

Long-Range Rover Localization by Matching LIDAR Scans to Orbital Elevation Maps

Patrick J.F. Carle, Paul T. Furgale, and Timothy D. Barfoot

Institute for Aerospace Studies, University of Toronto, Toronto, Ontario M3H 5T6, Canada

e-mail: pat.carle@utoronto.ca, paul.furgale@utoronto.ca, tim.barfoot@utoronto.ca

Received 23 June 2009; accepted 6 February 2010

Current rover localization techniques such as visual odometry have proven to be very effective on short- to medium-length traverses (e.g., up to a few kilometers). This paper deals with the problem of long-range rover localization (e.g., 10 km and up) by developing an algorithm named MOGA (Multi-frame Odometry-compensated Global Alignment). This algorithm is designed to globally localize a rover by matching features detected from a three-dimensional (3D) orbital elevation map to features from rover-based, 3D LIDAR scans. The accuracy and efficiency of MOGA are enhanced with visual odometry and inclinometer/sun-sensor orientation measurements. The methodology was tested with real data, including 37 LIDAR scans of terrain from a Mars-Moon analog site on Devon Island, Nunavut. When a scan contained a sufficient number of good topographic features, localization produced position errors of no more than 100 m, of which most were less than 50 m and some even as low as a few meters. Results were compared to and shown to outperform VIPER, a competing global localization algorithm that was given the same initial conditions as MOGA. On a 10-km traverse, MOGA's localization estimates were shown to significantly outperform visual odometry estimates. This paper shows how the developed algorithm can be used to accurately and autonomously localize a rover over long-range traverses. © 2010 Wiley Periodicals, Inc.

1. INTRODUCTION

The ongoing Mars Exploration Rover (MER) missions have proven to be historic landmarks in space exploration. However, they are also humbling reminders of the challenges ahead. For example, the MER Opportunity has operated on Mars for more than 5 years now but has driven only a total of about 20 km due to mechanical/energy limitations and a lack of autonomy (Li, Arvidson, Di, Golombek, Guinn, et al., 2007). An important goal for future generations of rovers will be to overcome these deficiencies to allow them to explore sites hundreds of kilometers away from their landers (Behar, Matthews, Carsey, & Jones, 2004). Rovers will consequently require an autonomous long-range localization system to aid them in their journey.

Currently, a rover employs a variety of techniques to determine its pose at any given time. The MERs were first localized with radio tracking (Guinn, 2001), with descent trajectory modeling, and by comparing orbital to ground camera imagery (Li et al., 2007). Upon leaving their landers, localization has been accomplished primarily with dead-reckoning techniques such as wheel odometry, visual odometry (VO), and local bundle adjustment (BA). Wheel odometry is not computationally intensive but is highly vulnerable to sensor noise and mechanical disturbances (e.g., wheel slippage) (Li, Squyres, & Arvidson, 2005). Computer vision techniques, such as VO and BA, complement wheel odometry when needed.

VO is automated and can work in real time but is computationally very demanding. It has yielded impressive results in the past with error as low as 0.1% over a 10-km traverse (Konolige, Agrawal, & Sola, 2007). BA can offer further gains in accuracy (Li, Archinal, Arvidson, Bell, Christensen, et al., 2006), but efforts to automate the process are ongoing (Li, Di, & Howard, 2007). Despite significant advances in the technology, such dead-reckoning approaches are not suitable for long-range localization (e.g., more than 10 km) because they will always exhibit unbounded error growth with distance traversed (Olson, Matthies, Schoppers, & Maimone, 2003).

Global localization techniques can be used to correct dead-reckoning pose estimates once these become unreliable. On Earth, the global positioning system (GPS) is commonly used for this purpose. However, the satellite infrastructure required for such a system is not feasible for non-Earth applications. This paper proposes an alternate solution that aligns a rover-based three-dimensional (3D) local map to a satellite-based 3D global map.

In this research, the local map is a point cloud obtained from a time-of-flight LIDAR (*Light Detection and Ranging*). This instrument can measure distance to far-away objects by rapidly firing a laser and measuring the time for reflected beams to return. In a surveying configuration, a LIDAR can sample terrain with centimeter accuracy at a range of up to 1.5 km, making it a vital guidance and navigation sensor. The technology is also quite

Table 1. Summary of coverage and resolutions for various satellite data sets.

Target	Instrument	Description	Coverage	Horiz. res.	Vert. res.
Moon	LOLA	Laser altimeter	Total	50–100 m	10 cm
	LROC	Camera	>10%	50 cm	NA
Mars	MOLA2	Laser altimeter	Total	100 m	1 m
	HiRISE	Camera	2%	<1 m	<1 m

Sources: For LOLA and LROC see Chin, Brylow, Foote, Garvin, Kasper, et al. (2007). For MOLA2 see Smith, Zuber, Frey, Carvin, Head, et al. (2001). For HiRISE see Johnston, Graf, Zurek, and Eisen (2005) and McEwen, Eliason, Bergstrom, Bridges, Hansen, et al. (2002).

robust, having been tested in numerous applications on Earth (Bakambu, Allard, & Dupuis, 2006; Vandapel, Donamukkala, & Hebert, 2006) and in space (Nimelman, Tripp, Allen, Hiemstra, & McDonald, 2006; Shotwell, 2005).

The global map may be acquired from a satellite-based laser altimeter (e.g., LOLA, MOLA2), or by extracting 3D information from a stereo pair of high-resolution satellite images (e.g., LROC, HiRISE) (Li, Di, Hwangbo, & Chen, 2007). Current satellites have extensive and accurate coverage of the Moon and Mars (see Table I). Therefore, relevant map data could be loaded into a rover even before it begins its mission. This would allow the rover to autonomously localize without requiring any additional map data from Earth.

The objective of this paper is to develop and demonstrate MOGA (Multi-frame Odometry-compensated Global Alignment), a novel algorithm for long-range localization of a rover by matching rover-based, 3D LIDAR maps to a 3D orbital map. Given the same inputs and making the same assumptions, performance is compared between MOGA and VIPER (Cozman, Krotkov, & Guestrin, 2000), a well-documented, competing algorithm that is currently the state of the art for this type of localization problem. It will also be shown how commonly available measurements such as heading from a sun sensor (Enright, Furgale, & Barfoot, 2009) and VO can be used to improve the efficiency, robustness, and accuracy of MOGA.

The paper is divided into a number of sections detailing the methodology and proof-of-concept experiments. Section 2 references key related work. The overall architecture is outlined in Section 3, followed by a detailed description of the methodology in Sections 4–6. In Sections 7 and 8, data and results are presented from field tests at a Mars/Moon analog site.

2. RELATED WORK

A series of matching techniques were considered to align the global and local maps. Iterative closest point (ICP) (Besl & McKay, 1992) and other full-surface matching algorithms are not appropriate because they attempt to minimize the distance between all points on both surfaces and are therefore too much of a computational burden. This is particularly true when using a LIDAR scan, which could con-

tain millions of points spread out over hundreds of meters. There is also no guarantee of convergence, particularly for maps expressed in highly differing reference frames. ICP is more often used as a postprocessing step to refine the pose estimate (Bakambu et al. 2006; Johnson, 1997).

Johnson and Montgomery (2008) investigate a number of localization techniques for precise lunar landing. Among these is cross correlation, which like ICP uses the entire LIDAR scan to find a match. However, this method is not ideal when dealing with a large global map or with the significant resolution and perspective differences between local and global maps.

A feature-based approach is instead considered in which interest points are first extracted from the global and local maps and then matched in search of global-local feature correspondences. If at least three, noncollinear feature correspondences are found, a rigid 3D transformation can be expressed that aligns the two maps (Horn, 1987).

Some examples of 3D feature matching techniques are spin images (Johnson, 1997), point fingerprints (Sun, Paik, Koschan, Page, & Abidi, 2003), and local-shape descriptors (Taati, Bondy, Jasiobedzki, & Greenspan, 2007). These algorithms compare features by creating a descriptor for each feature based on the local surface geometry. For example, a spin-image descriptor is a two-dimensional (2D) histogram representing the distances of surrounding map points to the local tangent plane and normal vector at the feature's location. Upon comparison of two features' descriptors, a correspondence is made if their descriptors are sufficiently similar. Perhaps more well-known are the 2D SIFT (Lowe, 1999) and SURF (Bay, Tuytelaars, & Van Gool, 2006) feature descriptors, which are used to analyze images in an analogous way.

The most prominent features visible to both a low-resolution orbital map and a high-resolution LIDAR map are topographic peaks. Because one side of a peak will always be hidden from the rover, features from the LIDAR map will be only partially describable as shown in Figure 1. It will therefore be difficult to compare features between the two maps if a descriptor-based approach is used. A more convenient alternative is to search for similar *feature constellations*, where the spacing between features now effectively acts as the descriptor. Chen, Hung, and Cheng (1999) developed a framework for this approach known as

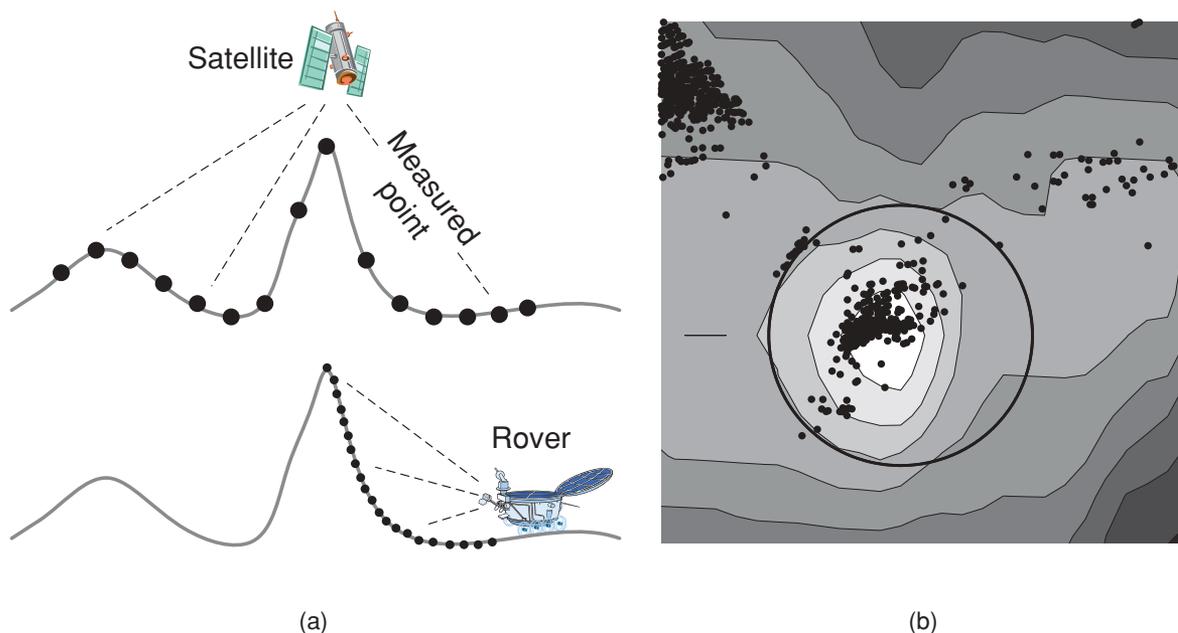


Figure 1. (a) Difficulties in matching a low-resolution orbital map (top) to a high-resolution, occluded LIDAR map (bottom). (b) Occluded LIDAR map overlaid onto a contour map of the terrain. LIDAR origin is located off the map in the upper left direction. Elevation increases with lighter shading.

the Data-Aligned Rigidity-Constrained Exhaustive Search (DARCES).

Previous work has investigated these techniques in the context of localization. Bakambu et al. (2006) examined the problem of matching two LIDAR scans. The spin-image and point fingerprint descriptors were evaluated and combined with DARCES to improve results. Vandapel et al. (2006) localized an unmanned, LIDAR-equipped, ground vehicle in heavily vegetated environments with the spin-image technique. The global map was obtained from a LIDAR-equipped helicopter.

Li, Di, and Howard (2007) present a framework that uses VO and BA to globally localize a rover with an image network containing orbital data. Their implementation suffers from a lack of automation due to the difficulty in overcoming significant differences in resolution and perspective between orbital and ground imagery.

The VIPER algorithm (Cozman et al., 2000) estimates the global position of a rover by matching the horizon skyline in a panoramic image to predicted skylines at various positions on the global map. To reduce the complexity of the problem, the algorithm assumes that the orientation of the platform has been estimated using celestial and gravitational sensors. As demonstrated in later sections, VIPER reaches a solution very quickly because the bulk of the computational work is in extracting predicted skylines from the global map, which can be done offline. However, VIPER

depends on good atmospheric conditions (e.g., high visibility, daytime use) and uses only topographic information on the skyline projected into image space. This ignores information below the skyline and may result in ambiguities between locations with significantly different topography that produce similar skyline profiles.

VIPER has been subjected to field tests, producing position estimates on more than 14 image panoramas from five different test sites with large, 150-km² global maps of resolution 30 m or less. Cozman et al. (2000) report localization accuracy between 84 and 564 m. As such, VIPER is considered the state of the art in automatic, global localization using a low-resolution global map. This paper presents MOGA, a novel solution to the long-range localization problem with an approach similar to that of VIPER but using 3D maps instead of 2D images.

3. ARCHITECTURE OVERVIEW

The architecture is developed for the general case in which a rover traverses over some distance and occasionally stops to scan the terrain with the LIDAR as shown in Figure 2. The general formulation can also be simplified by omitting odometry to examine single-scan localization.

The goal is to determine the rover's pose at each scan site with respect to the global map's reference frame, \mathcal{F}_o .

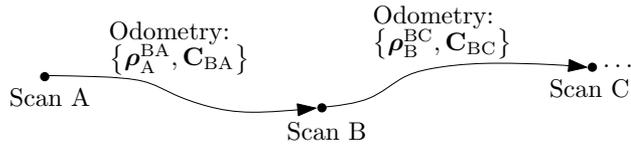


Figure 2. Rover traverses to LIDAR scan sites collecting odometry along the way (e.g., the odometry-measured translation and rotation from site A to B are, respectively, ρ_A^{BA} and C_{BA}).

The rover pose at scan site ℓ is defined as a transformation, $\mathbf{T}_{\ell o} := \{\mathbf{t}_{\ell o}, \mathbf{S}_{\ell o}\}$, from \mathcal{F}_o to the rover's local frame \mathcal{F}_ℓ , where $\mathbf{t}_{\ell o}$ and $\mathbf{S}_{\ell o}$ are, respectively, the translation vector and rotation matrix from \mathcal{F}_o to \mathcal{F}_ℓ . An overview is presented in Figure 3 and summarized as follows:

- Feature detection:** Features are detected from the global and local maps.
- Feature matching:** DARCES obtains global–local feature correspondences and an initial estimate of the rover's poses.
- Pose refinement:** Global–local feature correspondences are used in combination with orientation and odometry measurements to refine rover pose estimates.

The sections to follow discuss each architecture component in further detail.

4. FEATURE DETECTION

Features must first be detected [box (a) in Figure 3] from the global map (the *global features*) and from the local map (the *local features*). It is assumed that a global elevation map is given with resolution L_{xy} , as well as one or more local elevation maps. Generally, the most prominent features common to both maps are topographic peaks. These peaks are detected using a local maxima detector based on morphological dilation¹ (Haralick & Shapiro, 1992; van den Boomgaard & van Balen, 1992). Other feature detectors could also be used within the presented framework.

The scan is leveled using pitch and roll measurements (e.g., inclinometer). This ensures that the $+z$ directions (i.e., vertical) of the global and local maps roughly coincide. The local map is gridded [box (a) in Figure 4] to the global map resolution, L_{xy} , to ensure that the scale of detected global and local features is the same. When gridding, nearest-neighbor interpolation is used because it tends to reduce the number of poor, false peaks detected. Features may now be detected using morphological dilation.

Morphological dilation [see box (b) in Figure 4] replaces lower grid values with neighboring higher grid val-

ues. Applied to an elevation map, this will effectively blur out low elevations. The extent of the blurring depends on the size and shape of the window used. The operation replaces the window's center point value with the highest value within the bounds of the window. Once dilation is completed for all points on the grid, the blurred map is compared to the original map. Cells with no change in value are interpreted as local maxima.

The dilation window is chosen to be a pixelated circle as shown in Figure 5. This is done to make the coverage of the window as uniform as possible in all directions. The radius of this circle limits the size of the detected features, as well as the distance between features. The minimum distance between features, D_{detect} , depends on the global map resolution, L_{xy} , such that $D_{\text{detect}} := n \times L_{xy}$, where n is the circle's cell radius. The same window is applied to global and local maps.

