$  in front of the robot. This speeds

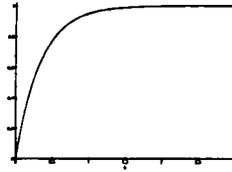


Figure 7: Function  $\lambda(h)$

up considerably the computation of the final votes.

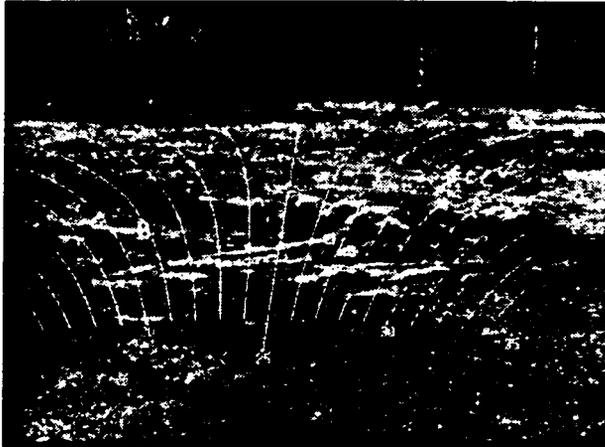


Figure 8: Two measured points. The left one is on a plane ground, the right one is on a bump and represents a small obstacle.



Figure 9: Votes corresponding to the measured points of Figure 8; Left: Point on the ground plane; Right: Point on an obstacle.

Figure 8 shows two measured points and their influence on the arcs. The first one, on the left, is on the ground plane and is not an obstacle. Its influence on the arcs is shown in Figure 9 on the left. If only one point had been measured, the votes for the arcs on the right would have been slightly higher. (Indeed, the votes for each visible arc are close to 1, and decrease a little for the arcs on the left.) On the contrary, the votes generated by the measure of the second point in the right of the image presented in Figure 9 show clearly that it represents a small obstacle. The values for the votes corresponding to the arcs on the right are much lower than the ones on the left.

## 2.4 System Performance and Future Work

In this section, we present results obtained using the HMMWV in a cross-country test area. In all the examples, the homography of the reference plane is computed on an initial image by using up to ten features on the ground plane. The relative height, stereo matching and rectification use the algorithms of Section 2.1 and Section 2.2. The evaluation of relative slope is not yet integrated in the system. Three hundred points are used in a typical configuration of the system. The computation times are 0.1s for the arc evaluation and 0.5s for stereo matching and rectification on a Sparc-10 workstations. The navigation module runs on a dedicated workstation while the other modules, such as the DAMN arbiter, run on a separate workstation.

The first example (Figure 10) shows a large obstacle in front of the robot, on the right. The small white dots are the points that have been measured, and the white circles are their projection on the ground. The bump has been detected since all the measured points located on its surface have been correctly projected at the bottom of the bump, on the ground plane. The corresponding votes are shown in Figure 11. The votes corresponding to the arcs on the left of the obstacle have a much better value than the ones going in the direction of the obstacle. The votes for the arcs on the extreme left and right are set to -1 because they are not visible in the image.

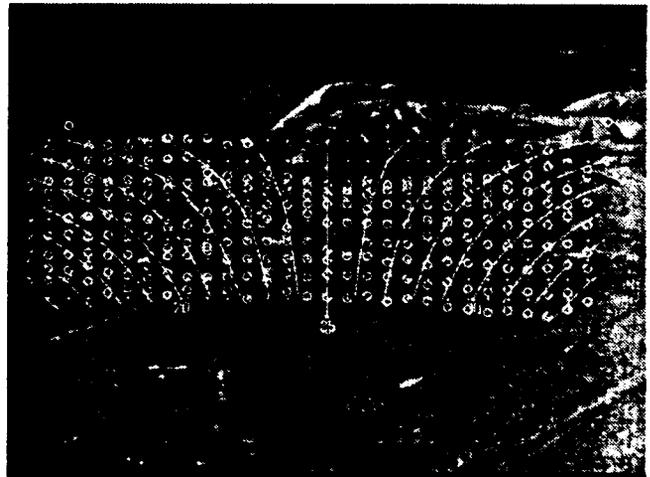


Figure 10: Result of the relative height computation in the vicinity of a large obstacle. The dots indicate the location of the measurement points; the circles indicate their projection on the ground.

