??, ??, 1–27 (??)

Outdoor Visual Position Estimation for Planetary Rovers*

FABIO COZMAN**

fgcozman@usp.br

Laboratory of Automation and Systems, University of São Paulo, São Paulo, Brazil

ERIC KROTKOV

krotkov@cytometrics.com

Robotics Institute, Carnegie Mellon University, Pittsburgh PA 15213

CARLOS GUESTRIN

guestrin@stanford.edu

Computer Science Department, Gates Computer Science Building, Stanford University, Stanford CA 94305

Received ??; Revised ??

Editors: ??

Abstract. This paper describes (1) a novel, effective algorithm for outdoor visual position estimation; (2) the implementation of this algorithm in the Viper system; and (3) the extensive tests that have demonstrated the superior accuracy and speed of the algorithm. The Viper system (*Vi*sual *P*osition *E*stimator for *R*overs) is geared towards robotic space missions, and the central purpose of the system is to increase the situational awareness of a rover operator by presenting accurate position estimates. The system has been extensively tested with terrestrial and lunar imagery, in terrains ranging from moderate — the rounded hills of Pittsburgh and the high deserts of Chile — to rugged — the dramatic relief of

the Apollo 17 landing site — to extreme — the jagged peaks of the Rockies. Results have consistently demonstrated that the visual estimation algorithm estimates position with an accuracy and reliability that greatly surpass previous work.

1. Introduction

This paper describes a novel, effective algorithm for outdoor visual position estimation. The algorithm was developed in the context of long-duration space missions in which a rover must be guided from Earth by human operators.

The algorithm has been implemented in the Viper system (*Vi*sual *P*osition *E*stimator for *R*overs), an interface for rover teleoperation with position estimation capabilities. The objective of the system is to inform operators about the rover location, so that operators can keep a sense of direction and an understanding of the rover's environment. The system has been extensively tested with terrestrial and lunar imagery, in terrains ranging from moderate — the rounded hills of Pittsburgh and the high deserts of Chile — to rugged — the dramatic relief of the Apollo 17 landing site — to extreme — the jagged peaks of the Rockies. Results have consistently demonstrated that the visual estimation algorithm estimates position with an accuracy and reliability that greatly surpass previous work.

*This research was conducted at the Robotics Institute, Carnegie Mellon University, and was partially sponsored by NASA under Grant NAGW-1175. The views and conclusions contained in this document are those of the author and should not be interpreted as representing official policies, either expressed or implied, of the funding agencies.

**Supported by a scholarship from CNPq, Brazil.

Position estimates are generated from two sources of information: visual imagery and attitude data collected and relayed by the rover, and maps of the rover's environment. Visual data about the skyline surrounding the rover is matched to data extracted from the maps; the position that leads to the best match is the best position estimate. The matching procedure is based on a probabilistic formulation: prior densities encode knowledge about the rover's location, and likelihood functions encode knowledge about the imaging device. The estimation algorithm is exact, handles sequences of measurements, and is particularly well-suited for low-dimensional problems in which grid representations can capture the space of all possible estimates.

The Viper system provides a complete infrastructure for teleoperation based on visual information. The system collects the raw sequence of images relayed by the rover, glues these images into a panorama, extracts visual information from the panorama, runs the position estimation algorithm and presents the estimates to the user. Images and maps are presented to the operator through an interface that displays information about incoming imagery, rover position and the relationship between map features and image features.

The objective of this paper is to describe the Viper system and the position estimation algorithm that forms its core, and to discuss the tests that were conducted with the system in a variety of environments. Preliminary versions of the system have been described previously [11]. The goal of this paper is to present the most recent implementation and discuss its performance.

2. The position estimation problem and the Viper system

The problem of interest is to estimate the pose of a mobile robot engaged in a lunar or planetary mission. This research project was developed in the context of the Lunar Rover group, a research team at the Robotics Institute, Carnegie Mellon University, focused on technologies for robotic exploration of the Moon [26, 54].

The pose of a robot is the six-dimensional vector that specifies (Figure 1):

- The Cartesian coordinates of the center of the robot. The *center* of the robot is defined here as the optical center of the camera that collects images of the robot's environment.
- The attitude of the robot. Attitude is specified by roll, pitch and yaw angles, indicating rotation around the x, y, and z axis respectively [32]. The *orientation* of the robot is the value of the yaw angle ϕ.

A topographic map of the robot's environment is assumed to be available. In this case, the two-dimensional *position* (the x, y coordinates) determines the elevation (the z coordinate). The position (x, y) of the rover is denoted by Γ.

While human navigation is well-understood in marine [18], aeronautic [31], spatial [53] and terrestrial [43] scenarios, automatic pose estimation for outdoor robotics has not been equally well-developed. Currently outdoor pose estimation relies primarily on dead-reckoning or GPS sensors. Dead-reckoning is only relative and degrades with distance [6]. GPS provides high absolute accuracy [15]; the disadvantage of GPS is that receivers cannot acquire satellite signals when satellites are occluded by trees, buildings, bridges, mountains, and even snow [43], and GPS is not suitable for lunar or planetary missions.

Other methods for position estimation exploit physical phenomena (like compasses and incli-

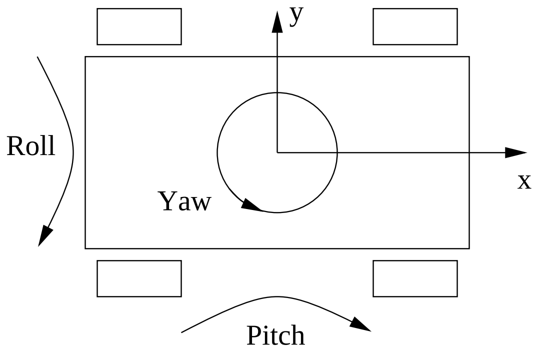


Fig. 1. Pose of a mobile robot (top view).

nometers) or natural/artificial landmarks [18]. Landmarks can be passive or active, as pose can be obtained by the reception of electro-magnetic waves from fixed stations [50]. When terrestrial landmarks are not visible, marine navigators use the Sun and other celestial bodies to obtain rough estimates of position (on the order of twenty kilometers) [10, 29]. Celestial bodies are also employed in space applications, where three types of sensors are responsible for generating attitude estimates: star sensors, Sun sensors and Earth sensors [53].

The conclusion to be drawn from this brief survey is that roll, pitch and yaw angles can be measured by appropriate equipment: compass and inclinometers on Earth; Sun and star trackers during lunar and planetary missions. In this context, the goal of this work is the recovery of position Γ given visual information. Visual information must be used because sonar functions only when immersed in a thick atmosphere, and laser rangefinders typically require moving parts and substantial power consumption.

This research was motivated by our experience in increasing the autonomy of teleoperated robots [27]. The basic idea that initiated the research was to offload position estimation tasks from the rover operator, so that the operator can drive the rover without getting disoriented or lost. The Viper system was conceived to assist operators driving remote mobile robots and to prevent the common episodes of loss of orientation [1, 23, 33]. Figure 2 summarizes the idea. The operator observes images from the rover and looks at a topographic map of the imaged area. The position of the robot is unknown but constrained to a region inside the topographic map. The images are analyzed and the position of the robot is estimated. Position information is overlaid on the maps and on rover-acquired images, so that the operator can grasp the relationship between images, maps and estimates. The interface presents an enhanced view of the robot's environment to the operator, re-

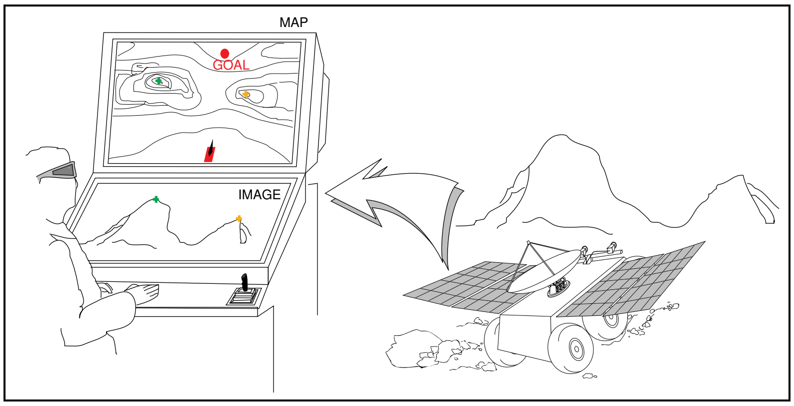


Fig. 2. Improving teleoperation of remote mobile robots.

sembling the augmented reality systems used for training and medical applications [3].