Features must now undergo some final processing to eliminate false peaks that might be detected in flat areas [see box (c) in Figure 4]. This is primarily done by ensuring that the minimum distance between any two features is D_{detect} . Global features are particularly prone to be detected in large clusters in flat areas as a result of the low resolution of the global map. Therefore, each global feature is also tested for flatness. The flatness metric is the maximum z deviation between a detected feature and all other map points within a range of D_{detect} . If this quantity is less than $E_{\text{flat}} := L_z/2$, where L_z is the global map z resolution, the feature is deleted because it is located in an uninteresting, flat area. Sample feature detection results are shown in Figure 6.

A good uncertainty model is necessary to correctly assess the quality of individual features and global–local matches. Global feature positional uncertainty is assumed to be equal to the global map's measurement uncertainty (i.e., the position uncertainty of a measured 3D point in the global map). The global feature covariance matrix is therefore $\mathbf{R}_G := \text{diag}\{\sigma_{r_{G_{xy}}}^2, \sigma_{r_{G_{xy}}}^2, \sigma_{r_{G_z}}^2\}$, where $\sigma_{r_{G_{xy}}}$ and $\sigma_{r_{G_z}}$ are, respectively, the standard deviations of x – y and z global map measurements.

The local map uncertainty (i.e., the position uncertainty of a measured 3D point from the LIDAR), $\sigma_{r_{LM}}$, is comparatively very small (<1 m) even for measurements far from the LIDAR's origin. However, the effect of occlusions will dominate the uncertainty in local feature measurements. To address this issue, the following heuristic was employed, which worked well in practice. Other approaches might also be possible. The x – y uncertainty is related to the size of a local feature, $\sigma_{r_{L_{xy}}} := D_{\text{detect}}/2$. The z uncertainty, $\sigma_{r_{L_z}}$, will depend on the elevation of the observed feature with respect to the LIDAR. If the feature is below the LIDAR, $\sigma_{r_{L_z}}$ is set to the local map uncertainty, $\sigma_{r_{LM}}$. If the feature is above the LIDAR, $\sigma_{r_{L_z}}$ also depends on the feature's elevation angle with respect to the LIDAR, θ_L , and the uncertainty in the x – y positions, $\sigma_{r_{L_{xy}}}$, as shown

¹This was inspired by code found on the Matlab Central repository as "localMaximum.m" by Yonathan Nativ.

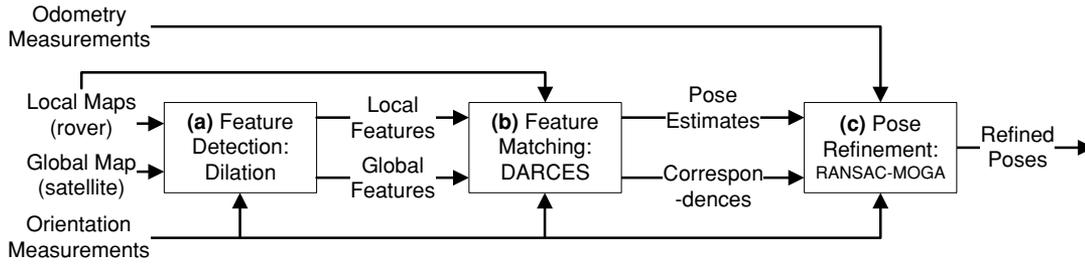


Figure 3. Overall architecture.

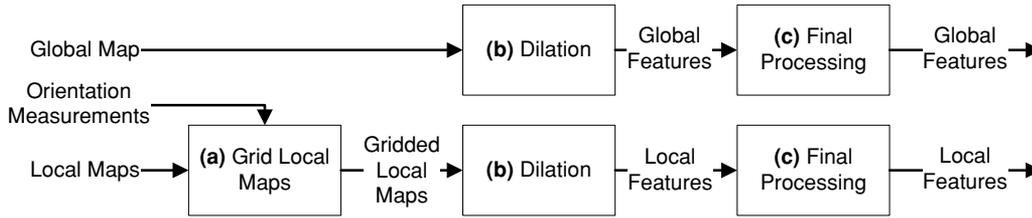


Figure 4. Feature detection architecture [blowup of box (a) in Figure 3].

in Figure 7. This gives

$$\sigma_{r_{Lz}}^2 := \begin{cases} \sigma_{r_{LM}}^2 + (\sigma_{r_{Lxy}} \tan \theta_L)^2, & \theta_L \geq 0 \\ \sigma_{r_{LM}}^2, & \theta_L < 0 \end{cases} \quad (1)$$

The local position covariance matrix is then $\mathbf{R}_L := \text{diag}\{\sigma_{r_{Lxy}}^2, \sigma_{r_{Lxy}}^2, \sigma_{r_{Lz}}^2\}$.

5. FEATURE MATCHING

The feature matching methodology [box (b) in Figure 3] is based on the DARCES algorithm (Chen et al., 1999) depicted in Figure 8. DARCES searches for similar constellations of features between two maps by comparing features' relative positions. LIDAR scans are processed individually.

5.1. Hypothesis Generation

Hypothesized correspondences must first be generated between global and local features [box (a) in Figure 8]. A *hypothesis* is defined as a group of possible correspondences

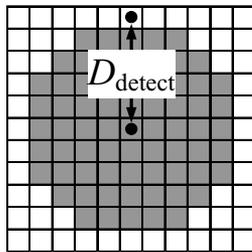


Figure 5. Pixelated circle with $n = 5$.

between three unique local features, called *control points*, and three unique global features. A hypothesis is generated if the distances between three global features are similar to the distances between three local features.

To quantify this statement for a pair of correspondences, consider the case in which two local control point features L1 and L2 are hypothesized to correspond, respectively, with global features G1 and G2. The two local features are separated by a distance d_L and the two global features by d_G . If the correspondences were correct and all measurements noiseless, d_L and d_G would be equal. However, noise will produce an error, $E_{GL} := d_G - d_L$.

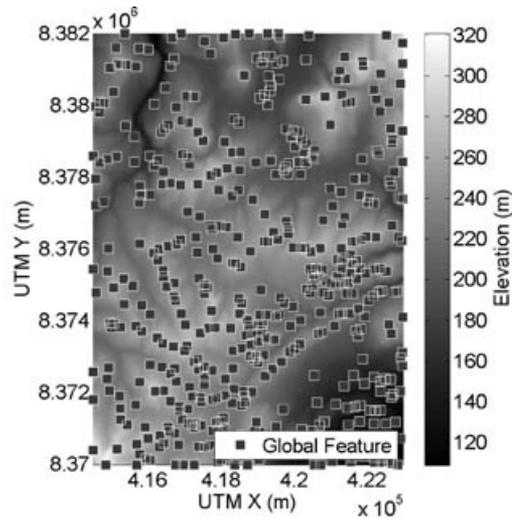
For these feature correspondences to be considered for a hypothesis, E_{GL} must satisfy $|E_{GL}| \leq t$. The threshold, t , is chosen such that $t := 2\sigma_{E_{GL}}$, where $\sigma_{E_{GL}}$ is one standard deviation of uncertainty in E_{GL} . Set at two standard deviations, 95% of valid correspondences should fall within this threshold and be correctly detected as a hypothesis. Therefore, assuming zero covariance,

$$|E_{GL}| \leq 2(\sigma_{d_G}^2 + \sigma_{d_L}^2)^{1/2}, \quad (2)$$

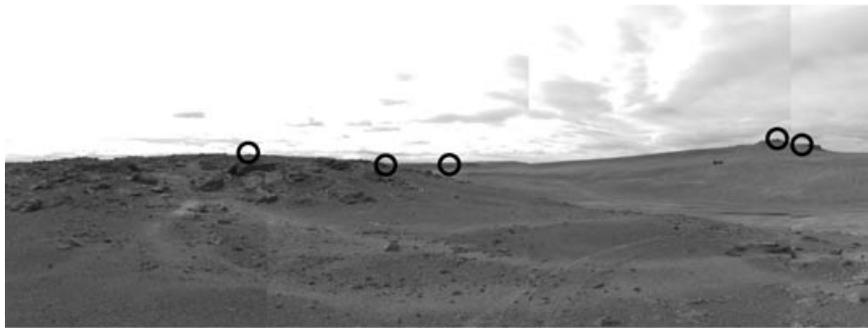
where σ_{d_G} and σ_{d_L} are, respectively, the uncertainties in d_G and d_L . To determine these quantities, uncertainty must be propagated from known feature position uncertainties. Again assuming zero covariance between measurements, it can be shown that

$$\sigma_{d_L}^2 := \frac{1}{d_L^2} (\mathbf{p}_{L1} - \mathbf{p}_{L2})^T (\mathbf{R}_{L1} + \mathbf{R}_{L2}) (\mathbf{p}_{L1} - \mathbf{p}_{L2}), \quad (3)$$

$$\sigma_{d_G}^2 := \frac{2}{d_G^2} (\mathbf{p}_{G1} - \mathbf{p}_{G2})^T \mathbf{R}_G (\mathbf{p}_{G1} - \mathbf{p}_{G2}), \quad (4)$$



(a)



(b)

Figure 6. Detected features. (a) 495 global features plotted on global map. (b) Local 3D features (scan A08) projected onto a section of a 2D panoramic image to show detection of peaks.

where the \mathbf{p} and \mathbf{R} terms are, respectively, the positions and covariance matrices of the indicated features. If three pairs of features satisfy these conditions, a hypothesis is generated. However, a hypothesis is not guaranteed to be correct due to noise in feature position measurements. Therefore, many control point groups are tested to increase the chances of finding a valid hypothesis.

5.2. Hypothesis Evaluation

The validity of a hypothesis is evaluated [box (b) in Figure 8] based on the transformation it produces between the global and local frame. For a hypothesis i , this transformation from \mathcal{F}_o to \mathcal{F}_l , $\mathbf{T}_{lo}^i := \{\mathbf{t}_{lo}^i, \mathbf{S}_{lo}^i\}$, can be obtained using a least-squares point-alignment algorithm (e.g. Arun,

Huang, & Blostein, 1987) to align the three global and three local features that comprise a hypothesis.

To improve the efficiency and robustness of DARCES, hypotheses are first screened with a number of simple tests [box (a) in Figure 9]. A map-boundary test rejects hypotheses that estimate the rover’s position outside the global map. A z -deviation test ensures that an estimated z position is within E_{zdev} of the global map’s elevation at the estimated x - y position. Finally, orientation tests check that estimates and measurements (e.g., inclinometer, sun sensor) of roll α , pitch β , and heading γ differ by no more than E_α , E_β , and E_γ . These thresholds must be chosen such that they reject a low proportion of valid hypotheses and a high proportion of invalid hypotheses. For example, $E_{zdev} := 2\sigma_{r_{Gz}}$, where $\sigma_{r_{Gz}}$ is one standard deviation of the global map z uncertainty.

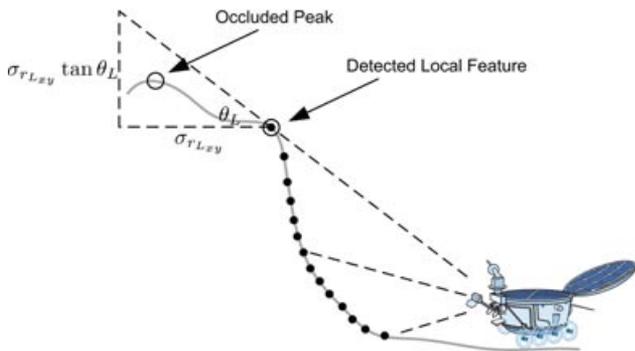


Figure 7. Local feature uncertainty due to occlusions.

A measure of fitness is then calculated for each hypothesis by examining how well the hypothesized transformation aligns LIDAR scan points to the global map [box (b) in Figure 9]. To more efficiently and robustly calculate this metric, the full LIDAR scan is decimated to half the global map resolution, $L_{xy}/2$. These more evenly spread local map points are called the *reference points*.

For a hypothesis i , the fitness metric, f_i , is the average absolute z error between the reference points, transformed to the global frame with T_{lo}^i , and the global map:

$$f_i := -\frac{1}{N_{ref}} \sum_{j=1}^{N_{ref}} |z_{i,j}^R - z_{i,j}^G|, \quad (5)$$

where N_{ref} is the number of reference points, $z_{i,j}^R$ is the z position of the transformed reference point j in the global frame, and $z_{i,j}^G$ is the interpolated global map elevation at the x - y position of the transformed reference point j .

The negative is applied so that a low average absolute error corresponds to a high fitness. After much experimentation, it has become evident that this hybrid approach, in which sparse features are used to generate hypotheses and denser scans to validate these hypotheses, is paramount in allowing the algorithm to function both efficiently and accurately.

Now that a fitness is associated with each hypothesis, a search is made for valid hypotheses. This is a difficult task because many hypotheses could remain even after the screening stages. Johnson (1997) handles this by setting a threshold for data points that have a fitness much better than all others. Data points exceeding this threshold would then be considered valid. This method was attempted (Carle, 2009), but was dropped in favor of an approach that proved to be more robust. The alternative method examines the clustering of hypothesized positions. This technique requires that several valid hypotheses are available in the data. If several valid hypotheses exist, then there should be a cluster of closely positioned, high-fitness hypotheses in the data.

The first step is to select the top N_{top} hypotheses with the highest fitness. The mean position, t_m , of these top hypotheses is then calculated. If all N_{top} hypotheses are within a distance D_{valid} of t_m , then all hypotheses in this group are considered valid. The hypothesis with the highest fitness is then returned, and its associated transformation estimate and global-local feature correspondences output to the next stage of the algorithm. If no group of valid hypotheses is found once all combinations of control points have been exhausted, then DARCES returns no solution. Sample data are shown in Figure 10 to help underscore the significant challenges in locating a valid hypothesis.

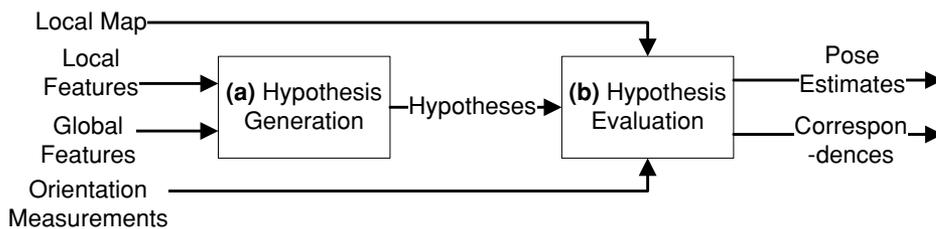


Figure 8. Feature matching architecture [blowup of box (b) in Figure 3].

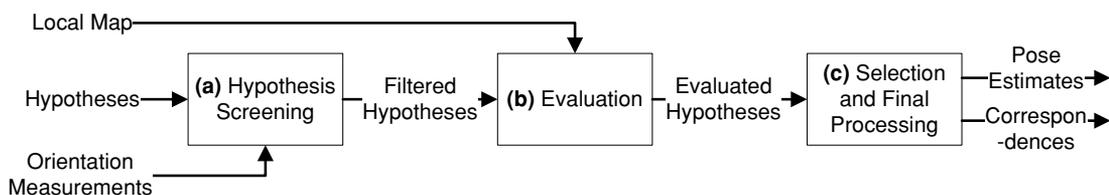


Figure 9. Overview of hypothesis evaluation and selection.