The example in Figure 12 is similar; a small bump

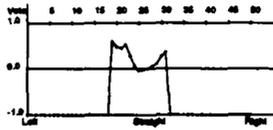


Figure 11: Votes corresponding to the measures from Figure 10

on the right **has** been detected. An interesting observation is that the votes corresponding to the arcs going toward the obstacle (Figure 13) have a value bigger than the **ones** going toward the obstacle detected in the first example (Figure 10). The **reason** is that the obstacle in this example is smaller than the one in the first example.

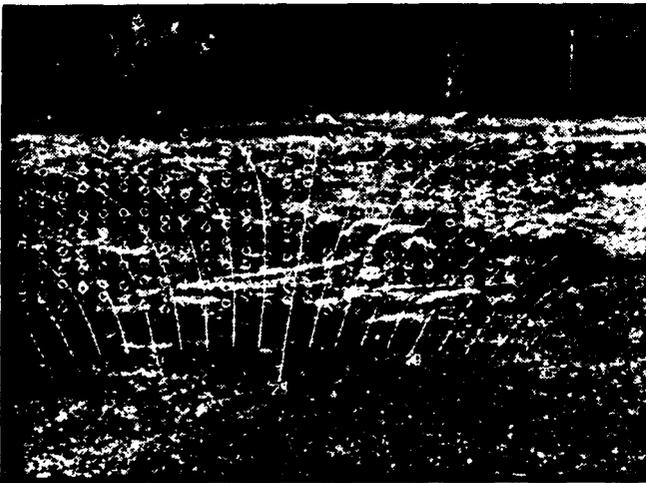


Figure 12: Result of the relative height computation in the vicinity of a small obstacle

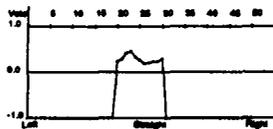


Figure 13: Votes corresponding to the measures from Figure 12

In the third example (Figure 14 and Figure 15), the robot is closer to the obstacle. Therefore, this time, the votes corresponding to the arcs going toward the obstacle have a smaller value than before (compare Figure 13 and Figure 15). This is because both the distance and the height of the detected obstacles are taken into account during the computation of the votes.

We demonstrated this initial system over short paths in our test site. We processed up to one hundred im-



Figure 14: Result of the height computation around the obstacle of 12 at closer range

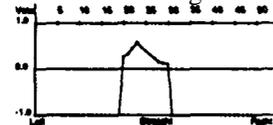


Figure 15: Votes corresponding to the measures from Fig 14

ages over a single path. Our current work involves adding a module that keeps a history of all the votes instead **on** relying solely **on** the votes from the most recent image, and integrating the slope computation into the system. These two extensions will make the system more robust and will allow for longer demonstrations. We are also automating some aspects of the system, such **as** the selection of the features in the initial training images of the reference plane. Longer-term goals include moving the system to the Ratler and using two pairs of cameras for increased field of view.

### 3 Sun Tracker

In this section we describe a system for localization on Earth given measurements of Sun altitude. This problem, known to ancient civilizations, is part of the basic knowledge of any **sea** navigator. Celestial information **has** been used in space applications for attitude control and accurate positioning [16]. Yet celestial navigation **has** not been heavily mentioned in connection with autonomous rovers. This comes **as** a surprise since planetary rovers will not be connected with GPS or similar localization systems and will need reliable means of checking dead reckoning data.

The localization system described here is based on

a camera with telephoto lenses. Our purpose was to validate the idea of celestial localization through image processing.