3. The many dimensions of position estimation

Position estimation is one of the critical capabilities of any autonomous device; as a consequence, position estimation has been investigated extensively [9, 20]. Here we review work that is closely related to our own.

3.1. Drop-off and updating problems

A position estimate can be generated with or without information about the initial state of the robot. At one extreme, the robot starts with complete knowledge of its pose; the robot is only required to update its pose. At the other extreme, the robot must start by locating itself in a map without any prior information. The latter situation is called the *drop-off* problem [46, 47].

This paper concentrates on the drop-off problem, because it is the most basic and challenging position estimation scenario; it is also the critical problem to be faced by a robotic probe landing on a celestial body. Updating problems can be addressed by the method presented in this paper with simple modifications on the algorithms; Section 6.4 discusses a test that employed multiple panoramas, acquired in different locations. To simplify the presentation, we assume that any mobile robot has a dead reckoning system that is accurate to the extent that errors incurred in relatively small displacements can be ignored.

3.2. Quantitative and qualitative methods

Most indoor mobile robots seek *quantitative*, numeric answers to questions regarding position estimates. The search for the best position estimate can be conducted in two different spaces: (1) the space of correspondences between image and map features [2, 24, 25], or (2) the space of position estimates [7, 8, 21, 40, 49]. The search in the space of position estimates is often conducted by quantizing this space into a two-dimensional array, an approach inspired by the *occupancy grids* developed for map making [13, 14, 34, 35].

Position can also be estimated *qualitatively*: instead of precise geometric estimates, only topological relationships are produced. This approach has been implemented in several indoor robots [22, 55].

The same division between quantitative and qualitative methods is illuminating when classifying the few works that deal with outdoor position estimation.

The work of Levitt et al. [30] proposes a number of qualitative constraints that can be used to generate region-based estimates for position and orientation. The work by Thompson et al. extends such qualitative navigation concepts to a complete computational theory of localization [4, 17, 46, 47, 48]. The work is motivated by biological examples, including cues and techniques used by experts to perform geometric reasoning; the objective of estimation is to produce a region containing all possible position estimates. These region-based estimates encode information about the "stability" of geometric inference; the larger the region, the poorer the estimate [43]. Their system produces region-based estimates, called *areas of uncertainty* [42], using correspondences between image and map features to produce pose estimates. In their test of the system, an area of uncertainty of 0.1405 km^2 around the correct pose was generated from a panoramic image (for comparison purposes, 0.1405 km^2 is the area of a square with 375 meters in each side). This result is particularly impressive because position and orientation are estimated from visual information alone.

Work by Stein and Medioni [41], Talluri and Aggarwal [45], and Naval et al. [37] pursue instead a quantitative approach.

Talluri and Aggarwal assume that orientation is known and conduct their search for the best estimate in the space of pose estimates. They use four points in the skyline as the main visual features in a matching procedure. Only the position estimates that match these four points correctly are further investigated, again using information in the skyline. Simulation results indicate an average accuracy of 26 meters for noise-free images, and of 55 meters for images corrupted with Gaussian noise. The result is highly artificial, because it ignores the mistakes that exist in a real skyline detection procedure and the errors in a map, but it indicates that the skyline contains a great deal of terrain information.

A more practical approach is advocated by Stein and Medioni, who use the full skyline. Their estimation algorithm uses line segments that approximate the image skyline. The best estimate is the position that produces the best match between terrain and image skylines (orientation is assumed to be known). Stein and Medioni report good results with a panoramic image (between 300 and 600 meters accuracy). To generate a fast algorithm, they precompute the line segments and subsample the topographic map; for their result, the resolution of the map is 300 meters. There is an important insight in their work: to efficiently search in pose space, it is necessary to precompute as much information as possible.

A different strategy is followed by Naval et al. [37]. They search in the space of correspondences; for each possible match between image and map features, they generate a pose estimate. Several images (32 images) of a single location have been processed by their system, with accuracy results as good as 90 meters, and mean accuracy of 373 meters. Unfortunately, the exhaustive search for correspondences consumes enormous resources: the reported mean processing time for each image is 1 hour and 36 minutes.

3.3. Feature-based and signal-based methods

Another possible classification of pose estimation algorithms focuses on the form of the search procedure [47]. *Feature-based* methods focus on correspondences among features; *signal-based* methods focus on correspondences among dense structures in the image. Thompson et al. and Naval et al. describe feature-based systems that employ natural landmarks (peaks and depressions); Talluri and Aggarwal, and Stein and Medioni describe signal-based systems that employ structures directly obtained from the incoming images (skylines).

4. The position estimation algorithm

Building upon the issues discussed in the previous sections, we now present a new algorithm for position estimation in natural, open environments. The algorithm adopts several ideas from Stein and Medioni: qualitative search is performed using all available skyline information, and pre-compilation of geographic information is used to tame computational effort. We introduce a statistical formulation for the position search problem. Note that we do not use skyline information to estimate attitude, as attitude can be obtained from celestial sensors during planetary missions.

4.1. Skyline matching in a grid representation

The central idea is to represent the space of positions (x, y) by a grid and evaluate all possible elements in this grid. Every position in the grid is evaluated by signal-based matching: the skyline extracted from the images is compared to the skyline rendered using the map. To simplify the presentation, assume that a full 360 degree panorama is available, and the image skyline is extracted from this panorama. Denote the image skyline by m, and the skyline rendered from a position Γ by $s(\Gamma)$. An arbitrary position Γ is evaluated by comparing m with $s(\Gamma)$. Estimates are compared through an evaluation function $e(\Gamma, m)$ that measures the match between $s(\Gamma)$ and m.

The algorithm is signal-based as it uses all information about the image skyline to estimate position (not just linear approximations), without any attempt to interpret the skyline or locate topographic features. The algorithm demonstrates how to take the preliminary efforts of Medioni and Stein, and Talluri and Aggarwal to full realization. This is an important departure from a previous version of the Viper system that used a feature-based algorithm for estimation [11]. The new, signal-based algorithm is more accurate than the previous algorithm because more information is taken into account. Mistakes that inevitably occur in image and map processing are less prone to corrupt the overall procedure.

We emphasize that our approach is to estimate position Γ and leave the determination of orientation to other sensors: celestial sensors in space applications and compasses and inclinometers on Earth (Section 2). Such sensors are quite effective, so there is no need to place an additional burden on the visual estimation system.

The skyline was selected as the main source of information for two reasons. First, the skyline is by far the most salient feature in any panorama; for environments with small mountains, or mountains that are relatively far away, the skyline contains almost all available information. Second, lunar environments are notoriously devoid of any texture, as reported by the Apollo astronauts [16].

4.2. Evaluating estimates

The estimation algorithm depends on the evaluation function $e(\Gamma, m)$. We adopted a statistical interpretation for $e(\Gamma, m)$, viewing it as a measure of posterior probability for the position Γ. Our evaluation combines a measure of prior probability and a measure of sensor quality, corresponding to the "prior" and the "likelihood" employed in Bayesian statistical analysis [12]. As we focus on drop-off problems, we used uniform distributions as prior probabilities to indicate lack of prior knowledge about position.