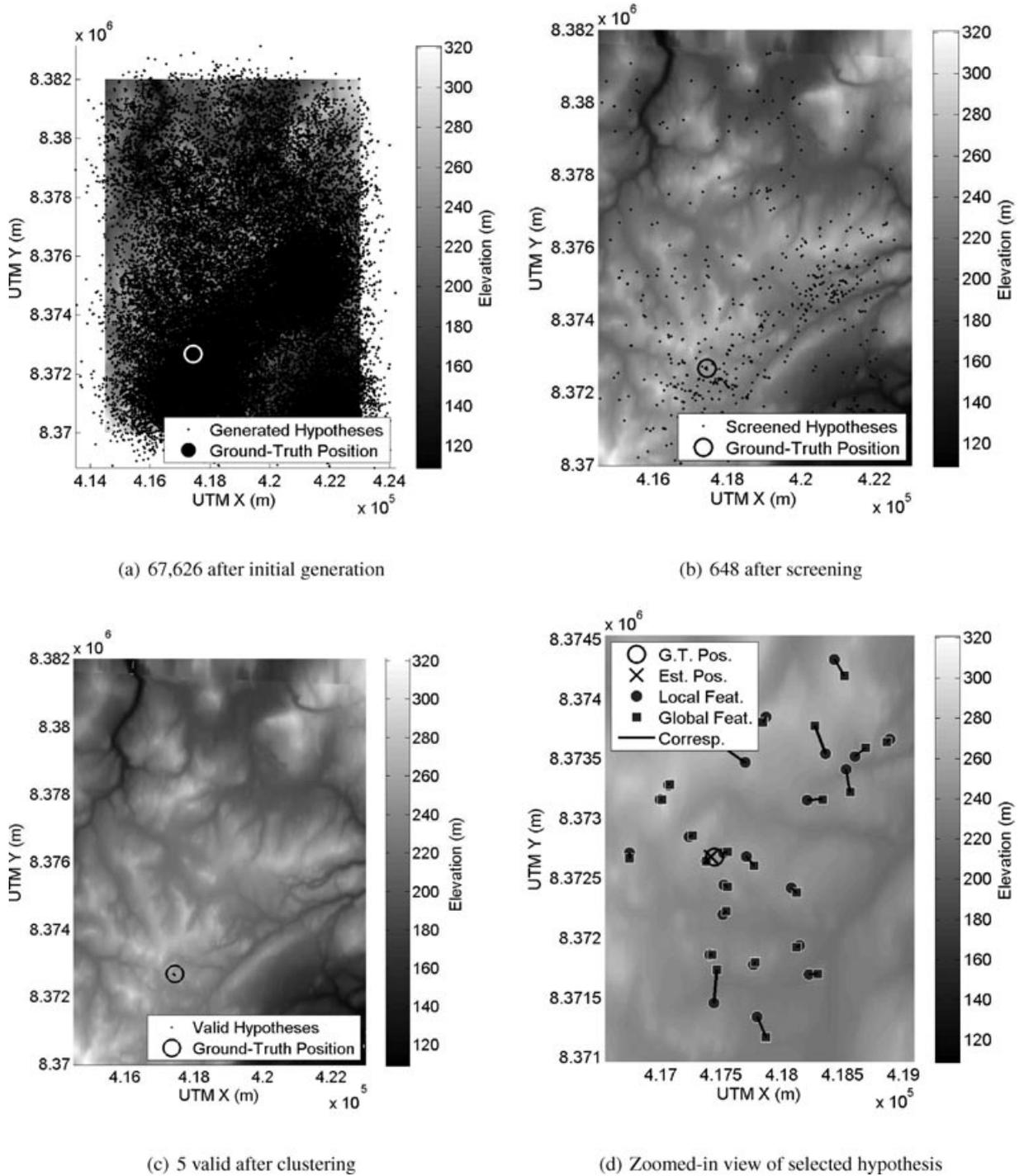


Figure 10. Hypotheses for scan A08 at various stages in the algorithm. In (d), local features are transformed to the global frame using the transformation from the hypothesis with the best fitness.

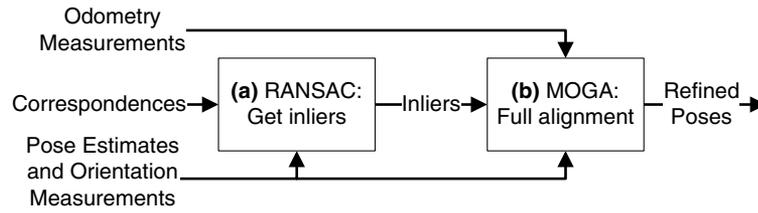


Figure 11. Pose refinement architecture [blowup of box (c) in Figure 3].

6. POSE REFINEMENT

The feature correspondences obtained from feature matching are now combined with odometry and orientation measurements into a simultaneous localization and mapping (SLAM) problem to refine pose estimates [box (c) in Figure 3]. The MOGA algorithm, developed for this paper, solves this problem by minimizing the errors between all available measurements and desired estimates. Outlier feature correspondences are also rejected with an enveloping RANSAC (*R*andom *S*ample *C*onsensus) algorithm. This is essentially a batch SLAM algorithm that fuses relative and absolute pose measurements over an entire rover traverse. The algorithm is outlined in Figure 11.

6.1. MOGA

The main goal is to estimate the transformations from the global frame, \mathcal{F}_o , to each of the M local frames, \mathcal{F}_ℓ . This M -frame alignment problem is depicted in Figure 12.

Input measurements are assumed to be corrupted with white, zero-mean, Gaussian noise. There are four types of measurement:

- In the global frame, \mathcal{F}_o , there are N measured feature positions, $\mathbf{r}_o^{f_j}$, with 3×3 covariance matrix, $\mathbf{R}_o^{f_j}$. These are the features detected in the orbital map.
- In each local frame, \mathcal{F}_ℓ , there are $N_\ell \leq N$ measured feature positions, where $\mathbf{r}_\ell^{f_{\ell,k}}$ is the position of a feature k in \mathcal{F}_ℓ . The 3×3 covariance matrix for this measurement is $\mathbf{R}_\ell^{f_{\ell,k}}$. These are the features detected in the LIDAR scans. Corresponding global features have been found for each local feature.
- There are odometry measurements of the rotation, $\mathbf{C}_{o_{\ell+1}o_\ell}$, and translation, $\boldsymbol{\rho}_{o_\ell}^{o_{\ell+1}}$, between adjacent local frames \mathcal{F}_ℓ and $\mathcal{F}_{\ell+1}$. The combined 6×6 covariance matrix is $\mathbf{Q}_{o_\ell}^{o_{\ell+1}}$. These would be obtained from VO, for example.
- For each local frame, \mathcal{F}_ℓ , there is an orientation measurement, $\mathbf{C}_{o_\ell o}$, from \mathcal{F}_o to \mathcal{F}_ℓ with 3×3 covariance matrix $\mathbf{R}_{o_\ell}^{o_\ell}$. These would be measured with a sun sensor and an inclinometer, for example.

Three $(2M + N)$ design parameters must be estimated. The three types are as follows:

- An estimated rotation, $\mathbf{S}_{o_\ell o}$, from the global frame \mathcal{F}_o to each of the M local frames \mathcal{F}_ℓ .
- An estimated translation, $\mathbf{t}_{o_\ell o}^{o_\ell}$, from the global frame \mathcal{F}_o to each of the M local frames \mathcal{F}_ℓ and expressed in \mathcal{F}_o .
- An estimated position, $\mathbf{p}_o^{l_j}$, expressed in \mathcal{F}_o for each of the N unique features. These feature landmarks will help to improve rotation and translation estimates.

The optimal design parameters will be obtained by minimizing the sum of squared errors between estimates and measurements. There are four types of error (one for each measurement):

- J_{f_j} , between estimated landmark feature positions and measured global feature positions.
- $J_{f_{\ell,k}}$, between estimated landmark feature positions and measured local feature positions.
- $J_{o_{\ell+1}o_\ell}$, between estimated and odometry-measured transformations from \mathcal{F}_ℓ to $\mathcal{F}_{\ell+1}$.
- $J_{o_\ell o}$, between estimated and measured frame rotations from \mathcal{F}_o to \mathcal{F}_ℓ .

To allow all design parameters to be optimized simultaneously, each of these error terms must be expressed as a function of a common design parameter column, \mathbf{z} . This parameter is further discussed in the next section. The overall objective function, $J(\mathbf{z})$, is then

$$J(\mathbf{z}) := \sum_{j=1}^N J_{f_j}(\mathbf{z}) + \sum_{\ell=1}^M \sum_{k=1}^{N_\ell} J_{f_{\ell,k}}(\mathbf{z}) + \sum_{\ell=1}^{M-1} J_{o_{\ell+1}o_\ell}(\mathbf{z}) + \sum_{\ell=1}^M J_{o_\ell o}(\mathbf{z}). \quad (6)$$

The optimal design parameter column, \mathbf{z}^* , is sought by minimizing the objective function through unconstrained optimization: $\mathbf{z}^* := \operatorname{argmin}_{\mathbf{z}} J(\mathbf{z})$. An iterative Gauss–Newton algorithm (Björck, 1996) is used to solve this nonlinear least-squares problem. The sections to follow derive the error terms and explain how they are combined to optimize the design parameters.

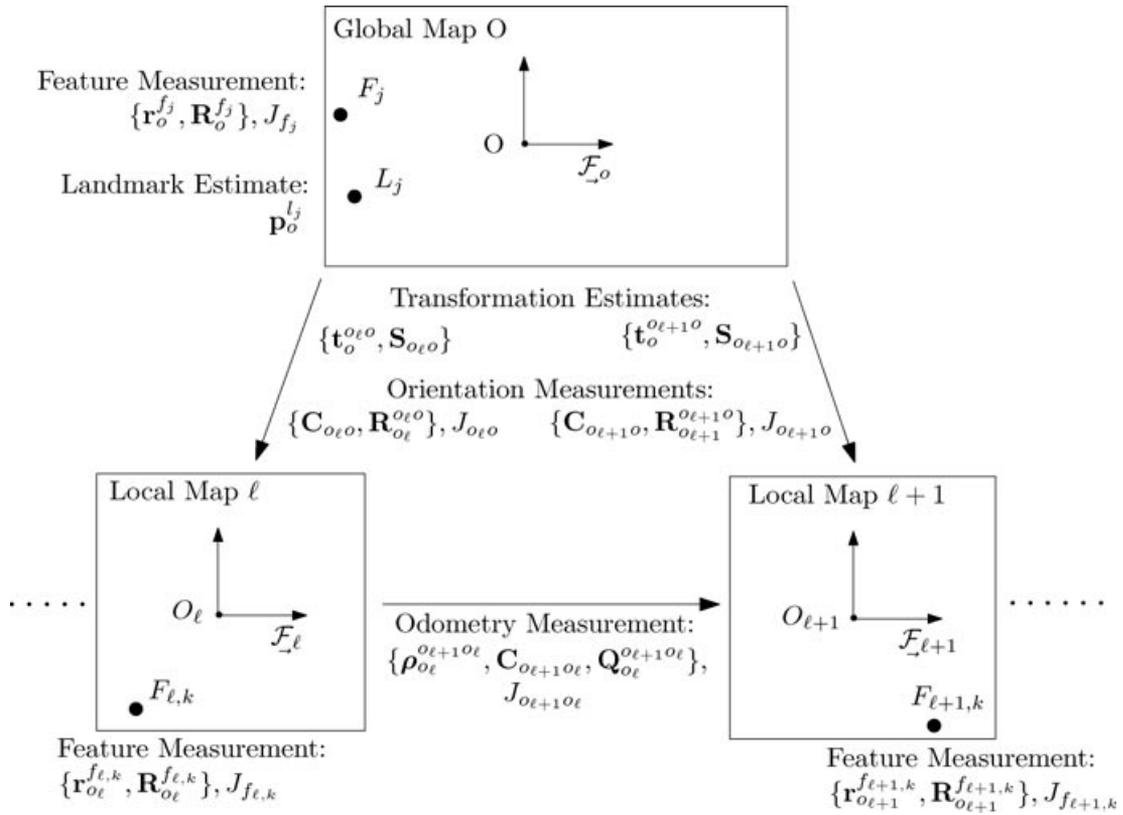


Figure 12. The M -frame alignment problem.

6.2. Local Feature Terms, $J_{f_{\ell,k}}$

A local feature’s error term, $J_{f_{\ell,k}}$, is simply the squared difference between its position measurement, $\mathbf{r}_{o_\ell}^{f_{\ell,k}}$, and its position estimate, $\mathbf{p}_o^{l_{\omega(\ell,k)}}$. The latter must be transformed to \mathcal{F}_ℓ using the estimated translation and rotation $\mathbf{t}_o^{o_\ell}$ and $\mathbf{S}_{o_\ell o}$. This produces

$$J_{f_{\ell,k}} := \frac{1}{2} [\mathbf{r}_{o_\ell}^{f_{\ell,k}} - \mathbf{S}_{o_\ell o} (\mathbf{p}_o^{l_{\omega(\ell,k)}} - \mathbf{t}_o^{o_\ell})]^T \times \mathbf{R}_{o_\ell}^{f_{\ell,k}}^{-1} [\mathbf{r}_{o_\ell}^{f_{\ell,k}} - \mathbf{S}_{o_\ell o} (\mathbf{p}_o^{l_{\omega(\ell,k)}} - \mathbf{t}_o^{o_\ell})]. \quad (7)$$

Note that maximum likelihood has been implemented by weighting the error term with the inverse covariance matrix of the measurement involved, $\mathbf{R}_{o_\ell}^{f_{\ell,k}}$. Also known as the Mahalanobis distance (Mahalanobis, 1936), this causes more precise measurements to be weighted more heavily.

When optimizing with the Gauss–Newton algorithm, an objective function quadratic in the design parameters is needed. This is not the case with the objective function in question due to the rotations involved. Therefore,

it is necessary to make a quadratic approximation for $J(\mathbf{z})$. The translation and landmark position parameters are perturbed such that $\mathbf{t}_{o_\ell}^{o_\ell} =: \bar{\mathbf{t}}_{o_\ell}^{o_\ell} + \delta \mathbf{t}_{o_\ell}^{o_\ell}$ and $\mathbf{p}_o^{l_{\omega(\ell,k)}} =: \bar{\mathbf{p}}_o^{l_{\omega(\ell,k)}} + \delta \mathbf{p}_o^{l_{\omega(\ell,k)}}$, where $\bar{\mathbf{t}}_{o_\ell}^{o_\ell}$ and $\bar{\mathbf{p}}_o^{l_{\omega(\ell,k)}}$ are nominal estimates and $\delta \mathbf{t}_{o_\ell}^{o_\ell}$ and $\delta \mathbf{p}_o^{l_{\omega(\ell,k)}}$ are perturbations.

Rotation perturbations must be handled more carefully because rotation matrices cannot be summed but must be multiplied in a specific order. This gives $\mathbf{S}_{o_\ell o} =: \delta \mathbf{S}_{o_\ell o} \bar{\mathbf{S}}_{o_\ell o}$, where $\bar{\mathbf{S}}_{o_\ell o}$ is the nominal estimate and $\delta \mathbf{S}_{o_\ell o}$ is a small, perturbing rotation. It will be convenient to express this small rotation as a vector using the approximation $\delta \mathbf{S}_{o_\ell o} \approx \mathbf{1} - \delta \boldsymbol{\theta}_{o_\ell}^\times$, where $\delta \boldsymbol{\theta}_{o_\ell}^\times$ is a skew symmetric matrix created by applying the cross operator to an infinitesimal rotation vector, $\delta \boldsymbol{\theta}_{o_\ell}$. This vector is an axis-angle parametrization of a rotation. For more details on these concepts, see Hughes (1986).

The overall rotation can then be decomposed as $\mathbf{S}_{o_\ell o} \approx (\mathbf{1} - \delta \boldsymbol{\theta}_{o_\ell}^\times) \bar{\mathbf{S}}_{o_\ell o}$, where the rotation perturbation, $\delta \boldsymbol{\theta}_{o_\ell}$, may be isolated knowing that the cross product is anticommutative, i.e., $\mathbf{u}^\times \mathbf{v} = -\mathbf{v}^\times \mathbf{u}$. Substituting these results into the objective function of Eq. (7), using this anticommutative property, and expressing the objective function for clarity

as $J_{f_{\ell,k}} = \frac{1}{2} \mathbf{e}_{f_{\ell,k}}^T \mathbf{R}_{o_\ell}^{f_{\ell,k}^{-1}} \mathbf{e}_{f_{\ell,k}}$ gives

$$\begin{aligned} \mathbf{e}_{f_{\ell,k}} &:= \mathbf{r}_{o_\ell}^{f_{\ell,k}} - \bar{\mathbf{S}}_{o_\ell o} (\bar{\mathbf{p}}_o^{l_{\omega(\ell,k)}} - \bar{\mathbf{t}}_o^{o_\ell o}) \\ &\quad - \left\{ [\bar{\mathbf{S}}_{o_\ell o} (\bar{\mathbf{p}}_o^{l_{\omega(\ell,k)}} - \bar{\mathbf{t}}_o^{o_\ell o})]^\times \delta \boldsymbol{\theta}_{o_\ell o} \right. \\ &\quad \left. + \bar{\mathbf{S}}_{o_\ell o} (\delta \mathbf{p}_o^{l_{\omega(\ell,k)}} - \delta \mathbf{t}_o^{o_\ell o}) \right\}. \end{aligned} \quad (8)$$