The usual procedure in marine applications is to get several altitude measurements from different stars and to determine position from them. We will modify this problem by tracking only the Sun over time; the result will be equivalent to obtaining several star measurements simultaneously.

### 3.1 Basic Definitions

In this section we summarize some basic concepts that are used in the celestial navigation literature. Some concepts are specific to Earth, even though they could be adapted to other celestial bodies. An extensive treatment is given by Hobbs [7].

Basic facts are:

- Position on Earth is given by a pair latitude/longitude. We take south latitude **as** negative and west longitude **as** negative.
- Every celestial body can be projected onto the terrestrial surface by considering the line that goes from the center of Earth to the body. The point of intersection between the terrestrial surface and this line is the geographic position (GP) of the body.
- The latitude of a body in the celestial sphere is denoted declination.
- Since the Sun is moving continuously, its GP longitude changes. We can recover GP longitude if we know Greenwich Mean Time. However, GMT gives only a mean position; we need to subtract a small correction that varies daily. This quantity is called *equation of time* and it is tabulated in the Astronomical Almanac [15].

At any given time, a measurement of Sun altitude constrains the observer to be in a circle on the terrestrial surface. Such a circle is called circle of equal altitude. A circle of equal altitude is defined by a measurement of Sun altitude *and* knowledge of the GP of Sun. Two measurements of altitude are enough to constrain the observer to two points; three measurements are enough to constrain the observer to one point. More measurements of altitude overconstrain position and can be used to check or correct previous measurements. Figure 16 illustrates a situation where two stars are measured, yielding two possible solutions. In general, the solutions are very far from each other and one can be discarded based on other sources, for example, dead reckoning.

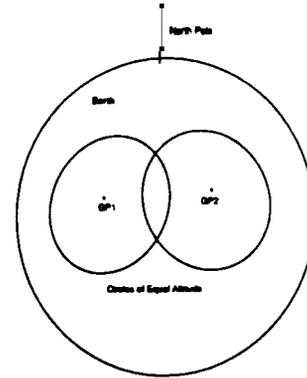


Figure 16: Two Measurements of Altitude: GPs and Circles of Equal Altitude

In marine applications, altitudes are obtained with sextants, and the final position is obtained through table look-up. Tables with necessary data are published regularly [7]. We propose below a fully automated method of calculation for position, based on non-linear least squares filtering.

### 3.2 Obtaining Circles of Equal Altitude

We need to measure Sun altitude and obtain the GP of the Sun. Sun altitude is measured with a camera and telephoto lenses. The camera is attached to a fixed horizontal platform and is aimed at the Sun. A series of neutral density filters is attached to the telephoto lens so that the Sun can be observed directly.

#### 3.2.1 Obtaining Altitude

Sun images are sent to a workstation, which performs the following operations:

1. A histogram is made with the pixel values, and the image is thresholded so that only 20% of pixels remain valid.
2. A grassfire transform is performed in order to remove noisy features in the thresholded image.
3. A region coloring algorithm is used to connect all remaining regions in the thresholded image. The algorithm scans the image and expands every region **as** much **as** possible.
4. Each connected region is analyzed. The best region based on area and aspect ratio is declared to be the Sun image.
5. The centroid of the chosen region is calculated and reported **as** the position of the Sun in the image.



Figure 17: Typical Sequence of Images in Altitude Measurement

Figure 17 shows a typical sequence of images generated by the system. The first image is the raw image obtained using the neutral density filters and the telephoto lens. The centroid determined by the system is marked with a cross in this image. The second image is the thresholded and filtered image.

The determination of the centroid of the Sun allows us to obtain the angle between the optical axis and the ray that emanates from the center of the image to the Sun. This angle is:

$$\alpha = \arctan\left(\frac{u - v_o}{f_v}\right)$$

where:

$u$  Row of centroid of Sun (in pixels).