Consider a skyline m represented as a vector of visual elevations m_i. Each elevation m_i is the angle between the horizontal plane and the skyline; the exact meaning of the term is depicted in Figure 3. We adopted a Gaussian model for the skyline measurements given position:

$$p_\sigma(m|\Gamma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\sum_i (m_i - s_i(\Gamma))^2}{2\sigma^2}\right), \tag{1}$$

where $s_i(\Gamma)$ is the elevation of the skyline $s(\Gamma)$ that corresponds to m_i.

There are practical difficulties in determining a single, sharp value for the standard deviation σ, and the Gaussian assumption may be questionable for real camera measurements. We have chosen to use an interval-valued model to represent imprecision in the evaluation function, in the spirit of robust Statistics research [5]. We consider an interval of possible standard deviations $[\sigma_l, \sigma_u]$, representing both the imprecision on the measurement of σ and the approximate value of the Gaussian assumption. The process used to obtain σ_l and σ_u is presented in Section 5.2.

The construction of an interval-valued evaluation function can be formalized using sets of probability measures [5, 51]. Take $K(\Gamma|m)$ as the set of all probability measures defined by Expression (1) with $\sigma \in [\sigma_l, \sigma_u]$. The minimum value of posterior probability for any estimate Γ is:

$$\underline{p}(\Gamma|m) = \min_{p \in K} p(\Gamma|m) = \min_{\sigma \in [\sigma_l, \sigma_u]} \frac{p_\sigma(m|\Gamma)p(\Gamma)}{p(m)}. \tag{2}$$

The maximum value $\overline{p}(\Gamma|m)$ is obtained maximizing $p_\sigma(\Gamma|m)$.

The computational burden imposed by this interval-valued model is minimal. To compute the minimum probability value, we have:

$$\underline{p}(\Gamma|m) = \left(1 + \sum_{\gamma \neq \Gamma} \frac{p(\gamma) \max_\sigma p_\sigma(m|\gamma)}{p(\Gamma) \min_\sigma p_\sigma(m|\Gamma)}\right)^{-1}.$$

The function $p_\sigma(m|\Gamma)$ is minmized (and maximized) either at σ_l or at σ_u or at $\sigma^2 = \sum_i (m_i - s_i(\Gamma))^2$. The maximum probability value $\overline{p}(\Gamma|m)$ is computed by a similar procedure.

The fact that the prior measure for Γ is a single prior distribution $p(\Gamma)$ poses no restriction to the method, as there are efficient algorithms for exact computation of $\overline{p}(\Gamma|m)$ and $\underline{p}(\Gamma|m)$ even when a set of prior distributions is used [52, page 18]. As uniform distributions are satisfactory representations for drop-off problems, we have not explored interval-valued prior probabilities.

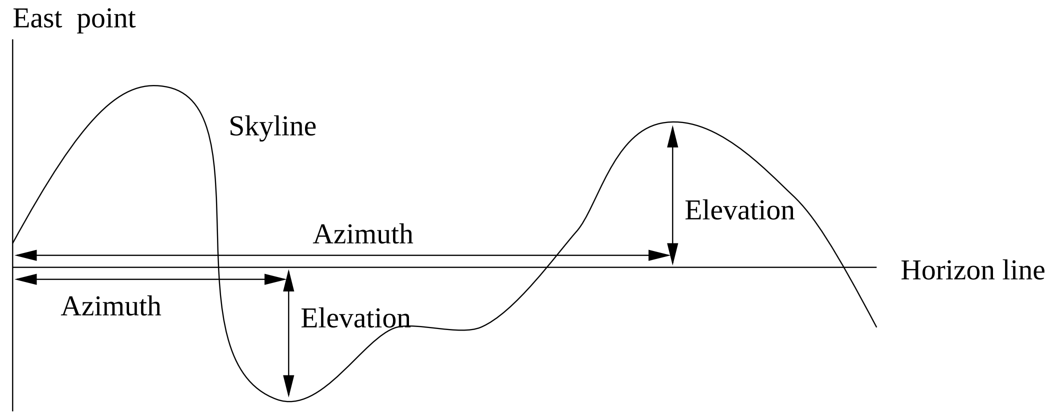


Fig. 3. Representing a skyline.

4.3. Scalar and region-based evaluation

An interval-valued evaluation function $e(\Gamma, m) = [\underline{p}(\Gamma|m), \overline{p}(\Gamma|m)]$ can represent both the relative value of estimates and the quality of the estimates. To compare estimates, one can use the lower value $\underline{p}(\Gamma|m)$ or even a combination of lower and upper values [51]. The quality of an estimate is the width of the interval $e(\Gamma, m)$: $\overline{p}(\Gamma|m) - \underline{p}(\Gamma|m)$.

An interval-valued evaluation function leads naturally to region-based measures of estimate quality. For example, consider an estimate Γ^* that maximizes $\underline{p}(\Gamma|m)$. The estimate Γ^* can be associated with a region $R(\Gamma^*)$, containing all estimates that are "essentially" equivalent to Γ^*:

$$R(\Gamma^*) = \left\{ \Gamma : \overline{p}(\Gamma|m) \geq \underline{p}(\Gamma^*|m) \right\}.$$

The larger $R(\Gamma^*)$, the less sharp is the "best" estimate Γ^*.

The use of an interval-valued function $e(\Gamma, m)$ smoothly integrates a region-based solution into the estimation problem, adopting the best ideas of the work on areas of uncertainty by Thompson et al. (Section 3). This formulation captures all essential aspects of the estimation problem: both the statistical variation in the measurements (captured by the standard deviation), and the effect of model variations (captured by the width of the evaluation intervals).

4.4. The estimation algorithm

The estimation algorithm can be summarized as follows. The skyline is obtained from a sequence of images. The image skyline m and the rendered skylines $s(\Gamma)$ are matched: for any position Γ, an evaluation interval $e(\Gamma, m) = [\underline{p}(\Gamma|m), \overline{p}(\Gamma|m)]$ is produced following Expression (2). The estimate with highest lower posterior probability (maximum of $\underline{p}(\Gamma|m)$) is selected as the "best" estimate. This "best" estimate Γ^* is selected as the position that is most likely correct even with the worst possible modeling errors. The region $R(\Gamma^*)$ is then calculated; any estimate Γ such that $\overline{p}(\Gamma|m) \geq \underline{p}(\Gamma^*|m)$ is potentially equivalent to Γ^* and is included in $R(\Gamma^*)$. In the end, the operator is able to visualize the estimate Γ^* and the region $R(\Gamma^*)$ (and to judge the quality of the estimate Γ^*).

A simple example can clarify the procedure. Suppose we are trying to decide among three possible positions Γ_1, Γ_2 and Γ_3. Suppose that the four skylines in Figure 4 are obtained: the first is extracted from a panoramic image and the other three are rendered from a map from each one of the three possible positions. Consider two situations:

- Suppose that $e(\Gamma_1, m) = [0.1, 0.2]$, $e(\Gamma_2, m) = [0.3, 0.5]$, $e(\Gamma_3, m) = [0.15, 0.25]$. In this case Γ_2 is the "best" estimate, and $R(\Gamma_2) = \{\Gamma_2\}$, so the choice of Γ_2 is quite robust against modeling errors.
- Suppose that $e(\Gamma_1, m) = [0.1, 0.2]$, $e(\Gamma_2, m) = [0.3, 0.5]$, $e(\Gamma_3, m) = [0.25, 0.45]$. In this case Γ_2 is still the "best" estimate, but $R(\Gamma_2) = \{\Gamma_2, \Gamma_3\}$, so the choice of Γ_2 does not have the same sharpness.

5. The Viper system

The Viper system implements the estimation algorithm described previously in the context of long-duration teleoperation missions. Figure 5 shows the basic elements of the Viper system. The user observes a stream of images and a topographic map, and can request the automatic construction of a panoramic image, the detection of the skyline in the panorama, and the generation of position estimates.