Note that an approximation has been made by dropping products of small terms. The remaining perturbed quantities make up the \mathbf{z} column, which can now be expressed as $\mathbf{z} := [\delta \mathbf{p}'_o \ \delta \mathbf{t}_o \ \delta \boldsymbol{\theta}]^T$, where $\delta \mathbf{p}'_o$ represents the column of all N landmark position perturbations in $\underline{\mathcal{F}}_o$ and $\delta \mathbf{t}_o$ and $\delta \boldsymbol{\theta}$ are, respectively, the columns of all M translation and M rotation perturbations from $\underline{\mathcal{F}}_o$ to $\underline{\mathcal{F}}_\ell$. The error term can then be written in a convenient form with $\mathbf{e}_{f_{\ell,k}} = \mathbf{b}_{f_{\ell,k}} - \mathbf{A}_{f_{\ell,k}} \mathbf{z}$, where

$$\mathbf{b}_{f_{\ell,k}} := \mathbf{r}_{o_\ell}^{f_{\ell,k}} - \bar{\mathbf{S}}_{o_\ell o} (\bar{\mathbf{p}}_o^{l_{\omega(\ell,k)}} - \bar{\mathbf{t}}_o^{o_\ell o}), \quad (9)$$

$$\mathbf{A}_{f_{\ell,k}} := [\mathbf{D}_{f_{\ell,k}} \ \mathbf{E}_{f_{\ell,k}} \ \mathbf{F}_{f_{\ell,k}}], \quad (10)$$

$$\mathbf{D}_{f_{\ell,k}} := \underbrace{[\mathbf{0} \cdots \overbrace{\bar{\mathbf{S}}_{o_\ell o}^{\omega(\ell,k)\text{th term}}} \cdots \mathbf{0}]}_{N \text{ landmarks}}, \quad (11)$$

$$\mathbf{E}_{f_{\ell,k}} := \underbrace{[\mathbf{0} \cdots \overbrace{-\bar{\mathbf{S}}_{o_\ell o}^{\ell\text{th term}}} \cdots \mathbf{0}]}_{M \text{ frames}}, \quad (12)$$

$$\mathbf{F}_{f_{\ell,k}} := \underbrace{[\mathbf{0} \cdots \overbrace{[\bar{\mathbf{S}}_{o_\ell o} (\bar{\mathbf{p}}_o^{l_{\omega(\ell,k)}} - \bar{\mathbf{t}}_o^{o_\ell o})]^\times \cdots \mathbf{0}}^{\ell\text{th term}}} \cdots \mathbf{0}]}_{M \text{ frames}}. \quad (13)$$

6.3. Global Feature Terms, J_{f_j}

The J_{f_j} error term is derived in a similar way. For a single feature, F_j , the result is

$$J_{f_j} := \frac{1}{2} (\bar{\mathbf{p}}_o^{l_j} + \delta \mathbf{p}_o^{l_j} - \mathbf{r}_o^{f_j})^T \mathbf{R}_o^{f_j^{-1}} (\bar{\mathbf{p}}_o^{l_j} + \delta \mathbf{p}_o^{l_j} - \mathbf{r}_o^{f_j}), \quad (14)$$

which can be conveniently expressed as $J_{f_j} = \frac{1}{2} \mathbf{e}_{f_j}^T \mathbf{R}_o^{f_j^{-1}} \mathbf{e}_{f_j}$, where $\mathbf{e}_{f_j} := \mathbf{b}_{f_j} - \mathbf{A}_{f_j} \mathbf{z}$ and

$$\mathbf{b}_{f_j} := \bar{\mathbf{p}}_o^{l_j} - \mathbf{r}_o^{f_j}, \quad (15)$$

$$\mathbf{A}_{f_j} := [\mathbf{D}_{f_j} \ \mathbf{0}_{3 \times 3M} \ \mathbf{0}_{3 \times 3M}], \quad (16)$$

$$\mathbf{D}_{f_j} := \underbrace{[\mathbf{0} \cdots \overbrace{-\mathbf{1}}^{j\text{th term}} \cdots \mathbf{0}]}_{N \text{ landmarks}}. \quad (17)$$

6.4. Odometry Terms, $J_{o_{\ell+1}o_\ell}$

The odometry measurement path from $\underline{\mathcal{F}}_\ell$ to $\underline{\mathcal{F}}_{\ell+1}$ must be compared to the estimated path from $\underline{\mathcal{F}}_\ell$ to $\underline{\mathcal{F}}_o$ to $\underline{\mathcal{F}}_{\ell+1}$. The translational error, $\delta \boldsymbol{\rho}_{o_\ell}^{o_{\ell+1}o_\ell}$, and rotational error, $\delta \boldsymbol{\psi}_{o_{\ell+1}o_\ell}$, are combined to obtain an overall, scalar odometry error, $J_{o_{\ell+1}o_\ell}$, giving

$$J_{o_{\ell+1}o_\ell} := \frac{1}{2} \begin{bmatrix} \delta \boldsymbol{\rho}_{o_\ell}^{o_{\ell+1}o_\ell} \\ \delta \boldsymbol{\psi}_{o_{\ell+1}o_\ell} \end{bmatrix}^T \mathbf{Q}_{o_\ell}^{o_{\ell+1}o_\ell} \begin{bmatrix} \delta \boldsymbol{\rho}_{o_\ell}^{o_{\ell+1}o_\ell} \\ \delta \boldsymbol{\psi}_{o_{\ell+1}o_\ell} \end{bmatrix}. \quad (18)$$

The derivation is based on the same approach as in the preceding sections but is lengthy. The result is thus immediately given (for full derivation, see Carle, 2009) with the error term written in the convenient form $J_{o_{\ell+1}o_\ell} = \frac{1}{2} \mathbf{e}_{o_{\ell+1}o_\ell}^T \mathbf{Q}_{o_\ell}^{o_{\ell+1}o_\ell} \mathbf{e}_{o_{\ell+1}o_\ell}$, where $\mathbf{e}_{o_{\ell+1}o_\ell} := \mathbf{b}_{o_{\ell+1}o_\ell} - \mathbf{A}_{o_{\ell+1}o_\ell} \mathbf{z}$, and

$$\mathbf{b}_{o_{\ell+1}o_\ell} := \begin{bmatrix} \boldsymbol{\rho}_{o_\ell}^{o_{\ell+1}o_\ell} - \bar{\mathbf{S}}_{o_\ell o} (\bar{\mathbf{t}}_o^{o_{\ell+1}o} - \bar{\mathbf{t}}_o^{o_\ell o}) \\ \delta \boldsymbol{\phi}_{o_{\ell+1}o_\ell} \end{bmatrix}, \quad (19)$$

$$\mathbf{A}_{o_{\ell+1}o_\ell} := [\mathbf{0}_{6 \times N} \ \mathbf{E}_{o_{\ell+1}o_\ell} \ \mathbf{F}_{o_{\ell+1}o_\ell}], \quad (20)$$

$$\mathbf{E}_{o_{\ell+1}o_\ell} := [\mathbf{0} \cdots \overbrace{-\bar{\mathbf{S}}_{o_\ell o}^{\ell\text{th term}} \ \bar{\mathbf{S}}_{o_\ell o}^{\ell+1\text{st term}}}_{\mathbf{0}_{3 \times M}} \cdots \mathbf{0}], \quad (21)$$

$$\begin{aligned} \mathbf{F}_{o_{\ell+1}o_\ell} &:= \begin{bmatrix} \mathbf{0} \cdots [\bar{\mathbf{S}}_{o_\ell o} (\bar{\mathbf{t}}_o^{o_{\ell+1}o} - \bar{\mathbf{t}}_o^{o_\ell o})]^\times & \mathbf{0} & \cdots \mathbf{0} \\ \mathbf{0} \cdots \underbrace{-\bar{\mathbf{S}}_{o_{\ell+1}o} \bar{\mathbf{S}}_{o_\ell o}^T}_{\ell\text{th term}} & \underbrace{\mathbf{1}}_{\ell+1\text{st term}} & \cdots \mathbf{0} \end{bmatrix} \quad 6 \times M \\ & \quad (22) \end{aligned}$$

The $\delta \boldsymbol{\phi}_{o_{\ell+1}o_\ell}$ term is found by converting $\mathbf{C}_{o_{\ell+1}o_\ell} \mathbf{S}_{o_\ell o} \mathbf{S}_{o_{\ell+1}o}^T$ to axis-angle form (see Appendix A).

6.5. Measured Orientation Terms, $J_{o_\ell o}$

Estimates can be further constrained if inclinometer and/or sun-sensor orientation measurements are available. Again skipping to the result (for full derivation, see Carle, 2009), the conveniently expressed error term is $J_{o_\ell o} := \frac{1}{2} \mathbf{e}_{o_\ell o}^T \mathbf{R}_{o_\ell}^{o_\ell o} \mathbf{e}_{o_\ell o}$, where $\mathbf{e}_{o_\ell o} := \mathbf{b}_{o_\ell o} - \mathbf{A}_{o_\ell o} \mathbf{z}$, and

$$\mathbf{b}_{o_\ell o} := \delta \boldsymbol{\phi}_{o_\ell o}, \quad (23)$$

$$\mathbf{A}_{o_\ell o} := [\mathbf{0}_{3 \times N} \ \mathbf{0}_{3 \times M} \ \mathbf{F}_{o_\ell o}], \quad (24)$$

$$\mathbf{F}_{o_\ell o} := \underbrace{[\mathbf{0} \cdots \overbrace{\mathbf{1}}^{\ell\text{th term}} \cdots \mathbf{0}]}_{M \text{ frames}}. \quad (25)$$

The $\delta \boldsymbol{\phi}_{o_\ell o}$ parameter is obtained by converting $\mathbf{C}_{o_\ell o} \bar{\mathbf{S}}_{o_\ell o}^T$ to axis-angle form (see Appendix A).

6.6. Combining Terms and Optimization

With all error terms now expressed in a convenient way, the solution to the overall minimization problem can be derived. The objective function of Eq. (6) becomes

$$\begin{aligned}
 J = & \sum_{j=1}^N \left(\frac{1}{2} \mathbf{e}_{f_j}^T \mathbf{R}_{o_j}^{f_j^{-1}} \mathbf{e}_{f_j} \right) + \sum_{\ell=1}^M \sum_{k=1}^{N_\ell} \left(\frac{1}{2} \mathbf{e}_{f_{\ell,k}}^T \mathbf{R}_{o_\ell}^{f_{\ell,k}^{-1}} \mathbf{e}_{f_{\ell,k}} \right) \\
 & + \sum_{\ell=1}^{M-1} \left(\frac{1}{2} \mathbf{e}_{o_{\ell+1}o_\ell}^T \mathbf{Q}_{o_\ell}^{o_{\ell+1}o_\ell^{-1}} \mathbf{e}_{o_{\ell+1}o_\ell} \right) \\
 & + \sum_{\ell=1}^M \left(\frac{1}{2} \mathbf{e}_{o_\ell o}^T \mathbf{R}_{o_\ell}^{o_\ell o^{-1}} \mathbf{e}_{o_\ell o} \right), \quad (26)
 \end{aligned}$$

where each \mathbf{e} term is a function of \mathbf{z} . Using the Gauss–Newton algorithm, it is desired to take a step, \mathbf{z}^* , to the minimum of the current local quadratic approximation of J . Thus \mathbf{z}^* is chosen so that $\partial J^T / \partial \mathbf{z} = \mathbf{0}$. The derivative of Eq. (26) is taken with respect to \mathbf{z} and set to zero. The result is then rearranged into the familiar linear form $\mathbf{B}\mathbf{z}^* = \mathbf{y}$, where

$$\begin{aligned}
 \mathbf{B} := & \sum_{j=1}^N (\mathbf{A}_{f_j}^T \mathbf{R}_{o_j}^{f_j^{-1}} \mathbf{A}_{f_j}) + \sum_{\ell=1}^M \sum_{k=1}^{N_\ell} (\mathbf{A}_{f_{\ell,k}}^T \mathbf{R}_{o_\ell}^{f_{\ell,k}^{-1}} \mathbf{A}_{f_{\ell,k}}) \\
 & + \sum_{\ell=1}^{M-1} (\mathbf{A}_{o_{\ell+1}o_\ell}^T \mathbf{Q}_{o_\ell}^{o_{\ell+1}o_\ell^{-1}} \mathbf{A}_{o_{\ell+1}o_\ell}) \\
 & + \sum_{\ell=1}^M (\mathbf{A}_{o_\ell o}^T \mathbf{R}_{o_\ell}^{o_\ell o^{-1}} \mathbf{A}_{o_\ell o}), \quad (27)
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{y} := & \sum_{j=1}^N (\mathbf{A}_{f_j}^T \mathbf{R}_{o_j}^{f_j^{-1}} \mathbf{b}_{f_j}) + \sum_{\ell=1}^M \sum_{k=1}^{N_\ell} (\mathbf{A}_{f_{\ell,k}}^T \mathbf{R}_{o_\ell}^{f_{\ell,k}^{-1}} \mathbf{b}_{f_{\ell,k}}) \\
 & + \sum_{\ell=1}^{M-1} (\mathbf{A}_{o_{\ell+1}o_\ell}^T \mathbf{Q}_{o_\ell}^{o_{\ell+1}o_\ell^{-1}} \mathbf{b}_{o_{\ell+1}o_\ell}) \\
 & + \sum_{\ell=1}^M (\mathbf{A}_{o_\ell o}^T \mathbf{R}_{o_\ell}^{o_\ell o^{-1}} \mathbf{b}_{o_\ell o}). \quad (28)
 \end{aligned}$$

This is simply a linear system of equations that has a unique solution iff $\det \mathbf{B} \neq 0$. This singularity can be avoided with a minimum of three unique, noncollinear features spread anywhere in the entire chain of odometry-connected frames.

After convergence of Gauss–Newton, \mathbf{B} represents the inverse covariance matrix of the estimated parameters. Therefore, the variances of the estimates may be obtained from the diagonal of \mathbf{B}^{-1} .

Using the Gauss–Newton algorithm, a solution is obtained at each iteration that represents a step to the minimum of the local quadratic approximation of the objective function. A line search is then implemented, where the so-

lution, \mathbf{z} , is repeatedly multiplied by a factor $\eta < 1$ until the objective function decreases. This solution is used to update the nominal design parameters. Convergence to a local minimum is achieved when the relative change in the objective function is less than E_{converge} .

6.7. Single-Frame Localization

It is also of interest to examine the performance of single-frame localization. In this simplification of the multiple-frame case, odometry measurements are not provided as inputs to the algorithm. Therefore, the odometry terms' contributions to the updates can be dropped, giving

$$\begin{aligned}
 \mathbf{B}' := & \sum_{j=1}^N (\mathbf{A}_{f_j}^T \mathbf{R}_{o_j}^{f_j^{-1}} \mathbf{A}_{f_j}) + \sum_{\ell=1}^M \sum_{k=1}^{N_\ell} (\mathbf{A}_{f_{\ell,k}}^T \mathbf{R}_{o_\ell}^{f_{\ell,k}^{-1}} \mathbf{A}_{f_{\ell,k}}) \\
 & + \sum_{\ell=1}^M (\mathbf{A}_{o_\ell o}^T \mathbf{R}_{o_\ell}^{o_\ell o^{-1}} \mathbf{A}_{o_\ell o}), \quad (29)
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{y}' := & \sum_{j=1}^N (\mathbf{A}_{f_j}^T \mathbf{R}_{o_j}^{f_j^{-1}} \mathbf{b}_{f_j}) + \sum_{\ell=1}^M \sum_{k=1}^{N_\ell} (\mathbf{A}_{f_{\ell,k}}^T \mathbf{R}_{o_\ell}^{f_{\ell,k}^{-1}} \mathbf{b}_{f_{\ell,k}}) \\
 & + \sum_{\ell=1}^M (\mathbf{A}_{o_\ell o}^T \mathbf{R}_{o_\ell}^{o_\ell o^{-1}} \mathbf{b}_{o_\ell o}). \quad (30)
 \end{aligned}$$

The Gauss–Newton algorithm is used in the same way as before, by iteratively solving $\mathbf{B}'\mathbf{z}^* = \mathbf{y}'$ and updating the nominal design parameters. Because odometry is not provided, this single-frame pose refinement is possible only if DARCES finds a solution for the frame in question.

6.8. RANSAC

The RANSAC algorithm (Fischler & Bolles, 1981) [box (a) in Figure 11] acts on scans individually and is used to reject poor, outlier feature correspondences before a full alignment using MOGA. Other measurements, such as odometry and orientation, are assumed to be inliers. The output DARCES hypothesis is used to create correspondences between global and local features. Three of these correspondences are chosen at random and are called the *model*. The global–local frame transformation associated with this model is then used to search for outliers. An outlier is defined as a poor correspondence for which the position error between the global–local feature pair does not obey assumed Gaussian uncertainties.

Once outliers have been detected, the remaining correspondences are labeled as inliers. The inlier set for a given model is then evaluated with the same fitness metric used in Section 5.2. It is possible to quickly test individual models because only three feature correspondences are used in each model. The inlier set with the highest fitness is called the consensus set. This consensus set of inliers is then used in MOGA.

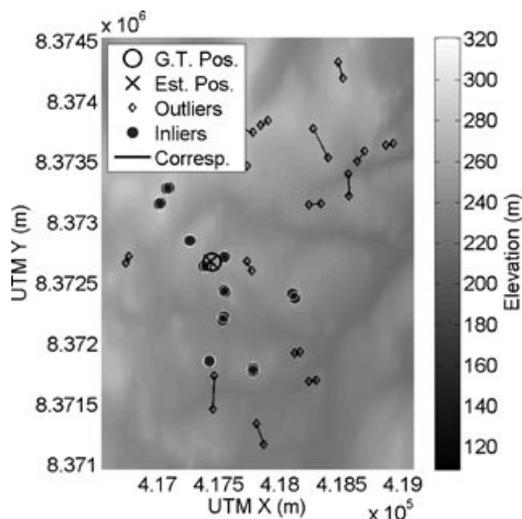


Figure 13. Detected outliers for A08.