$v_o$  Row of center of image (in pixels).

$f_v$  Focal length in camera's  $y$  axis (in pixels).

The parameters  $u$ , and  $f_v$  are intrinsic parameters of the camera and their values were obtained through a customized calibration procedure. Details of this procedure are left to Section 3.4.

In order to get altitude, we need to obtain the angle between the optical axis of the camera and the horizontal plane with respect to Earth's gravity. This angle is measured with a digital inclinometer. Raw altitude is given by:

$$h_a = \alpha + \beta$$

where  $\beta$  is the inclinometer reading.

There are some additional factors that must be taken into account in this type of measurement [7]. The gradual bending of an incoming light ray in the Earth's atmosphere is called atmospheric refraction; it must be corrected. Atmospheric refraction depends on the raw altitude, because rays bend in different ways when they reach the atmosphere at different angles. The refraction correction  $R$  in degrees is given by:

$$R = \frac{0.0167}{\tan\left(h_a + \frac{7.31}{h_a + 4.4}\right)}$$

The difference in the apparent altitude of a body within the solar system as viewed from the surface of

Earth and from its center is called parallax; this difference must be corrected since the images of the Sun are taken from its surface, not from its center. The parallax correction  $PA$  for the Sun is:

$$PA = 0.0024 \cos h_a.$$

The final observed altitude is  $h_o = h_a - R + PA$ .

### 3.2 Obtaining GP of Sun

The astronomical position of the Sun is fixed for a given Greenwich Mean Time value. The Astronomical Almanac [15] annually reports declination and the equation of time for all days of the year. A set of formulas is given in the Almanac for automatic calculation of these quantities. Since these formulas are easily available we do not repeat them here. Declination of the Sun directly gives Sun latitude for a given day. The equation of time must be used to obtain Sun longitude, together with precise Greenwich Mean Time. We obtain Greenwich Mean Time directly from the computer clock, corrected to the nearest second by a network call to a national server.

### 3.3 Computation of Localization Estimates

The intersection of all circles of equal altitude is performed numerically. The result of the measurement process previously outlined produces:

- A sequence of altitudes, indexed by time.
- A sequence of GPs of Sun, indexed by time.

From each pair altitude/GP, it is possible to define a circle of equal altitude on the surface of the Earth. As a two-dimensional circle in three dimensional space. The important point is that, if we parameterize the Earth surface in longitude/latitude coordinates, *the circles of equal altitudes are not two dimensional circles*. Instead, they are the loci of solutions of [7]:

$$\sin I \sin l_{GP} + \cos I \cos l_{GP} \cos(\lambda_{GP} + A) = \sin h, \quad (1)$$

where  $(I, A)$  is a point in latitude/longitude space,  $(l_{GP}, \lambda_{GP})$  is the GP of Sun in this space and  $h$ , is the measured altitude.

If we have  $n$  pairs altitude/GP, we have a series of  $n$  equations in two unknowns. We set the following non-linear least squares problem:

$$(i, \lambda) = \arg \min_{i, \lambda} \left[ \sum_{i=1}^n (\sin l \sin l_{GP_i} + \cos l \cos l_{GP_i} \cos(\lambda_{GP_i} + \lambda) - \sin h_{oi})^2 \right] \quad (2)$$

The minimization starts with a grid of “presumed positions.” The user defines a rectangular region in terms of latitude/longitude and spacings for a grid that covers a region on Earth. The size of the initial grid is arbitrary; the larger the grid, the longer the minimization will take.

If we have a large number of measurements and a large grid, the minimization involved will be impractical. Instead, we use the following procedure instead. Initially set  $n$  to 1 and minimize (2) using only the first pair altitude/GP. For each point in the grid, minimize numerically equation (2). This produces a list of solutions. Solutions that are too close are combined. The remaining solutions will serve as seeds for the next minimization process. The minimization step is repeated  $n$  times, each time including a new measurement and combining solutions that are too close.