The first step in the estimation procedure is the construction of a panorama from the raw images in the video window. The user selects individual images, which are then automatically stitched together into a single panoramic image. The second step is the detection of the skyline in the panorama. The detected skyline is superimposed to the panorama; note the skyline contour in the panorama in Figure 5. The last step is the search

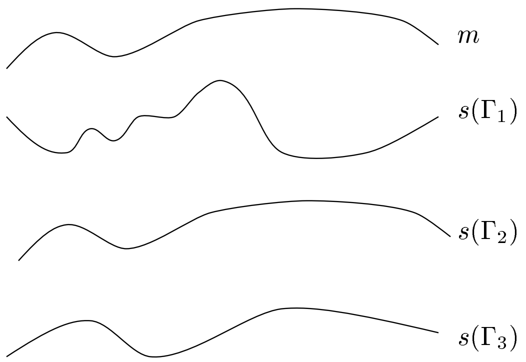


Fig. 4. Example for estimation algorithm.

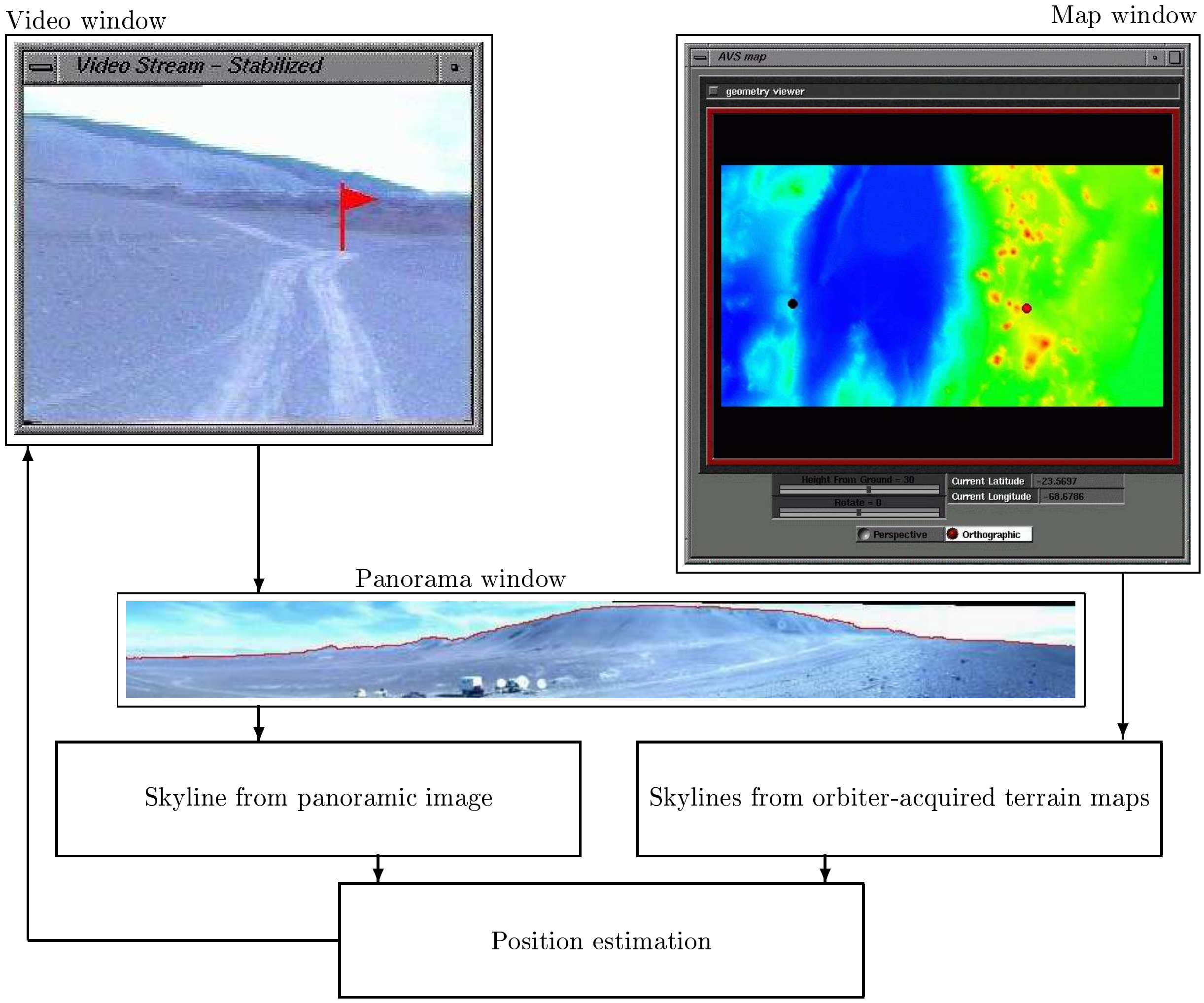


Fig. 5. The Viper system.

for the best position estimate. The best estimate is then displayed to the user and marked in the map as a black dot.

5.1. Overview of the user interface

The interface displays three independent windows to the user. The *map window* displays topographic maps. The *video window* displays a stream of images. The *panorama window* controls the formation of a panorama, the detection of the skyline, and the presentation of position estimates.

In the map window, the user selects the size, color and projection of the topographic map. Orthographic and perspective projections are available. In orthographic projection, the map is seen from above; in perspective projection, the map is rendered as it would be seen from a point on the ground. Figure 5 shows a topographic map of the Atacama desert in Chile rendered in orthographic projection. The map window is designed after the *maptool* program distributed by the University of Utah Computer Vision Group. The dark dot in the map indicates the rover position, and the light dot indicates the user-defined goal (the flag in the video window is in the direction of the goal).

The video window displays the video signal, either from a camera or from a recorded tape. The user can select individual images from this window through mouse clicks.

The panorama window controls the position estimation process. Every image selected by the user is displayed in the panorama window. The user can then accept or reject the image; if the image is accepted, the system automatically stitches the image onto the current panorama (or starts a new panorama). After a complete panorama is available in the panorama window, the user can activate the skyline detection algorithm. Once the skyline is detected, the user can ask for a position estimate. Position estimates are displayed both in the panorama and in the map windows.

5.2. System components

The Viper system is built as a series of modules attached to the Advanced Visualization System (AVS), a commercial package for manipulation of images and three-dimensional graphics. All modules are coded in the C language and run on a Silicon Graphics Impact R10000 workstation (195MHz processor) with a video board, 128MBytes of main memory and 4GBytes of disk.

Experiments with the Viper system in Pittsburgh and in the Atacama desert used data collected by a customized sensor platform. During outdoor data collection, images were taped by a video cassete recorder, with signal coming from a camera on top of a rotatable platform. The video cassete was then attached to the workstation video board so that images could be processed.

The sensor platform features a camera, an electronic compass, two digital inclinometers, and a differential GPS receiver. The sensor platform is attached to a tripod. The camera is a Sony XC-003 color camera, calibrated with the `detecproj` program [39]. A KVH Autocomp 1000/Azimuth 314AC digital heading sensor measures orientation and a Lucas Angle Star protractor provides a dual inclinometer system; these attitude sensors are used to emulate measurements from celestial sensors in a planetary mission. A Trimble SV6-CM3 GPS receiver and a Garmin GPSII GPS receiver provide ground-truth data, used to check the results of the algorithm.

Figure 6 shows two pictures of the sensor platform in the Atacama desert. The second picture shows the sensor platform attached to the Nomad robot [54].

A typical image acquisition sequence with the platform follows a sequence of steps: (1) the camera is leveled using the inclinometers, (2) the compass is calibrated and camera orientation is established, (3) the ground-truth for position is established using the GPS receiver, and (4) the camera is rotated around its vertical axis. The result of this procedure is a sequence of images tagged with six-dimensional ground-truth: both the camera position and orientation are known.

Note that the calibration of platform components is done by hand; a planetary mission would instead use rigidly calibrated celestial sensors [53].

The presence of metallic objects around the compass causes major disturbances. To obtain reliable, accurate orientation measurements, it is necessary to carefully calibrate the compass at least ten meters away from any sizable metallic object. In the Atacama desert, where conditions were quite harsh, compass measurements had an accuracy of roughly ± 4 degrees after calibration.