RANSAC is designed to operate on frames individually, not as a group tied together by odometry, because relatively large odometry errors might corrupt results when only three feature correspondences are used. Figure 13 shows RANSAC's ability to reject outliers.

7. FIELD DATA

To test the developed localization architecture, a realistic data set was collected from Devon Island, Nunavut, just north of the Houghton Crater at $75^{\circ}22'N$ and $89^{\circ}41'W$. The area's geological features and lack of vegetation make it a unique Mars/Moon analog site that has attracted significant interest from researchers (Lee, 2002; Parnell, Bowden, Osinski, Lee, Green, et al., 2007; Wettergreen, Tompkins, Urmson, Wagner, & Whittaker, 2005).

In total, 37 separate LIDAR scans were collected over 2 weeks using the LIDAR-tripod setup shown in Figure 14. These scans, each made up of millions of points spread over hundreds of meters, were used to test the single-scan localization performance of the MOGA algorithm. The LIDAR also produced image panoramas from an onboard, six-megapixel digital camera to test VIPER and benchmark MOGA.

To investigate MOGA's full capabilities, a long-range rover traverse was simulated by connecting 23 LIDAR scans with real odometry data. This odometry data set was collected by a pushcart outfitted with a variety of rover engineering sensors as shown in Figure 15(a). The cart's 10-km path, seen in Figure 15(b), is comparable in length to the distance traversed by the MERs to date. The orientation of the LIDAR and cart was matched by imaging a target in both systems (see Carle, 2009). The sections to follow describe the full data set in further detail.



Figure 14. LIDAR setup.

7.1. Maps

The global map was obtained from GeoBase, an online repository of digital terrain data covering all of Canada. These 3D maps are produced from stereo image pairs collected from orbiting satellites. The 8.5×12 km map shown in Figure 15(b) was used to localize scans on the 10-km traverse (A01–A23). Other scans (A24–A37) were localized with similar 100-km² global maps for their respective areas. Map resolution and accuracy parameters are defined in the GeoBase map files. The map's x and y resolutions were, respectively, 13 and 24 m. To be more compatible with the developed algorithm, the global map was interpolated at the smaller resolution to form a uniform grid. Therefore, the global map x - y resolution was effectively $L_{xy} = 13$ m. The global map's z resolution was $L_z = 1$ m, and its x - y and z positional uncertainties (i.e., the position uncertainty of a 3D point in the global map) were, respectively, $\sigma_{r_{G_{xy}}} = 14$ m and $\sigma_{r_{G_z}} = 13$ m.

The local maps were collected with an Optech ILRIS3D-ER LIDAR mounted on a pan-tilt unit as seen in Figure 14. Designed as an accurate, long-range mapping sensor, this LIDAR has a maximum range of about 1.5 km in extended-range mode, a beam divergence of 0.00974 deg, and a range accuracy of 7 mm at a distance of 50 m. The vertical and horizontal scan resolutions were set, respectively, to 0.03 and 0.06 deg. The vertical resolution was smaller to compensate for the oblique scanning angle. With these settings, a scan with a field of view of 360 deg in the horizontal

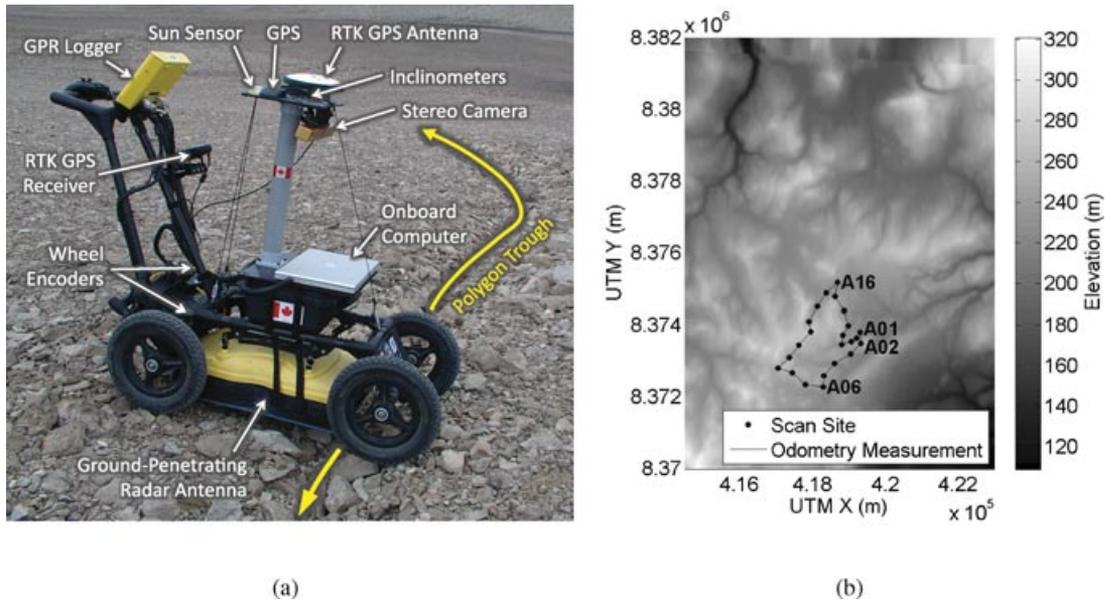


Figure 15. Cart collected odometry measurements between LIDAR scan sites on a 10-km traverse over rough terrain. (a) Cart with sensors. (b) Cart’s path.

and 20 deg in the vertical finished in about 30 min. The position uncertainty of a 3D point in the local map was estimated to be $\sigma_{rLM} = 0.5$ m. Although most scans were quite good, having good visibility of the surrounding terrain as shown in Figure 16(a), some scans were heavily occluded as in Figure 16(b).

7.2. Odometry Measurements

An implementation of VO was used for the odometry measurements in this study. Owing to space restrictions, only

the basic outline of the algorithm will be presented here. In general, this implementation is similar to other stereo VO algorithms based on sparse feature correspondences (Howard, 2008; Konolige et al., 2007; Maimone, Cheng, & Matthies, 2007; Nistér, Naroditsky, & Bergen, 2006).

Stereo pairs of 640×480 images are captured, corrected for lens distortion, and rectified so that they model a pair of pinhole cameras with parallel optical axes. For each stereo pair, SURF features (Bay et al., 2006) are used for both *matching* (across stereo pairs) and *tracking* (over time). Feature descriptor matches between consecutive images

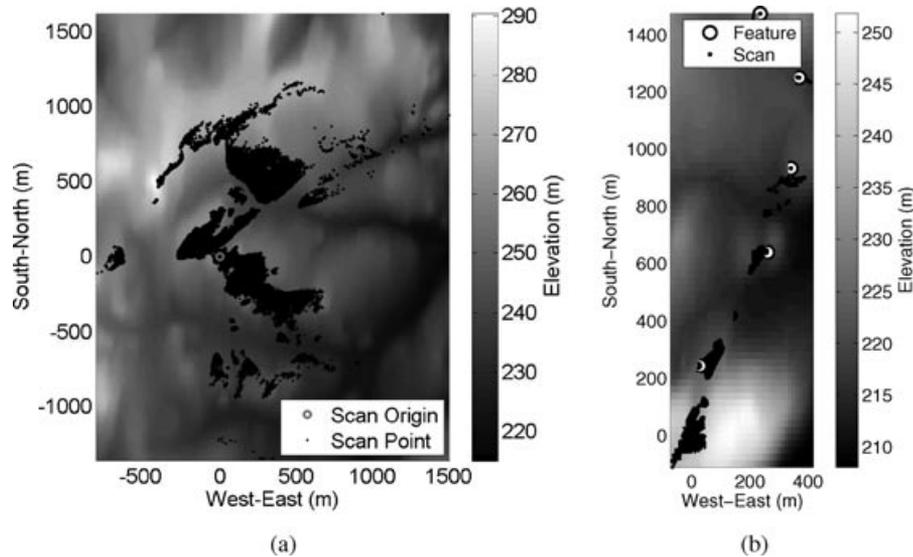


Figure 16. Examples of LIDAR scan with (b) good visibility (A08) and (a) poor visibility due to nearby, occluding terrain (A02).

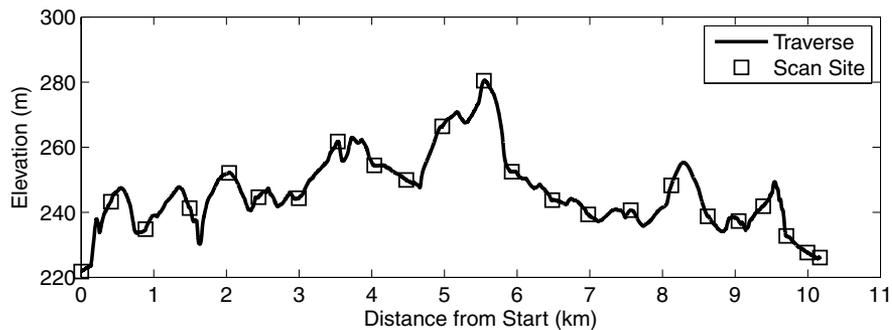


Figure 17. The elevation reported by real-time kinematic (RTK) GPS over the course of the traverse.

are used as candidate tracks, and RANSAC is used to simultaneously reject outlier feature tracks and produce a coarse motion estimate. This implementation used preemptive RANSAC as it will, on average, produce the best set of inlying tracks given a fixed computational budget (Nistér, 2005). The coarse motion estimate is used to initialize an iterative numerical solution for the change in pose between consecutive images. The numerical solution used in this study was an implementation of the sliding window VO described by Konolige et al. (2007). Both the vehicle poses and feature positions are solved for using sparse BA, a maximum likelihood solution for motion and structure given all of the measurements (Hartley & Zisserman, 2000; Triggs, McLauchlan, Hartley, & Fitzgibbon, 2000). To maintain constant computational complexity over the entire traverse, the solution is limited to the feature observations made from a small subset of poses, some of them fixed and some free in the optimization. These results use five poses fixed and four poses free. The cart was also equipped with an inclinometer set to an output rate of 1 Hz. To ensure consistency over the entire traverse, an extended Kalman filter is used to fuse the inclinometer measurements with the VO estimates.

The VO results reported here are not as accurate as those reported by Konolige et al. (2007). However, the comparison is not direct for two reasons. First, Konolige et al. (2007) also integrated heading measurements from a high-quality gyroscope. Second, the performance is dependent on the data set itself. The cart captured 49,413 stereo images over a 10-km traverse through a wide range of different terrain from highly textured rock fields to nearly textureless sandy areas. As shown in Figure 17, the platform experienced significant 3D motion throughout the traverse and, due to lack of suspension on the cart, the video is quite choppy over rough terrain. Despite the limited accuracy of the VO implemented in this paper, it can still be of beneficial use to the MOGA algorithm, as will be demonstrated in later sections.

7.3. Orientation Measurements

Orientation measurements were used to improve the performance of the localization algorithm and were a required

input for the VIPER algorithm. Heading measurements for A01–A25 were obtained from a sun sensor (see Enright et al., 2009), and those from A26–A37 were computed knowing the GPS and local positions of a distant target (see Carle, 2009). Measurements from both methods were assumed to have uncertainty $\sigma_\gamma = 1$ deg. Roll and pitch were effectively measured by leveling the LIDAR with its built-in, two-axis bubble levels. In practice, these parameters would be measured with an inclinometer. The uncertainties in roll and pitch were, respectively, taken to be $\sigma_\alpha = \sigma_\beta = 1$ deg. The covariance matrix for all orientation measurements was therefore $R_{\text{orient}} = (1 \text{ deg } \frac{\pi}{180})^2 \mathbf{1}$.

7.4. Ground Truth

Ground-truth x - y position measurements, $\rho_{x,t}$ and $\rho_{y,t}$, were obtained from a Garmin GPSMAP 76CSx with uncertainty $\sigma_{\rho_{x,t}} = \sigma_{\rho_{y,t}} = 10$ m after averaging measurements for several minutes.

8. RESULTS

This section presents results from field tests and discusses the performance of the developed algorithm. A summary of the various parameters input into the algorithm is presented in Table II.

The algorithm was coded in Matlab R2007a and run on a system with an Intel Core2 2GHz CPU and 2 GB of RAM. With these parameters, about 500 global features were detected within 10 s on the 100-km² global map. Local map feature detection was typically slower and took about 1 min per scan. A low number of features in a frame tended to make it more difficult to find a DARCES solution. A DARCES solution was typically found within 10 min. A single-frame run of MOGA converged in 2 min after about 10 iterations. Most of these 2 min was spent within RANSAC, because all possible models of features were tested. Overall, localization for a single scan from feature detection to pose refinement was completed in under 15 min. With outliers already rejected, a multiframe MOGA run converged within 30 s.

Table II. Summary of key input parameters; all uncertainties correspond to one standard deviation.

Parameter	Value	Section	Description
L_{xy}	13 m	7.1	Global map x, y resolution
L_z	1 m	7.1	Global map z resolution
$\sigma_{rG_{xy}}$	14 m	7.1	Global map x, y uncertainty
σ_{rG_z}	12 m	7.1	Global map z uncertainty
σ_{rLM}	0.5 m	7.1	Local map x, y, z uncertainty
$\sigma_\alpha, \sigma_\beta, \sigma_\gamma$	1 deg	7.3	Roll, pitch, heading uncertainties
n	5 cells	4	Dilation window radius
D_{detect}	60 m	4	Minimum distance between detected features
E_{flat}	0.5 m	4	Feature flatness threshold
$E_{z\text{dev}}$	24 m	5.2	z -deviation filter threshold
$E_\alpha, E_\beta, E_\gamma$	5 deg	5.2	Orientation filter thresholds
N_{top}	5 hypotheses	5.2	Number of top hypotheses for validity check
D_{valid}	50 m	5.2	Maximum distance between valid hypotheses
E_{converge}	10^{-10}	6.6	MOGA convergence threshold
η	0.9	6.6	MOGA line search factor

8.1. Single-Frame MOGA

In this configuration, local frames were localized individually. There was a stochastic aspect to the DARCES algorithm because control points were randomly selected. It was therefore necessary to demonstrate robustness by running 100 randomly seeded trials for each of the 37 local maps.

Figure 18 summarizes single-frame localization performance. Although MOGA outputs full position and orientation estimates, only position performance is discussed because orientation is assumed to be well-known from measurements (e.g., sun sensor, inclinometer). Most trials showed position error well below 50 m, a good result considering that prior position was known to be only within a 100-km² map. This initial map size could be even larger, but the algorithm would be slowed and might eventually cease to produce a DARCES solution as the number of competing hypotheses grows. A simple solution to this problem would be to use a higher resolution global map, which would allow more features to be detected.

Note that some frames produced no DARCES solution. This occurred when features were either out of the LIDAR's range (flat areas; e.g., A10, A13) or too close to the LIDAR (canyons; e.g., A02, A20–A23). In canyon-like settings, nearby hills occlude much of the view, leaving few features for the rover to detect. Furthermore, a nearby detected peak is more likely to be a poor representation of the true peak, which would be occluded at close range (see Figure 7).

MOGA also calculates uncertainties for all output estimates. Figure 19 shows these uncertainties for a sample trial. The uncertainties produced by the algorithm seem to be reasonable in most cases. A few data points lie outside the estimated uncertainty, indicating that the uncertainty in some features' positions was underestimated.

Figure 20 shows how the number of features in a frame determines the output localization error and uncertainty. Position error seems to be weakly, negatively correlated to the number of features in a frame but is more likely a function of the *quality* of features. For example, well-resolved peaks would produce much lower error than flat hilltops. Uncertainty in position is clearly inversely correlated to the number of features. The selection of a scan site with many good-quality features is beneficial to position accuracy and uncertainty.

8.1.1. Performance without Heading

The performance of sample trials with and without heading was compared for each frame. To ensure a fair comparison, the same inputs were given to both test cases including the order for selection of control points. Figure 21 shows that the input of heading measurements into the algorithm can lead to significant improvements. With heading information, the hypothesis search space can be greatly reduced, leading to an improvement in computation time. It also increases the chances of finding a DARCES solution (see frames A01, A24, A25, and A29) because hypotheses interfering with the selection of a solution might be discarded. However, the inclusion of heading does not appear to significantly affect the accuracy of position estimates.

8.1.2. VIPER Comparison

To evaluate the performance of the single-frame MOGA, it was compared to the VIPER algorithm. The VIPER algorithm is the most thoroughly tested algorithm in the literature for the localization problem investigated in this paper. It is therefore considered to be the current state of the art.