The remaining solutions are the result of the whole procedure.

### 3.4 System Calibration

The system is highly sensitive to errors due to miscalibration. Three measurements are made every time a pair altitude/GP is obtained:

- o An image is acquired.
- o An angle is obtained through a digital inclinometer.
- o Greenwich Mean Time is stored.

Measurement of time is precise and can be easily calibrated. We are using a digital inclinometer with simple re-calibration procedure, but limited to  $\pm 0.2$  degrees of precision. Currently the precision of our system is limited by the inclinometer precision. Finally, camera calibration is necessary for minimizing errors in the detection of Sun altitude.

We calibrated our camera/lens apparatus using a calibration method proposed by Robert [10]. The method requires a series of measurements of a target object. The dimensions of the target object must be precisely known. The user starts the calibration process by indicating correspondences in the calibration image and the real target object. The method generates the perspective projection matrix [4], which contains all the relevant information about the imaging system.

Since we work with huge zoom lenses, our images are essentially ruled by orthographic projection. In fact, our perspective projection matrix is (up to a constant

factor):

$$\begin{bmatrix} 39.764 & -40.567 & 1.365 & 290.104 \\ -12.129 & -13.576 & -52.750 & 465.703 \\ 0.00249 & 0.00172 & -0.00041 & 1.000 \end{bmatrix}.$$

Numeric determination of intrinsic parameters using a perspective model is highly unstable due to ill-conditioning of this matrix. Instead we set the first three elements of the last column to zero; by so doing we obtained an orthographic projection matrix. Obtaining the image center and the magnification factor is immediate. In our case (units in pixels):

$$\begin{aligned} v_o &= 242 \\ f_v &= 23068.5. \end{aligned}$$

### 3.5 Experiments

The system described here was implemented and tested in a real environment. A fixed platform was built to which the camera/lens/filter apparatus was attached. The whole platform was leveled using the digital inclinometer. Series of measurements were taken at different times of day, in periods of one hour. In each hour, six or eight measurements were taken. All experiments were performed at  $(40^\circ 26', -79^\circ 59')$  latitude/longitude (obtained with GPS). We obtain accuracy of  $\pm 0.8^\circ$ , comparable to the accuracy of the inclinometer.

We show some representative results in what follows. Figure 18 shows the minimization of Equation (2) for two independent sets of two measurements. The first two measurements were taken respectively at 22:50:24 and 23:0:41 (GMT), June 3 1994. The temporal distance between measurements was small, so the circles of equal altitude are almost equal. Minimizing equation (2) with both measurements yields  $(40^\circ 22', -79^\circ 26')$ . Performance varies with random disturbances; the second set of measurements was taken May 23 1994, respectively at 22:11:23 and 22:37:59 (GMT). Result of both measurements is  $(40^\circ 37', 79^\circ 52')$ .

## 4 Discussion

In this paper, we have presented two new approaches to planetary rover perception.

One approach concerns stereo driving without 3-D reconstruction. This approach begins with weakly calibrated stereo images, and evaluates the traversability of terrain using shape indicators such as relative slope and relative elevation. The approach then evaluates candidate paths based on the traversability analysis,

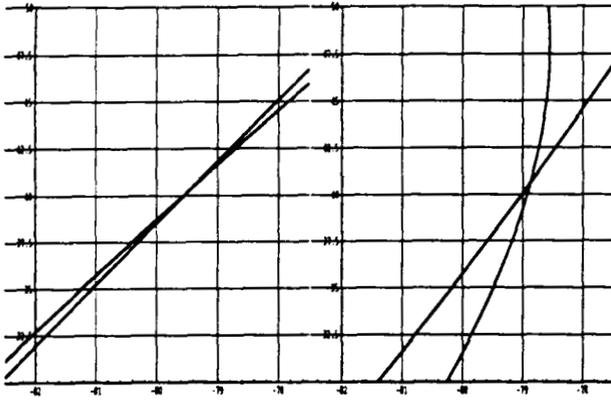


Figure 18: Data from Sun Measurements

and computes new driving commands for the vehicle based on the path evaluations.