5.3. Panorama generation and skyline detection

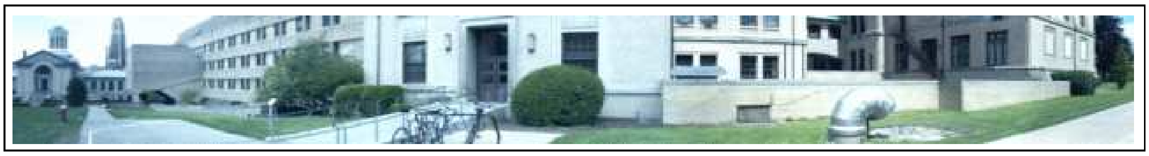
To construct a panorama, images selected from the video window are first projected into a cylindrical surface, and then the motion among images is estimated. Once the motion is obtained, images can be transformed into a common spherical coordinate system and stitched together.

The panorama generation algorithm begins with a coarse estimate of translational motion, obtained with the Kuglin/Hines method [28, 44]. The method uses the phase difference of the 2D Fourier transform of two images to estimate the translational displacement between the images. This initial estimate of translational motion between images is the seed for a Levenberg-Marquardt optimization procedure that extracts the rigid motion between two images.

Every panorama has a center, defined as the center of the first image used to build the panorama. As the sensor platform is leveled and the camera is calibrated, the horizontal line that goes through the center of the panorama is the horizon line. Any point in a skyline is defined by two angles, the *azimuth* and the *elevation* angles (Figure 3). The azimuth measures the horizontal

Sensor platform

Sensor platform in the Nomad robot



Panorama used to calibrate the sensor platform

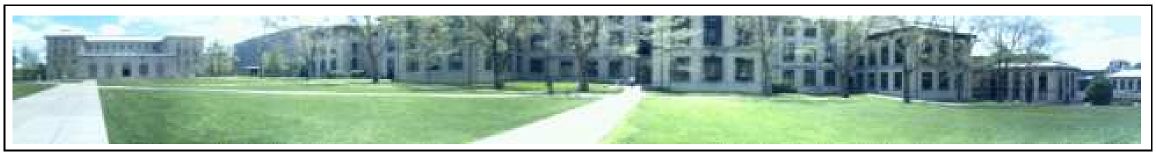


Fig. 6. The sensor platform.

angular displacement from the East point to the point in question. The elevation measures the vertical angular displacement from the horizon line to the point in question. Values of azimuth and elevation are obtained directly through the camera calibration parameters; for example, the elevation value of pixel p is $\arctan(\Delta p / f)$, where Δp is the vertical distance in pixels from p to the horizon line and f is the focal length of the camera in pixels.

Once a panorama is constructed, the user can ask the system to detect the skyline. The skyline detection algorithm analyzes every column of the panorama separately. The smoothed intensity gradient [19] is calculated from top to bottom. The system takes the first pixel with gradient higher than a threshold to be a skyline element. Two user-defined parameters control this procedure: (1) the size of the template used for Gaussian smoothing; (2) the threshold that determines when a point belongs to the skyline. In practice, relatively large errors in the skyline (up to ten percent of the skyline) did not affect the results we report in Section 6. The key point is that, as long as there is enough topographic relief in the environment, skyline errors can be overcome. In the Viper system, the user has the option of quickly erasing parts of the panorama with mouse movements: The skyline detection algorithm is best seen as an aid to human operators — instead of requiring the operator to draw the skyline, which can take around 30 seconds of a trained person, the system produces a best guess and lets the operator fix it if necessary.

The critical parameter in skyline detection is the intensity threshold. The skyline detection algorithm does *not* have any high-level concept about structures in the images. The algorithm does not attempt to avoid urban structures as it is geared towards planetary missions. The thresholding operation can also be fooled by the presence of clouds or intense sunlight; consequently, any measured panorama is subject to errors. We have tried more sophisticated options for skyline detection, such as adaptive thresholding with respect to average intensity, and eliminating clouds before thresholding. Because none of these algorithms is perfect, it is always necessary to let the user overrule the system; we selected the simplest and fastest implementation for skyline detection. The descriptions of lunar missions and the images we obtained from these missions indicate that a simple thresholding operation is effective in these situations. The skyline detection algorithm takes one to two seconds to process a panorama (typically containing 300 by 3500 pixels); the panorama for the Wasatch mountains described in Section 6 is exceptionally large (370 by 6000 pixels) and requires four to five seconds for processing. All timing values refer to the time elapsed from the start of an operation to the presentation of the result to the user; variations are due to memory swapping and processor load.

5.4. Skyline detection model

The position algorithm depends on the measurement model summarized by Expression (1). To obtain this model, the sensor platform was taken to open areas and its measurements were compared with the measurements taken by a theodolite. Figure 6 shows a typical panorama used in these experiments. The elevations of 17 points in this panorama were calculated using data from the sensor platform. The elevations of the same 17 points were measured with the theodolite, with accuracy of thirty seconds of arc.

Even though the measurements of angular displacements obtained with the camera are discrete, a Gaussian model was chosen to approximate the characteristics of the sensor platform. A confidence interval estimate for the standard deviation σ was produced based on the classical estimation formula: $\left[\sqrt{(n-1)s^2/\chi^2_{n-1,2.5\%}}, \sqrt{(n-1)s^2/\chi^2_{n-1,97.5\%}}\right]$, where n is the number of theodolite-calibrated points (17 points) and $\chi^2_{n,\alpha}$ is given by a χ-square distribution with n degrees of freedom at level α [51]. This range of standard deviations was used to define lower and upper likelihoods (Section 4.2).

5.5. Position estimation

A grid of position estimates is the basic data structure employed by the estimation algorithm. The algorithm receives the image skyline and compares the image skyline with skylines rendered from each possible position estimate. In the Viper system, the grid resolution is defined by the resolution of the underlying terrain map (Section 6). For example, the grid for estimation in the vicinity of Pittsburgh is a uniform grid with a 30 meter spacing between lines; the grid for estimation in the Atacama desert is a grid with an 85 meter spacing between vertical lines and a 93 meter spacing between horizontal lines.

The estimation algorithm assumes that orientation is measured accurately, but orientation measurements are difficult to obtain with good accuracy on Earth due to compass errors (Section 5.2). To remedy this problem, the estimation procedure was enlarged to include the orientation ϕ (yaw angle). Instead of two-dimensional grids for position (x, y), three-dimensional grids for (x, y, ϕ) are generated. For a compass measurement $\phi_o \pm \beta$, orientation values from $\phi_o - \beta$ to $\phi_o + \beta$ are considered in the search (in one-degree increments). The constant β was fixed at 2 degrees in Pittsburgh and at 4 degrees in the Atacama desert.

The most time-consuming operation in the estimation algorithm is the comparison between the skyline detected in the panorama and the rendered skylines. The key idea to speedup the estimation algorithm is to precompute and store the rendered skylines. Skylines are rendered from the topographic map in batch mode, before any images are processed. A ray-tracing algorithm is used for rendering skylines, based on Stein and Medioni's algorithm for construction of super-segments [41].

Each rendered skyline is stored as a one-dimensional vector. a skyline vector contains elevation values; azimuth angles are given by the indices of these values. A vector with 180 elevation values represents a skyline. To save memory, the elevation angles are encoded into 128 levels. Each skyline vector occupies 184 bytes of memory: 180 bytes to represent the azimuth angles and 4 bytes to represent the encoding for elevation angles.

All rendered skylines for a map are stored in a single data structure (the AVS-specific `field` structure). The memory occupied by the rendered skylines is a function of the geographic area covered by the skylines. For example, tests with the Pittsburgh map used a block of rendered skylines covering approximately 154 km^2 (in a 30 meter grid). Approximately 172,000 rendered skylines were produced for this area; the encoded skyline vectors occupy approximately 31MBytes. Another example is the map that covers the Wasatch mountains area in Utah. Approximately 331,000 rendered skylines were produced to cover this area; the skylines occupy approximately 61MBytes. The pre-processing scheme is effective to speedup the algorithm. For example, the position estimation algorithm processes the complete topographic map of Pittsburgh, which is a rectangular array with 468×367 points, in approximately 15 seconds.