In these comparison tests, MOGA and VIPER were provided the same orbital maps and heading

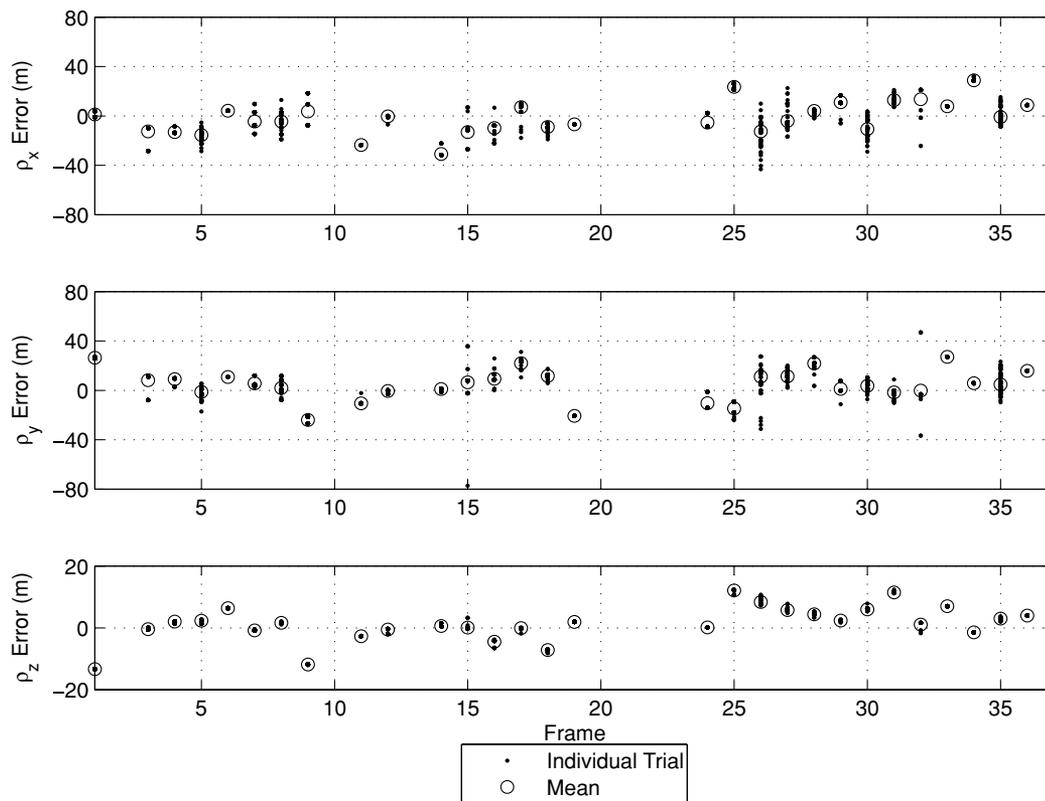


Figure 18. Position errors for 100 single-frame trials on all 37 scans. Mean errors for each frame are also shown.

measurements. The similarity of the input data ensures a fair comparison of the two approaches. The original VIPER implementation is no longer available, and so it was necessary to reimplement VIPER (Appendix B) based on its description by Cozman et al. (2000).

The results of these experiments show that although VIPER was computationally less demanding, MOGA's localization performance was generally superior, as is evident in Figure 22. The difference can primarily be attributed to the use of 3D information in MOGA, because VIPER is unable to distinguish between areas that produce similar 2D skylines. Additionally, VIPER uses information only at the horizontal boundary, whereas MOGA is able to use peaks visible below the horizon, as shown in Figure 23(a).

VIPER outperformed MOGA in a small number of cases, notably for scan A13. MOGA found no solution for this scan because the LIDAR was located in a relatively flat region where good features were out of range. Meanwhile, the lack of occlusions nearby allowed VIPER to detect the horizon many kilometers in the distance. This result suggests that there are situations in which VIPER should be used instead of MOGA, particularly where the nearby terrain is flat but distant features are visible from camera images as depicted in Figure 23.

However, there are still scenarios where neither algorithm performed very well, such as the canyon locations of A02 and A20–A23. In these canyons, only short-range features are visible to either LIDAR or camera due to occlusions, which makes localization much more difficult for both algorithms. Furthermore, the canyon walls were thinner than the resolution of the digital elevation model (DEM) and therefore are not accurately represented in the global map. In the end, a number of common factors limit the performance of both algorithms:

- **DEM quality:** Higher quality and higher resolution DEMs will increase the ability of these algorithms to discern position.
- **The amount of topographic relief in the area:** Areas with little topographic relief do not provide useful information for either algorithm. This may explain the high failure rate observed for this implementation of VIPER compared to the results reported by Cozman et al. (2000); the panoramas gathered in the Atacama desert as part of that work² show large-scale relief along the horizon—much greater relief than exhibited on Devon Island.

²<http://www.cs.cmu.edu/~VIPER/AtacamaMission/>

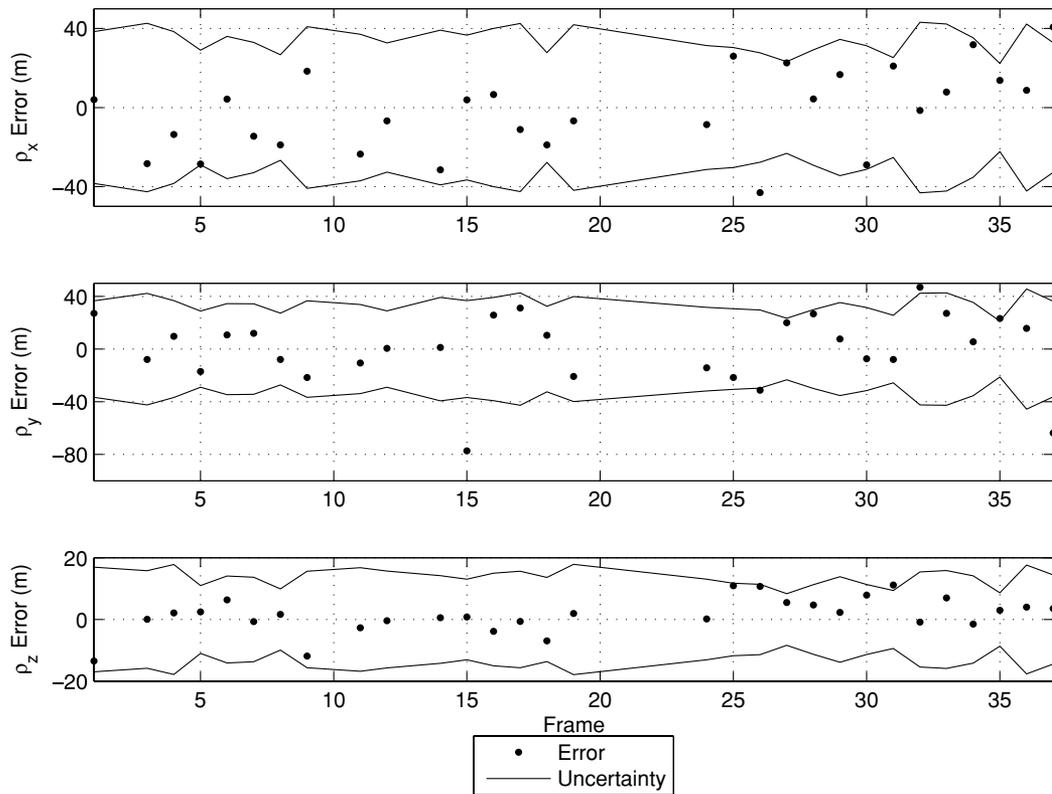


Figure 19. Single-frame localization uncertainties (three standard deviations) for trials with the *worst* radial error. Ground-truth measurement uncertainty is not included in the shown uncertainties.

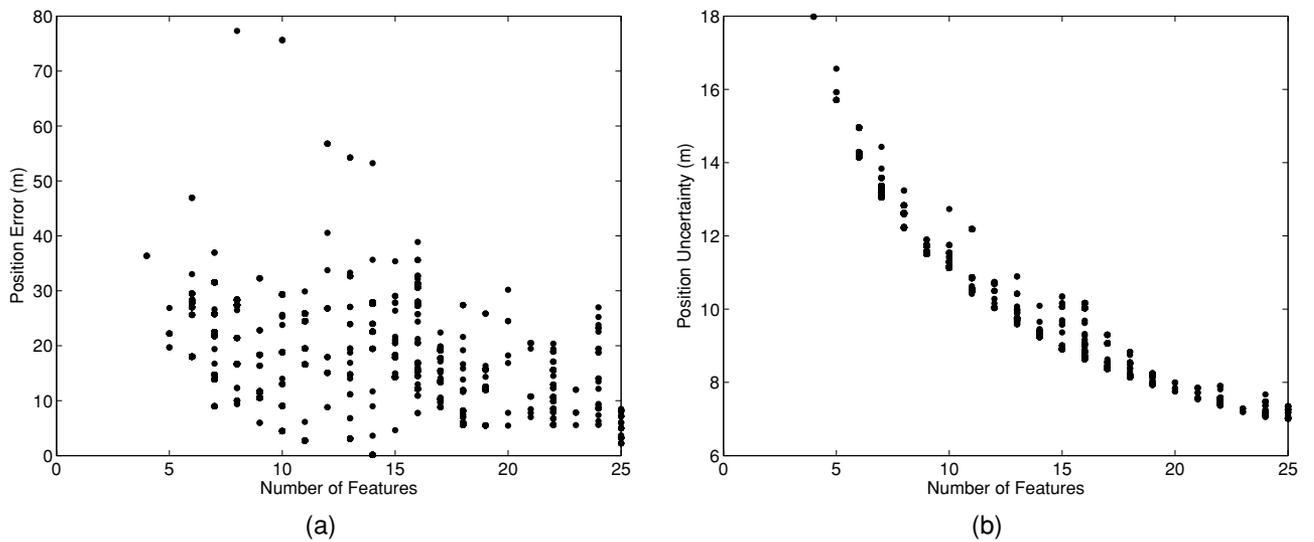


Figure 20. Correlation between number of features and (a) position error and (b) uncertainty.

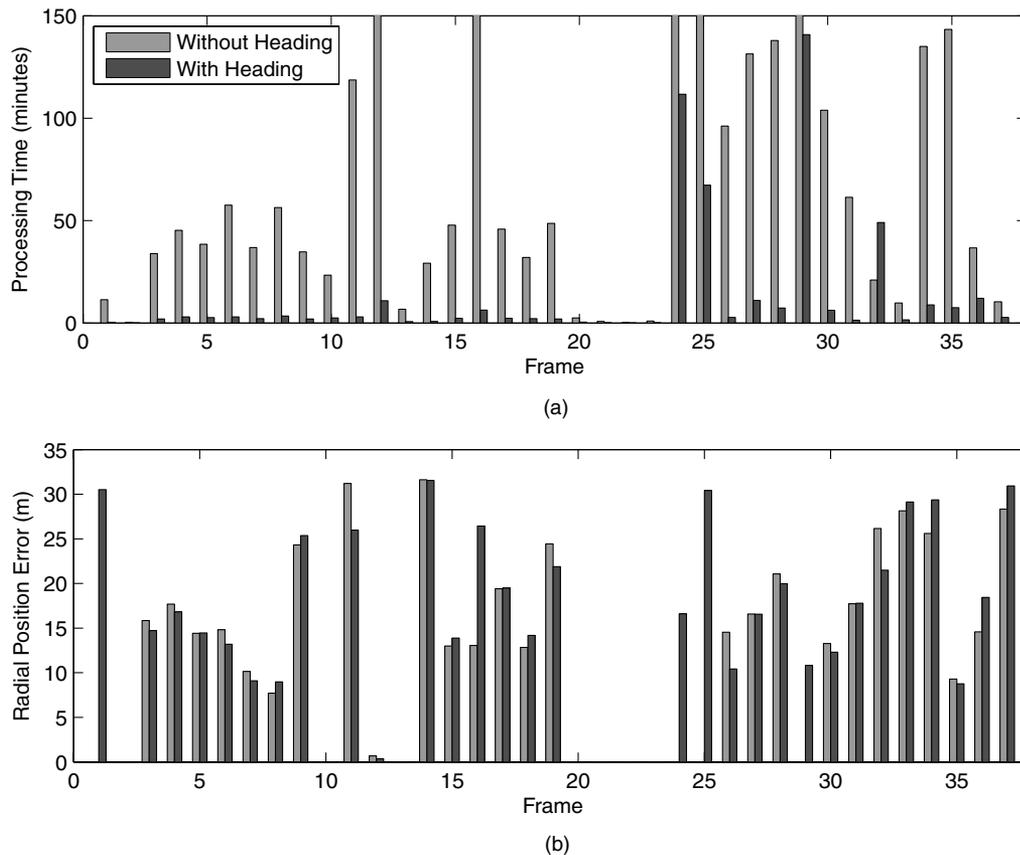


Figure 21. Significance of heading measurements for sample trials: (a) Time required to find a DARCES solution. Bars for frames 12, 16, 24, 25, and 29 were truncated for clarity. (b) MOGA position errors obtained when a DARCES solution was found.

- **The amount of topographic relief within the range of the sensor:** VIPER is limited to using the topographic relief visible on the horizontal boundary, whereas MOGA is limited by the range of the LIDAR.
- **Availability and quality of the platform orientation estimate:** Relatively small errors in the orientation of the platform can cause large differences in the perceived features. Although this may be mitigated through careful attention during data collection, future work in this area should address the problem of uncertainty in orientation.

8.2. Multiframe MOGA

In the multiframe configuration, VO measurements between scans A01 and A23 were available to MOGA. This allowed for a single, large MOGA optimization to be executed. DARCES transformation estimates served as initial guesses for MOGA. However, if a particular frame did not have a DARCES solution, its transformation was estimated using VO to the next-closest, solved frame. As shown in

Figures 24 and 25, MOGA can now produce localization estimates for frames with no DARCES solution using information from all available measurements in the chain of frames. VO estimates are provided in Figure 25 for two cases. One uses the VO algorithm’s output orientation estimates, and the other replaces these with absolute orientation measurements from the sun sensor and inclinometer. It should also be made clear that the VO algorithm was given the rover’s ground-truth pose for frame A01, whereas MOGA used no ground truth whatsoever.

Note the significant improvement in the estimates as more information is given to the estimator. With absolute orientation measurements, VO reduces its error from up to several kilometers to a few hundred meters. With absolute position measurements provided by matched features in MOGA, the error can be further reduced to as low as a few meters.

Multiframe MOGA clearly performs better than VO over long ranges, which is to be expected because MOGA uses additional information (i.e., orbital map and LIDAR scan) to produce its estimates. However, VO information

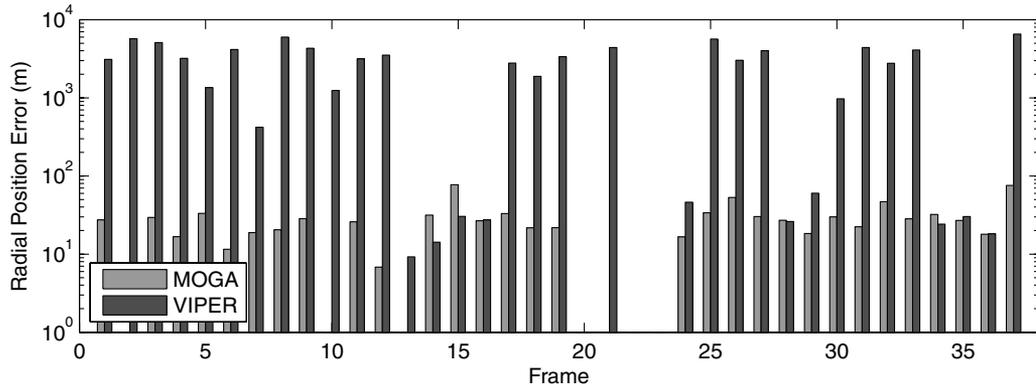
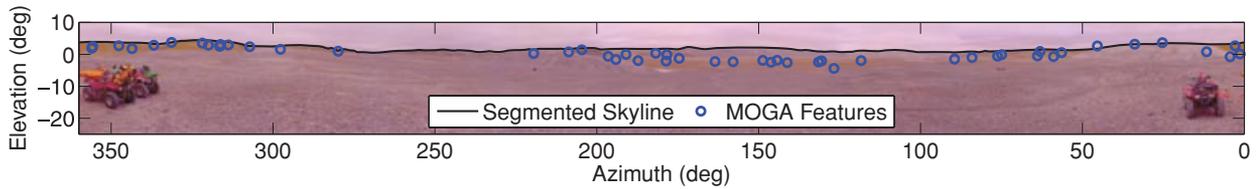
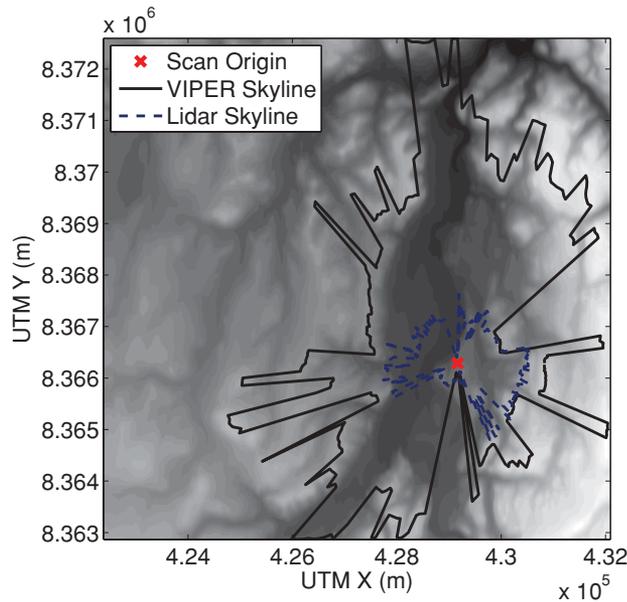


Figure 22. VIPER and MOGA position errors. MOGA errors shown are the worst from all single-frame trials. The vertical axis is on a log scale.