The second approach involves estimating vehicle position by observing the Sun. At a given time, a measurement of the Sun's altitude constrains the observer to lie on a circle on the terrestrial surface called the circle of equal altitude. We determine the position of the observer by intersecting circles of equal altitude identified at different times.

To date, we have developed these approaches enough to be confident of their feasibility, and to be convinced of their potential effectiveness. However, considerable effort is still required to achieve the levels of technical maturity and generality required for Lunar Rover demonstration and flight programs.

Future work on stereo driving will concentrate on extensive testing on robot vehicles, and refinements based on this experience. Our goal is a fully automated system operating with uncalibrated stereo cameras demonstrated on several platforms.

Future work on sun tracking will concentrate on achieving altitude measurement accuracies better than 0.1 degree. This will require design of a highly stable camera platform, and use of a high precision digital inclinometer.

## References

- [1] F. Devernay. Calcul de Proprietes Differentielles de Formes Tridimensionnelles a Partir d'Images Stereoscopiques. Technical Report Rapport de stage, University of Nice, 1993.
- [2] D. Langer, J. Rosenblatt, and M. Hebert. A Reactive System for Autonomous Navigation in Unstructured Environments. In *Proc. International Conference on Robotics and Automation*, San Diego, 1994.
- [3] O. Faugeras. What Can be Seen in Three Dimensions with an Uncalibrated Stereo Rig. In *Proc. European Conference on Computer Vision*, Santa Margherita, 1992.
- [4] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.
- [5] M. Hebert and E. Krotkov. Autonomous Navigation in Natural Environments: Two Approaches. In *Proc. Computer Vision for Space Applications*, Antibes, 1993.
- [6] M. Hebert, D. Pomerleau, A. Stentz, and C. Thorpe. A Behavior-Based Approach to Autonomous Navigation Systems: The CMU UGV Project. In *To Appear in IEEE Expert*, 1994, 1994.
- [7] R. R. Hobbs. *Marine Navigation*. Naval Institute Press, Maryland, 1974.
- [8] E. Krotkov and R. Hoffman. Terrain Mapping for a Walking Planetary Rover. *IEEE Trans. Robotics and Automation*, 1994.
- [9] L. Matthies. Stereo Vision for Planetary Rovers: Stochastic Modeling to Near-Real Time Implementation. *International Journal of Computer Vision*, 1(8), 1992.
- [10] L. Robert. Camera Calibration without Feature Extraction. In *Proc. International Conference on Pattern Recognition*, 1994.
- [11] L. Robert and O. Faugeras. Relative Positioning and 3D Convex Hull Computation from a Weakly Calibrated Stereo Pair. In *Proc. International Conference on Computer Vision*, Berlin, 1993.
- [12] L. Robert and M. Hebert. Deriving Orientation Cues from Stereo Images. In *Proc. European Conference on Computer Vision*, Stockholm, 1994.
- [13] J.K. Rosenblatt and D.W. Payton. A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control. In *Proc. of the IEEE/INNS International Joint Conference on Neural Networks*, volume 2, pages 317-324, Washington D.C, 1989.
- [14] B. Ross. A Practical Stereo Vision System. In *Proc. Computer Vision and Pattern Recognition*, New York, 1993.
- [15] US Navy Department. *Astronomical Almanac*, 1994.
- [16] J. R. Wertz. *Spacecraft Attitude Determination and Control*, volume 73 of *Astrophysics and space science library*. Reidel, Boston, 1978.
- [17] Z. Zhang, R. Deriche, O. Faugeras, and Q.T. Luong. A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. Technical Report 2273, INRIA, 1994.