Once a position estimate is generated, the system:

- Marks the "best" position estimate Γ^* in the topographic map with a black dot (Figure 5).
- Writes the position estimate in the panorama.
- Displays the region-based measure of estimation quality $R(\Gamma^*)$.

6. Experiments with the Viper system

The Viper system was tested in a variety of environments and terrain types. This section reviews the performance of the system and discusses the insights and lessons learned from the experiments.

6.1. Topographic maps

The system was tested with digital topographic maps, following either the 7.5 minute USGS format or the 1 degree USGS format [38]. Both formats specify rectangular arrays of altitude values. A 7.5 minute map is a uniformly spaced grid with 30 meter resolution. A 1 degree map is a uniformly spaced grid with 3 minute resolution.

Four sources provided the maps for regions of the United States, the Atacama desert in Chile, and the Apollo 17 landing site on the Moon. Maps for areas in Pennsylvania and California were purchased directly from the U. S. Geological Survey. The map for the Wasatch mountains in Utah was produced and distributed by the University of Utah Computer Vision Group. The map for the Atacama desert was purchased by the Lunar Rover group for their tests with the Nomad robot [54]. The map of the Apollo 17 landing site was produced from the Apollo 17 topophotomap 43D1S1(50) [36] by Andrew Johnston in the Smithsonian Institute.

6.2. *Tests in Pittsburgh*

The Viper system was initially tested in Pittsburgh. Figure 7 illustrates these tests. The topographic map covers approximately 154 km^2.

The panoramas in Figure 7 are typical urban panoramas. Portions of the skyline were deleted manually to avoid urban structures. Note that even with a substantial portion of the skyline deleted, estimation is successful. For the REC (Robotics Engineering Consortium) panorama, the best estimate is off by 108 meters. The map shows the REC location as a black dot in the upper left region of the map; the region covered by this dot also contains the estimated position. For the Island panorama, the best estimate is off by 84 meters. For the Waterworks mall panorama, the best estimate is off by 153 meters.

The Pittsburgh panoramas contain relatively small, round mountains, but these mountains are sufficient to discriminate among estimates. The most effective way to appreciate the estimation results is to simultaneously observe the scalar measure and the region-based measure associated with the best estimate obtained with the REC scenery. The estimation results presented on the top of Figure 7 show details of the region-based (left) and scalar (right) measures for the REC panorama. Looking at the scalar measure (right picture), it is easy to verify where the best position estimates are, and how they compare to each other in terms of lower probability values, but it is difficult to grasp whether the best estimate is robust or not. The scalar measure is roughly divided in three areas: very low probability (light), low probability (darker) and high probability (blobs with light shades). For the region-based measure, the lighter estimates are all equivalent to the best estimate, and the darker estimates are all worse than the best estimate. The region-based measure clearly indicates that the best estimates are in fact grouped in a rather small region, and that these estimates are significantly better than all other estimates.

6.3. *Apollo 17 and Wasatch mountains*

Figure 8 illustrates panoramas for the Apollo 17 landing site on the Moon (best position estimate is off by 180 meters) and for the Wasatch mountains in Utah (best position estimate is off by 128 meters).

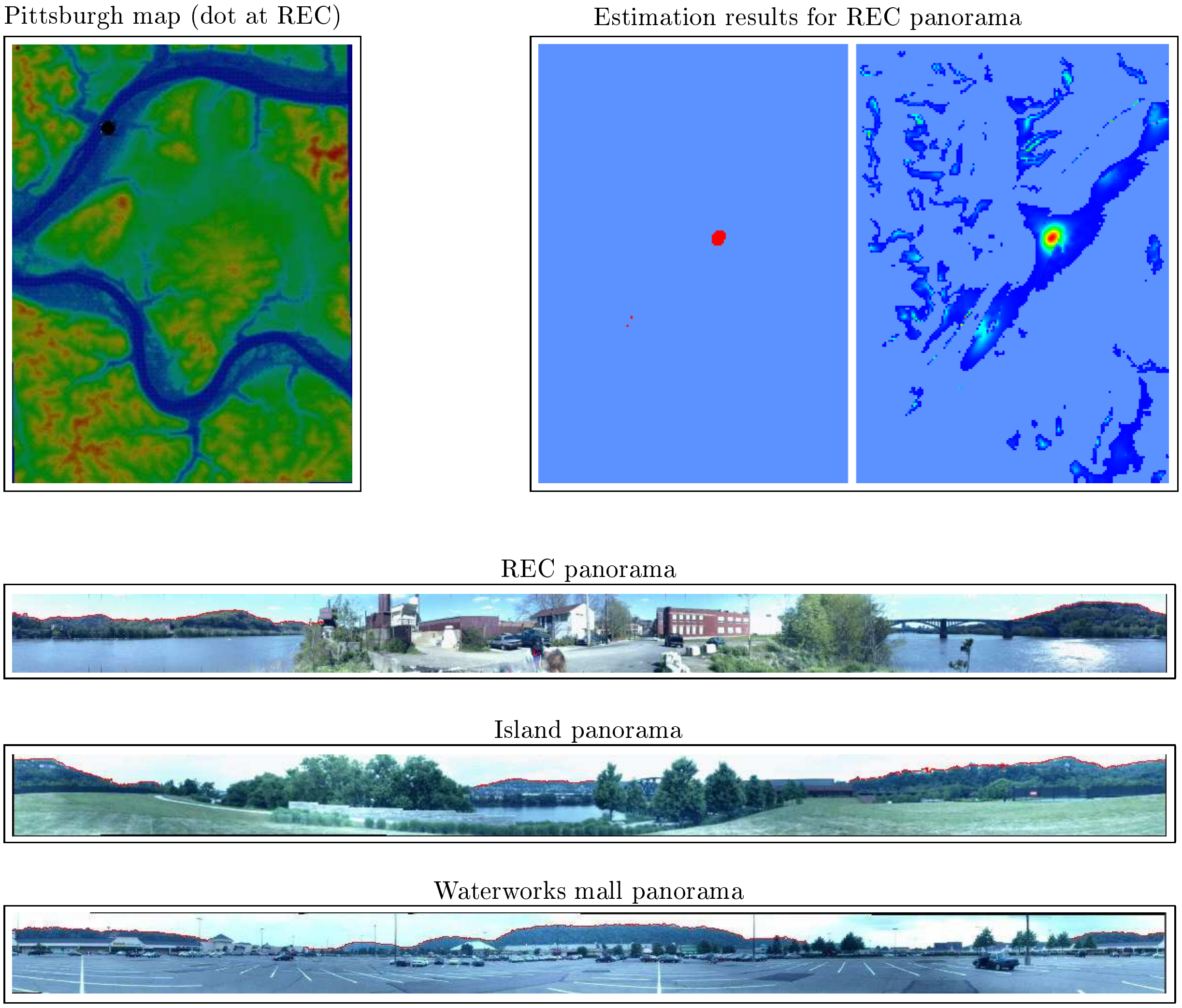


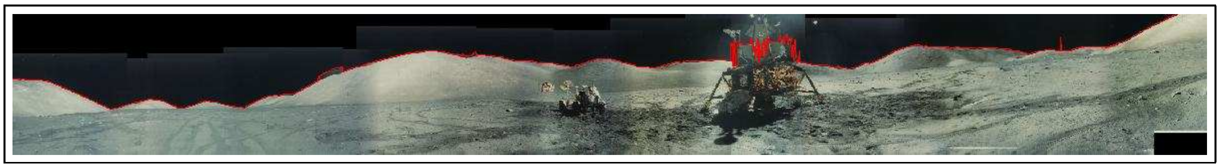
Fig. 7. Results with imagery from Pittsburgh (Pennsylvania).