(a)



(b)

Figure 23. A comparison of the data input into VIPER and MOGA for location A37: (a) VIPER’s segmented skyline and MOGA’s features plotted on image panorama. VIPER is unable to use topographic information below the skyline. (b) Horizon of the LIDAR scan compared to VIPER’s rendered horizon. VIPER can use topographic information that is much farther away from the sensor.

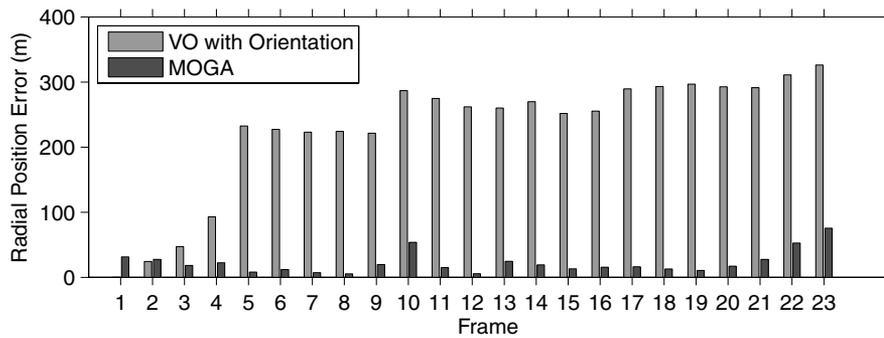


Figure 24. VO estimates compared to multiframe MOGA estimates. VO was given ground truth for frame A01. The error using VO without absolute orientation measurements was very large and was therefore omitted.

can still be useful when used in MOGA. In Figure 26, single-frame MOGA is compared to multiframe MOGA. For the presented single-frame results, position estimates are obtained for frames with no DARCES solution (i.e., A02, A13, A20–23) using VO to the nearest solved frame. The resulting average errors for single-frame and multiframe MOGA are, respectively, 29.6 and 22.1 m. This sig-

nificant improvement can be attributed not only to VO measurements but also to feature sharing between frames. Sharing features allows a given frame to use local features from other nearby frames with common global correspondences to improve estimates. VO will benefit MOGA only if the uncertainties of VO and feature position measurements are correctly modeled. If not, the algorithm will improperly weight measurements and possibly arrive at a poorer solution.

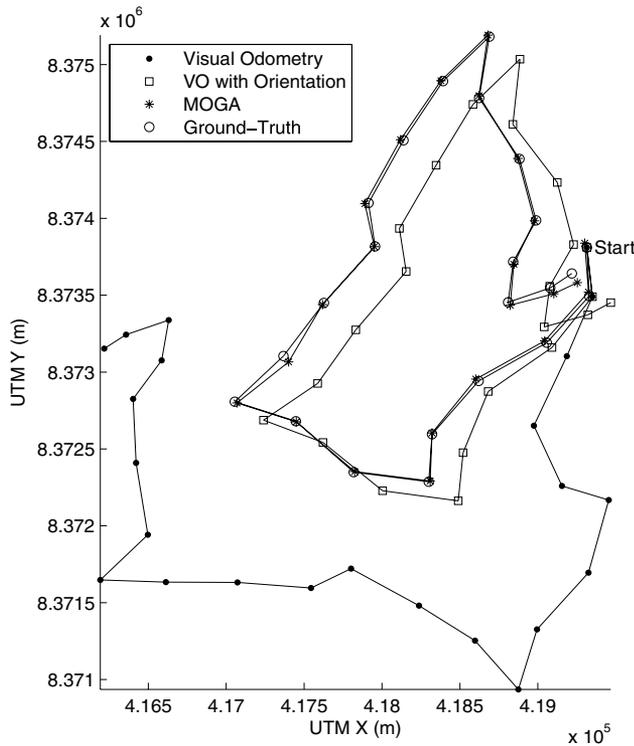


Figure 25. VO, MOGA, and ground-truth positions. Odometry was given ground-truth position and orientation for frame A01, but MOGA was not. Connecting lines do not indicate the intermediate path and serve only as a visual aid.

9. FUTURE WORK

There are many interesting avenues to investigate for future work. A different feature detector, such as a curvature-based detector, could be used to more accurately estimate the positions of features. The main challenge would be in dealing with the occlusions in the LIDAR scans. It would also be interesting to quantitatively understand the effect of orbital map resolution on feature detection and localization accuracy.

For the rover to successfully localize, it must collect a rich set of global and local features with many valid correspondences. The LIDAR should therefore not scan in canyons where nearby hills could occlude distant terrain features. Future research could investigate how to autonomously select scan sites that would be favorable to localization. Once the developed algorithm is coded in a more efficient language, the scan itself could become the bottleneck in the procedure. Therefore, the scan patterns should also be optimized to scan more intelligently (e.g., improving the control of the scan resolution).

DARCES could be sped up in a number of ways as explained by Chen et al. (1999). Odometry information could be used in DARCES for further gains in efficiency and accuracy. The translational odometry data could serve as another filter on position between two frames. If carefully implemented, this would greatly reduce the initial hypothesis search space. Odometry could also allow features to be shared between frames during DARCES. This would

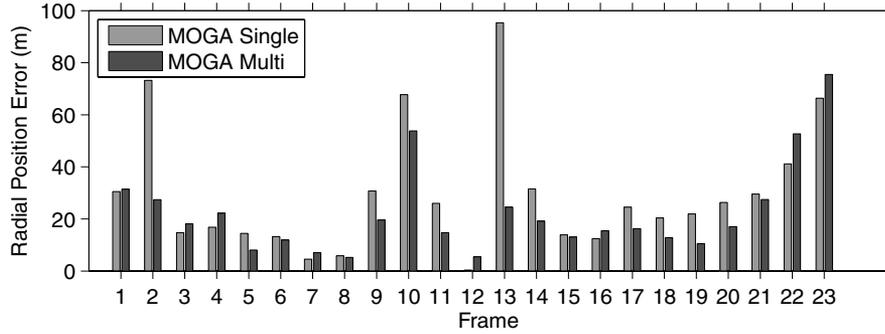


Figure 26. Single-frame MOGA position errors compared to multiframe MOGA errors. Single frame uses odometry to the nearest solved frame when a given frame has no DARCES solution.

be useful for frames with low numbers of features. For example, a frame with only two features would not normally have a DARCES solution. If odometry were available, a third feature could be used from a nearby frame to satisfy the conditions for a solution.

This paper has shown that VIPER and MOGA can complement one another in certain situations. It would therefore be interesting to develop a localization framework that combines the best aspects of the two algorithms. For example, VIPER analyzes images that may contain information many kilometers out of a LIDAR’s range. In contrast, LIDAR scans used by MOGA provide much more detailed information of the nearby terrain and do not depend on lighting conditions.

10. CONCLUSION

This research has produced a number of novel contributions. A global localization technique was developed that matches rover-based LIDAR scans to an orbital elevation map using DARCES feature constellations. MOGA, a multiple-frame, least-squares alignment technique, was designed that uses feature position, orientation, and odometry measurements to refine pose estimates. The architecture was also validated by testing it on a realistic Mars/Moon analog data set from Devon Island, Nunavut. The developed algorithm was tested against the current state of the art, VIPER, and showed significant improvement. With additional work, it is believed that the architecture presented in this paper could be used to autonomously localize a rover over long ranges with high accuracy.

11. APPENDIX A: AXIS-ANGLE CONVERSION

Given a rotation matrix, C , it is desired to convert it to an equivalent rotation about an axis $\hat{a} := [a_1 \ a_2 \ a_3]^T$ over an

angle ω . These are given by

$$\omega := \arccos\left(\frac{\text{trace}(C) - 1}{2}\right), \tag{A1}$$

$$a_1 := \frac{C_{23} - C_{32}}{2 \sin(\omega)}, \tag{A2}$$

$$a_2 := \frac{C_{31} - C_{13}}{2 \sin(\omega)}, \tag{A3}$$

$$a_3 := \frac{C_{12} - C_{21}}{2 \sin(\omega)}. \tag{A4}$$

Therefore, the axis-angle vector representation of C is $\mathbf{a} := \omega \hat{\mathbf{a}}$. Hughes (1986) discusses this procedure in greater detail.

12. APPENDIX B: AN IMPLEMENTATION OF THE VIPER ALGORITHM

This appendix describes an implementation of VIPER, an algorithm that attempts to estimate the position of a rover within a DEM through an exhaustive search. A skyline segmented from a panoramic image is compared to skylines rendered at every place in the map, and the closest match is considered the most likely position. Cozman (1997), Cozman and Krotkov (1997), and Cozman et al. (2000) describe various implementations of VIPER. The latter of these is investigated in this paper for comparison against MOGA. The algorithm assumes that the heading of the platform is well known (from celestial and gravitational observations as in Enright et al., 2009) and that the platform is leveled before collecting the image panorama.

In this section, *altitude* will be used to refer to the height above mean sea level and *elevation* to the angle between the true horizontal plane and a point of interest on the skyline.

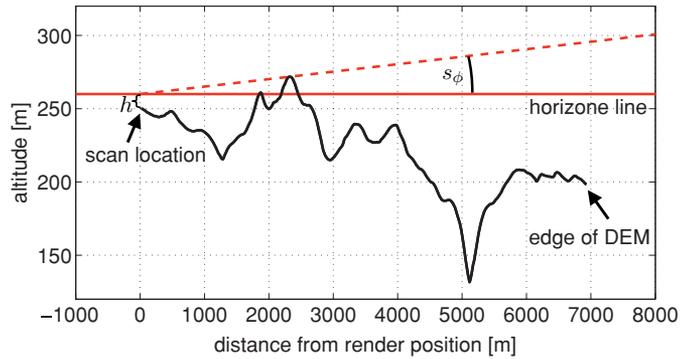
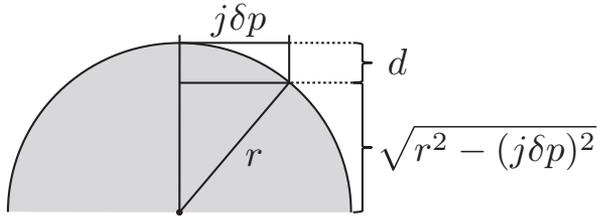


Figure B1. The altitude of points along a line are looked up in the DEM. Altitude samples are corrected for dip due to curvature of the planet. For each azimuth angle, ϕ , the maximum elevation angle between the horizon line, an altitude sample is returned as s_ϕ . (a) The geometry of the altitude correction based on the curvature of the planet. (b) The geometry of the calculation of a single elevation sample.

12.1. Skyline Segmentation

The commercial photo stitching software AutoPano Pro³ was used to generate a panoramic image from images captured at each scan location. In the resulting panorama, horizontal pixels correspond to azimuth angle and vertical pixels (up from the center of the image) correspond to elevation angle.

Each skyline profile is manually segmented from the panoramic image. The resulting curve is sampled to render a single elevation angle for each azimuth angle. Let m_ϕ be the elevation sample at azimuth ϕ . The segmented skyline is sampled at $\phi = 0 \dots 359$ deg, and these samples are stacked into a column, \mathbf{m} :

$$\mathbf{m} := \begin{bmatrix} m_0 \\ \vdots \\ m_{359} \end{bmatrix}.$$

12.2. Skyline Rendering

This section describes the skyline rendering algorithm. Let \mathbf{p} be a position of interest in the DEM:

$$\mathbf{p} := \begin{bmatrix} x \\ y \end{bmatrix},$$

where x and y are the UTM easting and northing coordinates, respectively. The DEM may be modeled as a function, $z(\mathbf{p})$, which maps positions to altitudes. The skyline-rendering algorithm produces skyline elevations sampled in 1-deg azimuth increments. Each elevation sample, s_ϕ , is rendered separately.

To determine the elevation seen at a particular azimuth angle, the DEM is sampled along a line in the viewing direction. Given the azimuth angle, ϕ , the maximum number

of samples, N , and a step size, δp , a set of sampled altitudes, $S(\phi, \mathbf{p}, \delta p, N)$, is built from the DEM:

$$S(\phi, \mathbf{p}, \delta p, N) = \{a_j | j = 0 \dots N\},$$

where

$$a_j = z\left(\mathbf{p} + j \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix} \delta p\right) - d(j\delta p).$$

The step size is set based on the resolution of the DEM, the number of steps is set to ensure that the DEM is sampled to its edge, and the function $d(\cdot)$ is a dip correction for the curvature of the planet. Figure B1(a) shows the geometry of the dip correction. For an individual altitude sample, a_j , the horizontal distance from the viewing location is $j\delta p$. From this distance the dip of the horizon may be computed as

$$d(j\delta p) = r - \sqrt{r^2 - (j\delta p)^2},$$

where r is the radius of the planet.

Finally, using the height of the sensor above the ground, h , each altitude sample, a_j , is converted to an elevation angle, e_j :

$$e_j = \text{atan2}(a_j - (a_0 + h), j\delta p).$$

The maximum elevation, s_ϕ , determines the elevation at the horizon:

$$s_\phi = \max_j(e_j).$$

Figure B1(b) depicts the geometry of the elevation calculation.

A value of s_ϕ is calculated for azimuth angles $\phi = 0 \dots 359$ deg, and these are stacked into a column, \mathbf{s} :

$$\mathbf{s}(\mathbf{p}) = \begin{bmatrix} s_0 \\ \vdots \\ s_{359} \end{bmatrix}.$$

³Available at <http://www.autopano.net>

To speed up the algorithm, the skylines are prerendered at every grid cell in the DEM. The rendering can take a long time (days for each DEM), but it has to be performed only once, and it speeds up the position estimation algorithm by several orders of magnitude.

12.3. Position Estimation

The position estimation algorithm uses a Bayesian approach to determine the most likely position in the map given the skyline. The estimator computes the posterior probability of the rover being at every place in the map, $p(\mathbf{p}|\mathbf{m})$. This value is expanded using Bayes' rule:

$$p(\mathbf{p}|\mathbf{m}) = \frac{p(\mathbf{m}|\mathbf{p})p(\mathbf{p})}{p(\mathbf{m})}. \quad (\text{B1})$$

The measurement model is

$$\mathbf{m} = \mathbf{s}(\bar{\mathbf{p}}) + \delta\mathbf{m}, \quad (\text{B2})$$

where $\bar{\mathbf{p}}$ is the true location of the rover and the measurement noise, $\delta\mathbf{m}$, is assumed to be drawn from a zero-mean Gaussian distribution:

$$\delta\mathbf{m} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}).$$

This model is simplified by assuming that the form of \mathbf{R} is

$$\mathbf{R} = \sigma^2 \mathbf{1},$$

where σ is the measurement's standard deviation. Under this model, the likelihood of \mathbf{m} given \mathbf{p} is

$$p(\mathbf{m}|\mathbf{p}) = \frac{1}{\sqrt{\det(2\pi\mathbf{R})}} \exp\left(-\frac{1}{2}(\mathbf{m} - \mathbf{s}(\mathbf{p}))^T \mathbf{R}^{-1}(\mathbf{m} - \mathbf{s}(\mathbf{p}))\right). \quad (\text{B3})$$

This observation likelihood equation is different from the one published in Cozman et al. (2000), which uses $\sqrt{2\pi}\sigma^2$ as the denominator in the fraction on the left. Using this denominator, $p(\mathbf{m}|\mathbf{p})$ is no longer a valid probability density function. Equation (B3) has been used in this implementation as it is true to the noise model.

The prior likelihood, $p(\mathbf{p})$, encapsulates prior knowledge of the rover's position. In this paper's implementation, which evaluates a single-image panorama against a DEM [Cozman et al. (2000) refers to this as the *dropoff problem*], every position in the map is assumed to be equally likely.

Finally, the likelihood of the skyline measurement, $p(\mathbf{m})$, is expanded as

$$p(\mathbf{m}) = \sum_{\rho \in K} p(\mathbf{m}|\rho)p(\rho), \quad (\text{B4})$$

where ρ is a position in the DEM and K is the set of all such positions.

Rather than finding the single maximum-likelihood solution for Eq. (B1), the VIPER algorithm uses an interval-

valued model to represent imprecision in the model equations. Specifically, the algorithm considers a range of values of σ in Eq. (B3), $[\sigma_l, \sigma_u]$. Within this range, Eq. (B3) is minimized or maximized at σ_l, σ_u , or $\sigma^2 = [\mathbf{m} - \mathbf{s}(\mathbf{p})]^T [\mathbf{m} - \mathbf{s}(\mathbf{p})]$. Let the terms $\underline{p}(\mathbf{m}|\mathbf{p})$ and $\bar{p}(\mathbf{m}|\mathbf{p})$ represent the minimized and maximized values of Eq. (B3).

From the bounds of Eq. (B3), an upper bound, $\bar{p}(\mathbf{p}|\mathbf{m})$, and a lower bound, $\underline{p}(\mathbf{p}|\mathbf{m})$, on the posterior probability may be defined:

$$\underline{p}(\mathbf{p}|\mathbf{m}) = \frac{\underline{p}(\mathbf{m}|\mathbf{p})p(\mathbf{p})}{\underline{p}(\mathbf{m}|\mathbf{p})p(\mathbf{p}) + \sum_{\rho \neq \mathbf{p}} \bar{p}(\mathbf{m}|\rho)p(\rho)}, \quad (\text{B5})$$

$$\bar{p}(\mathbf{p}|\mathbf{m}) = \frac{\bar{p}(\mathbf{m}|\mathbf{p})p(\mathbf{p})}{\bar{p}(\mathbf{m}|\mathbf{p})p(\mathbf{p}) + \sum_{\rho \neq \mathbf{p}} \underline{p}(\mathbf{m}|\rho)p(\rho)}. \quad (\text{B6})$$

From these bounds on the posterior, the estimate \mathbf{p}^* is defined to be the position in the DEM that maximizes $\underline{p}(\mathbf{p}|\mathbf{m})$. This estimate defines a region, R ,

$$R(\mathbf{p}^*) = \{\mathbf{p} | \bar{p}(\mathbf{p}|\mathbf{m}) \geq \underline{p}(\mathbf{p}^*|\mathbf{m})\}, \quad (\text{B7})$$

of all estimates that are essentially equivalent under the interval-valued model. The number of points in R is a measure of the confidence of the estimator, with large regions corresponding to low confidence.

To determine σ_l and σ_u in the original VIPER study, they compared the skyline measurements from a calibration image panorama to n elevation measurements gathered with a theodolite, setting

$$\sigma_l = \sqrt{(n-1)s^2/\chi_{n-1,2.5\%}^2}, \quad (\text{B8a})$$

$$\sigma_u = \sqrt{(n-1)s^2/\chi_{n-1,97.5\%}^2}, \quad (\text{B8b})$$

where s is the sample standard deviation and $\chi_{n,\alpha}^2$ is the χ -square distribution with n degrees of freedom and a significance level of α . No theodolite measurements were collected with this data set so σ_l and σ_u are determined by a comparison of the segmented skylines to the rendered skylines for a subset of the data. For each panorama, 20% of the other panoramas are randomly selected to compute σ_l and σ_u .

12.4. Putting the Pieces Together

Using the definitions in the above sections, the VIPER algorithm may be described step by step for a single panorama:

1. Build the image panorama.
2. Segment the skyline in the image, sampling the curve at 1-deg increments to produce a vector of elevation measurements, \mathbf{m} .
3. Randomly select 20% of the other panoramas to calculate σ_l and σ_u using Eqs. (B8a) and (B8b).
4. Evaluate the upper and lower bound measurement likelihoods, $\bar{p}'(\mathbf{m}|\mathbf{p})$ and $\underline{p}'(\mathbf{m}|\mathbf{p})$, at every place in

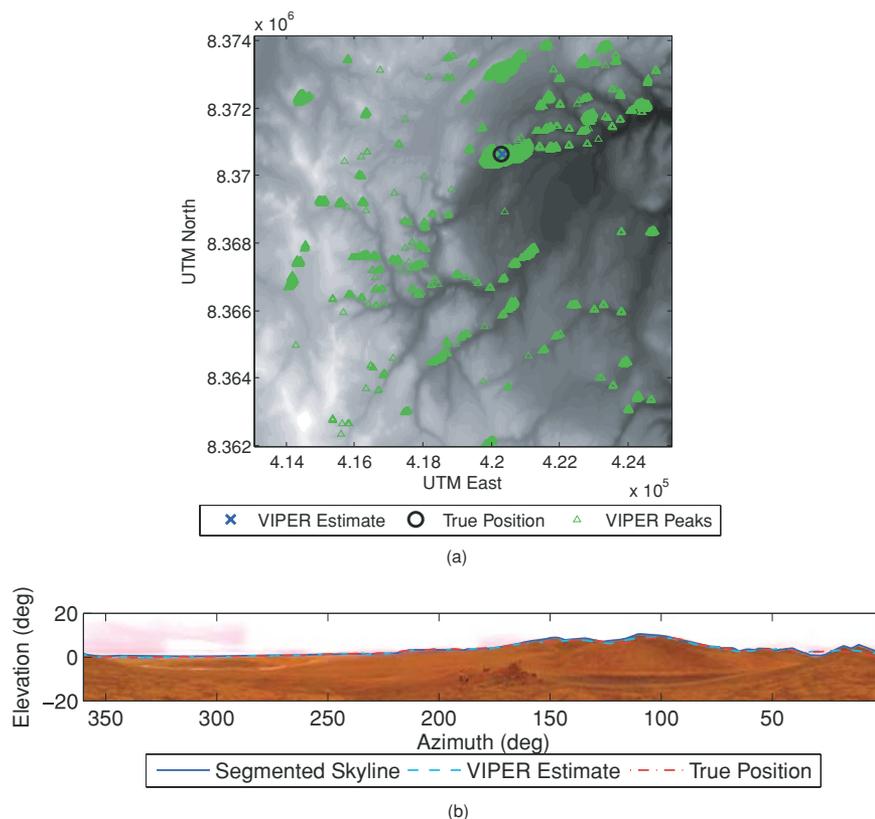


Figure B2. Typical VIPER results shown for location A34: (a) Triangles represent the locations in R under the interval-valued noise model. (b) The segmented skyline, the skyline rendered at the true position, and the skyline rendered at the estimated position are plotted on the image panorama.

- the DEM using Eq. (B3) at $\sigma = \sigma_l$, $\sigma = \sigma_u$, and $\sigma^2 = [\mathbf{m} - \mathbf{s}(\mathbf{p})]^T [\mathbf{m} - \mathbf{s}(\mathbf{p})]$. If this last value of σ is outside of the region $[\sigma_l, \sigma_u]$, it is not used.
5. Evaluate the upper and lower bound posterior probabilities, $p(\mathbf{p}|\mathbf{m})$ and $\bar{p}(\mathbf{p}|\mathbf{m})$, at every place in the DEM using Eqs. (B5) and (B6).
 6. Find $\mathbf{p}^* = \max_{\mathbf{p}}(p(\mathbf{p}|\mathbf{m}))$.
 7. Evaluate the confidence of the estimate by finding the region of equivalent estimates, R , using Eq. (B7).

A typical result is shown in Figures B2(a) and B2(b).

ACKNOWLEDGMENTS

Special thanks to Joseph Bakambu for helpful discussions on feature matching techniques. We would also like to thank the Mars Institute, the Canadian Space Agency, and the Canada Foundation for Innovation for making our field trials on Devon Island possible. Finally, thanks to Optech for providing us with a great sensor.

REFERENCES

- Arun, K., Huang, T., & Blostein, S. (1987). Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5), 698–700.
- Bakambu, J. N., Allard, P., & Dupuis, E. (2006, June). 3D terrain modeling for rover localization and navigation. In *Proceedings of the the 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*, Quebec City, Canada (p. 61). Washington, DC: IEEE Computer Society.
- Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded up robust features. *European Conference on Computer Vision*, 1, 404–417.
- Behar, A., Matthews, J., Carsey, F., & Jones, J. (2004, March). NASA/JPL Tumbleweed polar rover. In *Proceedings of IEEE Aerospace Conference, Big Sky, MT* (vol. 1).
- Besl, P., & McKay, H. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 239–256.
- Björck, Å. (1996). *Numerical methods for least squares problems*. Philadelphia, PA: Society for Industrial Mathematics.

- Carle, P. J. (2009). Long-range rover localization by matching lidar scans to orbital elevation maps. Master's thesis, University of Toronto.
- Chen, C.-S., Hung, Y.-P., & Cheng, J.-B. (1999). RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11), 1229–1234.
- Chin, G., Brylow, S., Foote, M., Garvin, J., Kasper, J., Keller, J., Litvak, M., Mitrofanov, I., Paige, D., Raney, K., Robinson, M., Sanin, A., Smith, D., Spence, H., Spudis, P., Stern, S. A., & Zuber, M. (2007). Lunar Reconnaissance Orbiter overview: The instrument suite and mission. *Space Science Reviews*, 129(4), 391–419.
- Cozman, F. (1997). Decision making based on convex sets of probability distributions: Quasi-Bayesian networks and outdoor visual position estimation. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- Cozman, F., & Krotkov, E. (1997, April). Automatic mountain detection and pose estimation for teleoperation of lunar rovers. In *International Conference on Robotics and Automation*, Albuquerque, NM (vol. 3, pp. 2452–2457).
- Cozman, F., Krotkov, E., & Guestrin, C. (2000). Outdoor visual position estimation for planetary rovers. *Autonomous Robots*, 9(2), 135–150.
- Enright, J., Furgale, P., & Barfoot, T. (2009, March). Sun sensing for planetary rover navigation. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT.
- Fischler, M., & Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.
- Guinn, J. (2001, February). Mars surface asset positioning using in-situ radio tracking. In *Proceedings of the 11th Annual AAS/AIAA Space Flight Mechanics Meeting*, Santa Barbara, CA (pp. 45–53).
- Haralick, R., & Shapiro, L. (1992). *Computer and robot vision*, Vol. 1. Boston, MA: Addison-Wesley.
- Hartley, R. I., & Zisserman, A. (2000). *Multiple view geometry in computer vision*. Cambridge, UK: Cambridge University Press.
- Horn, B. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4), 629–642.
- Howard, A. (2008, September). Real-time stereo visual odometry for autonomous ground vehicles. In *IEEE/RSJ Intelligent Conference on Intelligent Robots and Systems, 2008. IROS 2008, Nice, France* (pp. 3946–3952).
- Hughes, P. (1986). *Spacecraft attitude dynamics*. New York: Wiley.
- Johnson, A. (1997). *Spin-images: A representation for 3-D surface matching*. Pittsburgh, PA: Carnegie Mellon University.
- Johnson, A., & Montgomery, J. (2008, March). Overview of terrain relative navigation approaches for precise lunar landing. In *2008 IEEE Aerospace Conference, Big Sky, MT* (pp. 1–10).
- Johnston, M., Graf, J., Zurek, R., & Eisen, H. (2005, March). The Mars Reconnaissance Orbiter mission. In *IEEE Conference on Aerospace, Big Sky, MT* (pp. 1–18).
- Konolige, K., Agrawal, M., & Sola, J. (2007, November). Large scale visual odometry for rough terrain. In *Proceedings International Symposium on Robotics Research*, Hiroshima, Japan.
- Lee, P. (2002). Mars on Earth: The NASA Haughton–Mars Project. *Ad Astra: Magazine of the National Space Society*, 14(3), 12–17.
- Li, R., Archinal, B. A., Arvidson, R. E., Bell, J., Christensen, P., Crumpler, L., and Des Marais, D. J., Di, K., Duxbury, T., Golombek, M., Grant, J., Greeley, R., Guinn, J., Johnson, A., Kirk, R. L., Maimone, M., Matthies, L. H., Malin, M., Parker, T., Sims, M., Thompson, S., W. Squyres, S. W., & Soderblom, L. A. (2006). Spirit rover localization and topographic mapping at the landing site of Gusev Crater, Mars. *Journal of Geophysical Research*, 111(E2), E02506.
- Li, R., Arvidson, R. E., Di, K., Golombek, M., Guinn, J., Johnson, A., Maimone, M., Matthies, L. H., Malin, M., Parker, T., Squyres, S. W., & Watters, W. A. (2007). Opportunity rover localization and topographic mapping at the landing site of Meridiani Planum, Mars. *Journal of Geophysical Research*, 112(E2), E02590.
- Li, R., Di, K., & Howard, A. B. (2007). Rock modeling and matching for autonomous long-range Mars rover localization. *Journal of Field Robotics*, 24(3), 187–203.
- Li, R., Di, K., Hwangbo, J., & Chen, Y. (2007, June). Integration of orbital and ground images for enhanced topographic mapping in Mars landed missions. In *Proceedings of the Annual NASA Science Technology Conference (NTSC)*, College Park, MD.
- Li, R., Squyres, S. W., & Arvidson, R. E. (2005). Initial results of rover localization and topographic mapping for the 2003 Mars Exploration Rover mission. *Photogrammetric Engineering & Remote Sensing*, 71(10), 1129–1142.
- Lowe, D. G. (1999, September). Object recognition from local scale-invariant features. In *International Conference on Computer Vision, Corfu, Greece* (vol. 2, p. 1150).
- Mahalanobis, P. (1936, April). On the generalized distance in statistics. In *Proceedings of the National Institute of Sciences India, Calcutta, India* (vol. 2, pp. 49–55).
- Maimone, M., Cheng, Y., & Matthies, L. (2007). Two years of visual odometry on the Mars exploration rovers. *Journal of Field Robotics*, 24(3), 169–186.
- McEwen, A. S., Eliason, E. M., Bergstrom, J. W., Bridges, N. T., Hansen, C. J., Delamere, W. A., Grant, J. A., Gulick, V. C., Herkenhoff, K. E., Keszthelyi, L., Kirk, R. L., Mellon, M. T., Squyres, S. W., Thomas, N., & Weitz, C. M. (2002, March). HiRISE—The High Resolution Imaging Science Experiment for Mars Reconnaissance Orbiter. In *Lunar and Planetary Science XXXIII*, Houston, TX.
- Nimelman, M., Tripp, J., Allen, A., Hiemstra, D., & McDonald, S. (2006, April). Spaceborne scanning lidar system (SLS) upgrade path. In *Proceedings of SPIE, Orlando, FL* (vol. 6201, p. 62011V).

- Nistér, D. (2005). Preemptive RANSAC for live structure and motion estimation. *Machine Vision and Applications*, 16(5), 321–329.
- Nistér, D., Naroditsky, O., & Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1), 3–20.
- Olson, C. F., Matthies, L. H., Schoppers, M., & Maimone, M. W. (2003). Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4), 215–229.
- Parnell, J., Bowden, S. A., Osinski, G. R., Lee, P., Green, P., Taylor, C., & Baron, M. (2007). Organic geochemistry of impactites from the Haughton impact structure, Devon Island, Nunavut, Canada. *Geochimica et Cosmochimica Acta*, 71(7), 1800–1819.
- Shotwell, R. (2005). Phoenix—the first Mars Scout mission. *Acta Astronautica*, 57(2–8), 121–134.
- Smith, D. E., Zuber, M. T., Frey, H. V., Garvin, J. B., Head, J. W., Muhleman, D. O., Pettengill, G. H., Phillips, R. J., Solomon, S. C., Zwally, H. J., Banerdt, W. B., Duxbury, T. C., Golombek, M. P., Lemoine, F. G., Neumann, G. A., Rowlands, D. D., Aharonson, O., Ford, P. G., Ivanov, A. B., Johnson, C. L., McGovern, P. J., Abshire, J. B., Afzal, R. S., & Sun, X. (2001). Mars Orbiter Laser Altimeter—Experiment summary after the first year of global mapping of Mars. *Journal of Geophysical Research*, 106(E10), 689–722.
- Sun, Y., Paik, J., Koschan, A., Page, D. L., & Abidi, M. A. (2003). Point Fingerprint: A new 2-D object representation scheme. *IEEE Transactions on Systems, Man, and Cybernetics*, 33(4), 712–717.
- Taati, B., Bondy, M., Jasiobedzki, P., & Greenspan, M. (2007, October). Variable dimensional local shape descriptors for object recognition in range data. In *IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil (pp. 1–8).
- Triggs, B., McLauchlan, P., Hartley, R., & Fitzgibbon, A. (2000). Bundle adjustment—A modern synthesis. In *Vision algorithms: Theory and practice*, Lecture Notes in Computer Science (pp. 153–177). Berlin: Springer.
- Vandapel, N., Donamukkala, R. R., & Hebert, M. (2006). Unmanned ground vehicle navigation using aerial lidar data. *International Journal of Robotics Research*, 25(1), 31–51.
- van den Boomgaard, R., & van Balen, R. (1992). Methods for fast morphological image transforms using bitmapped binary images. *CVGIP: Graphical Models and Image Processing*, 54(3), 252–258.
- Wettergreen, D., Tompkins, P., Urmson, C., Wagner, M., & Whittaker, W. (2005). Sun-synchronous robotic exploration: Technical description and field experimentation. *International Journal of Robotics Research*, 24(1), 3–30.