The Wasatch mountains panorama illustrates a situation in which the estimation quality is exceptional. The estimation algorithm produced an extremely small region-based estimate, indicating that the best estimate is in fact much superior to all other positions.

6.4. The Atacama Desert Trek

In January 1997, the Viper system was selected to participate in the Atacama Desert Trek [54], a large robotics project developed by the Lunar Rover group at the Robotics Institute. The Nomad robot (top of Figure 6) traversed 220 kilometers in the Atacama desert; the robot was remotely

Apollo 17 panorama



Wasatch mountains panorama

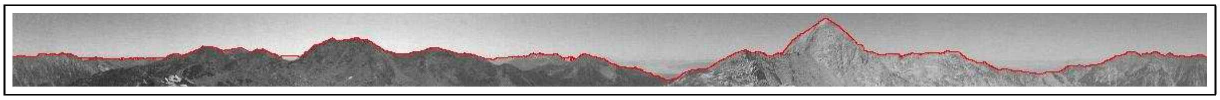


Fig. 8. Results with imagery from the Apollo 17 landing site and from the Wasatch mountains (Utah).

driven from a variety of locations, including the Carnegie Science Center and the NASA Ames Research Center. The Viper system was tested in the Atacama desert from July 8th to July 14th.

Seventeen panoramas were collected in widely different areas, some featuring few mountains and small elevations, others featuring the Andes and large active volcanos. The system display in Figure 5 shows a topographic map of the Atacama desert; the black dot in the map is the location of the operations center during the Atacama Desert Trek. Figure 9 shows three panoramas from the desert; the first panorama contains an image of the trucks and antennas in the operations center. Most of the images and data were collected in the Salar de Atacama, a vast plane area in the middle of the map.

The first panorama in Figure 9 was taken July 8th; the position estimate was off by 186 meters. Other panoramas taken during the Atacama Desert Trek produced estimates with similar accuracy (from 200 to 400 meters). Table 1 contains a summary of the position estimation results in the Atacama Desert Trek. The second and third panoramas in Figure 9 produced quite disappointing results; these panoramas were taken July 10th, in a very windy evening (calibration was almost impossible). To study how the system could be used in such scenarios, we extended the estimation algorithm to combine skylines in nearby areas. We obtained a fast updating algorithm by using the scalar measure generated by the first skyline as the prior information for processing the sec-

Table 1. Results in the Atacama Desert Trek.

Panorama	Description	Accuracy of estimate (meters)
First, July 8th	Test of system set-up	
Second, July 8th	System run	186
Three panoramas, July 9th	Tests of calibration algorithm	
First, July 10th	System run	400
Second, July 10th	System run	327
Third, July 10th	System run	564
July 10th	Combined second, third panoramas	194
First, July 11st	System run	350
Second, July 11st	System run	250
Two panoramas, July 12nd	Tests with equipment	
July 12nd	System run	300
Two panoramas, July 13rd	Test in the Andes area (map not available)	
First, July 14th	Test of system in the Nomad robot	
Second, July 14th	System run in the Nomad robot	280

ond skyline. The algorithm amounts to using the lower posterior probability values from the first panorama as the prior probability for the second panorama. This algorithm was tested with the images taken July 10th. The accuracy of the estimates obtained with each of the second and third panoramas separately is rather poor: The best estimate for the second panorama is off by 327 meters; the best estimate for the third panorama is off by 564 meters. But the estimate produced by both panoramas is off by 194 meters, a significant improvement that indicates the benefits of combining panoramas in a long duration mission.

More details about the Atacama Desert Trek, including all panoramas acquired in the desert, can be viewed in the world-wide-web at `http://www.cs.cmu.edu/~viper`. All panoramas mentioned in Table 1 can be downloaded from the project web site.

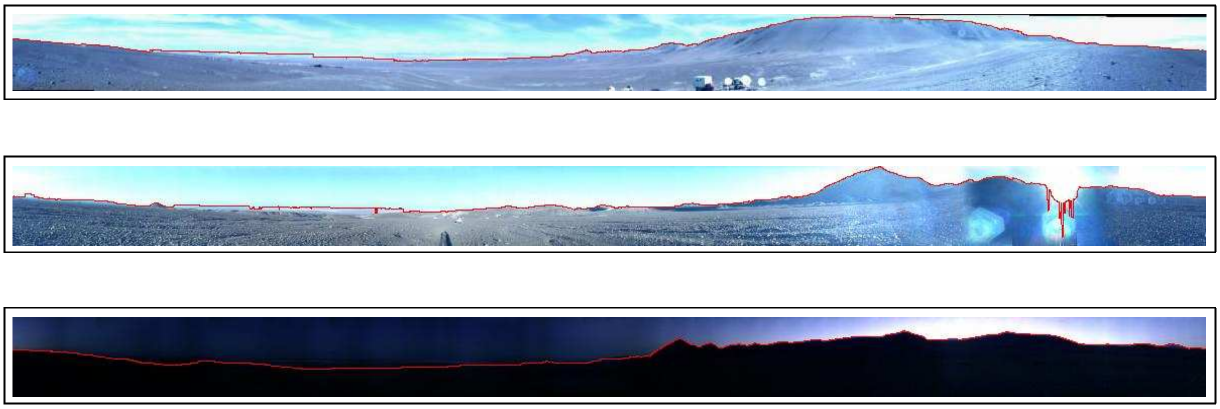


Fig. 9. Results with imagery from the Atacama desert.

7. Discussion

The Viper system demonstrates the feasibility of accurate and fast position estimation in natural, open environments. The system was tested in five completely different types of terrain, and results indicate that the system can determine position estimates with accuracy from 84 to 200 meters (with 7.5 minute maps) and accuracy from 186 to 564 meters (with 1 degree maps). The collection of real data used to test the system greatly exceeds previous work in outdoor position estimation. The accuracy depends on the map resolution because the estimation grids have the same resolution as the map. Overall, the system determines position with accuracy from 2.5 to 6.5 map cells.

The position estimation algorithm in the Viper system handles grid structures that are much larger than related research based on occupancy grids. The size of the areas handled by the algorithm and the techniques for storage and retrieval of rendered skylines advance the state-of-the-art in mobile robot localization.

The Viper system provides effective region-based measures of estimation quality for the robot operator. Region-based and scalar measures provide guidance in determining whether or not more imagery should be collected to determine position confidently. The algorithm is unique in its simultaneous treatment of measurement errors and estimation quality; it combines the best characteristics of previous approaches in a principled, short derivation.

Future work must investigate methods that combine feature-based methods with the dense matching algorithm used here. Other panorama features and topographic formations could be used to evaluate poses; this strategy would produce a mix of feature-based and signal-based approaches.

Acknowledgements

The Nomad team substantially contributed to the tests with the Viper system; thanks particularly to Mark Maimone for insisting that the Viper system should be used in the Atacama desert. Andrew Johnston digitized the map of the Apollo 17 landing site and helped select the lunar imagery. Jim Moody and Chuck Miller provided invaluable help with hardware and software issues. Luc Robert shared his camera calibration code, and David LaRose generously shared his expertise in image registration.

References

1. W. A. Aviles, T. W. Hughes, H. R. Everett, A. Y. Umeda, S. W. Martin, A. H. Koyamatsu, M. R. Solorzano, R. T. Laird, and S. P. McArthur. Issues in mobile robotics: The unmanned ground vehicle program teleoperated vehicle (TOV). *Mobile Robots SPIE*, V:587–597, November 1990.
2. N. Ayache. *Artificial vision for mobile robots: stereo vision and multisensory perception.* Artificial intelligence. MIT Press, Cambridge, Mass., 1991.
3. R. T. Azuma. A survey of augmented reality. *Presence*, 6(4):355–385, August 1997.
4. B. K. H. Bennett. *A Problem-Solving Approach to the Localization Problem.* PhD thesis, University of Minnesota, February 1992.
5. J. O. Berger. Robust Bayesian analysis: Sensitivity to the prior. *Journal of Statistical Planning and Inference*, 25:303–328, 1990.
6. J. Borenstein and L. Feng. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation*, 12(6):869–880, December 1996.
7. W. Burgard, D. Fox, D. Hennig, and T. Schnmidt. Position tracking with position probability grids. *First Euromicro Workshop on Advanced Mobile Robots*, 1996.
8. A. R. Cassandra, L. P. Kaebling, and J. A. Kuriien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.
9. I. J. Cox and G. T. Wilfong. *Autonomous Robot Vehicles.* Springer-Verlag, 1990.
10. F. Cozman and E. Krotkov. Robot localization using a computer vision sextant. *International Conference on Robotics and Automation*, pages 106–111, May 1995.
11. F. Cozman and E. Krotkov. Automatic mountain detection and pose estimation for teleoperation of lu-

nar rovers. *Proc. of the International Conference on Robotics and Automation*, pages 2452–2457, 1997.

12. M. DeGroot. *Optimal Statistical Decisions.* McGraw-Hill, New York, 1970.
13. A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265, June 1987.
14. A. Elfes. Dynamic control of robot perception using stochastic spatial models. *International Workshop on Information Processing in Mobile Robots*, March 1991.
15. I. A. Getting. The global positioning system. *IEEE Spectrum*, pages 36–47, December 1993.
16. G. Heiken, D. Vaniman, and B. French, editors. *Lunar Sourcebook.* Cambridge University Press, 1991.
17. M. R. Heinrichs, K. Smith, R. L. Pick Jr., Bonnie H. Bennett, and W. B. Thompson. Strategies for localization. In *DARPA Image Understanding Workshop*, pages 321–325, Scottsdale, AZ, January 1992.
18. R. R. Hobbs. *Marine Navigation.* Naval Institute Press, Maryland, 1974.
19. B. K. P. Horn. *Robot Vision.* The MIT Press, Cambridge, Massachusetts, 1986.
20. S. S. Iyengar and A. Elfes. *Autonomous Mobile Robots.* IEEE Computer Society Press, 1991.
21. S. Koenig and R. G. Simmons. A robot navigation architecture based on partially observable Markov decision process models. In D. Kotenkamp, R. P. Bonasso, and R. Murphy, editors, *Artificial Intelligence based Mobile Robots: Case Studies of Successful Robot Systems.* The MIT Press, 1997.
22. D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. *XII National Conference on Artificial Intelligence*, 2:979–984, 1994.
23. G. Kress and H. Almaula. Sensorimotor requirements for teleoperation. Technical Report R-6279, Corporate Technology Center, FMC Corporation, Santa Clara, CA, December 1988.
24. D. J. Kriegman, E. Triendl, and T. O. Binford. Stereo vision and navigation in buildings for mobile robots. *IEEE Transactions on Robotics and Automation*, 5(6), December 1989.
25. E. Krotkov. Mobile robot localization using a single image. In *IEEE Robotics and Automation Conference*, pages 978–983, Scottsdale, AZ, May 1989. IEEE.
26. E. Krotkov, M. Hebert, and R. Simmons. Stereo perception and dead reckoning for a prototype lunar rover. *Autonomous Robots*, 2:313–331, 1995.
27. E. Krotkov, R. Simmons, F. Cozman, and S. Koenig. Safeguarded teleoperation for lunar rovers: From human factors to field trials. *Proc. IEEE Workshop on Planetary Rover Technology and Systems*, April 1996.
28. C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. *IEEE International Conference on Cybernetics and Society*, pages 163–165, September 1975.
29. M. M. Kuritsky and M. S. Goldstein. Inertial navigation. In I. J. Cox and G. T. Wilfong, editors, *Autonomous Robot Vehicles*, pages 96–121. Springer-Verlag, 1990.

30. T. Levitt, D. Lawton, D. Chelberg, and P. Nelson. Qualitative Navigation. In *DARPA Image Understanding Workshop*, 1987.

31. T. C. Lyon. *Practical Air Navigation.* Jeppesen Sanderson Inc., 1972.

32. A. C. McDonald. *Robot technology: theory, design, and applications.* Prentice-Hall, Englewood Cliffs, 1986.

33. D. E. McGovern. Experiences and results in teleoperation of land vehicles. Technical Report SAND90-0299, Sandia National Laboratories, Albuquerque, NM, April 1990.

34. H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, Summer 1988.

35. H. P. Moravec. Robot spatial perception by stereoscopic vision and 3D evidence grids. Technical Report CMU-RI-TR-96-34, Robotics Instititue, Carnegie Mellon University, Pittsburgh, Pennsylvania, September 1996.

36. NASA. Apollo 17 landing area topophotomap. Sheet 43D1S1(50), 1972.

37. P. C. Naval Jr., M. Mukunoki, M. Minoh, and K. Ikeda. Estimating camera position and orientation from geographical map and mountain image. *38th Research Meeting of the Pattern Sensing Group, Society of Instrument and Control Engineers*, pages 9–16, 1997.

38. Branch of Technical Standards and Product Development. *Data Users Guides.* U. S. Geological Survey (USGS), Reston, Virginia, 1993.

39. L. Robert. Camera calibration without feature extraction. *International Conference on Pattern Recognition*, 1994.

40. B. Schiele and J. L. Crowley. A comparison of position estimation techniques using occupancy grids. *IEEE International Conference on Robotics and Automation*, pages 1628–1634, 1994.

41. F. Stein and G. Medioni. Map-based localization using the panoramic horizon. *IEEE Transactions on Robotics and Automation*, 11(6):892–896, December 1995.

42. K. T. Sutherland. *The Stability of Geometric Inference in Location Determination.* PhD thesis, Department of Computer Science, University of Utah, July 1994.

43. K. T. Sutherland. Ordering landmarks in a view. *Image Understanding Workshop*, pages 1101–1105, 1995.

44. R. Szeliski. Image mosaicing for tele-reality applications. Technical Report CRL94/2, DEC Cambridge Research Lab, May 1994.

45. R. Talluri and J. Aggarwal. Position estimation for an autonomous mobile robot in an outdoor environment. *IEEE Trans. Robotics and Automation*, 8(5):573–584, October 1992.

46. W. Thompson, H. Pick, B. Bennett, M. Heinrichs, S. Savitt, and K. Smith. Map-Based Localization: The "Drop-Off" Problem. In *DARPA Image Understanding Workshop*, pages 706–719, Pittsburgh, September 1990.

47. W. B. Thompson, T. C. Henderson, T. L. Colvin, L. B. Dick, and C. M. Valiquette. Vision-based localization. In *DARPA Image Understanding Workshop*, pages 491–498, Maryland, April 1993.

48. W. B. Thompson, C. M. Valiquette, B. H. Bennet, and K. T. Sutherland. Geometric reasoning for map-based localization. Technical Report UUCS-96-006, Department of Computer Science, University of Utah, May 1996.

49. S. Thrun and A. Bucken. Integrating grid-based and topological maps for mobile robot navigation. *XIII National Conference on Articial Intelligence*, August 1996.

50. D. Torrieri. Statistical theory of passive location systems. *IEEE Trans. Aerospace and Electronic Systems*, AES-20(2):183–198, 1984.

51. P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.

52. P. Walley. Measures of uncertainty in expert systems. *Artificial Intelligence*, 83:1–58, 1996.

53. J. R. Wertz. *Spacecraft Attitude Determination and Control*, volume 73 of *Astrophysics and space science library*. Reidel, Boston, 1978.

54. W. Whittaker, D. Bapna, M. W. Maimone, and E. Rollins. Atacama desert trek: A planetary analog field experiment. *International Symposium on Artificial Intelligence, Robotics and Automation for Space*, July 1997.

55. B. Yamauchi and P. Langley. Place recognition in dynamic environments. *Journal of Robotic Systems*, 14(2):107–120, February 1997.