A Comparison of Global Localization Algorithms for Planetary Exploration

Paul Furgale, Pat Carle, and Timothy D. Barfoot

***Abstract*— Global localization of a planetary-exploration rover in the absence of a satellite-based global positioning system (GPS) is still an open problem. Although a satellite network is not available for localization around any near-term exploration targets, topographic maps derived from satellite imagery are available. This has spurred the development of several algorithms that perform global localization by matching data collected from onboard sensors to a global digital elevation map (DEM). This paper reviews two of these algorithms—Multiple-frame Odometry-compensated Global Alignment (MOGA) and VIsual Position Estimation for Rovers (VIPER)—and compares their performance on a common dataset, collected in a planetary analog environment. The comparison demonstrates the common factors limiting the performance of these algorithms, but also highlights the benefits and drawbacks of each method. Overall, the MOGA algorithm performed significantly better; however, running both algorithms is seen to be the best option as the computational cost of VIPER is low and it may succeed in some situations wherein MOGA will fail.**

I. INTRODUCTION

An important goal for future generations of planetary exploration rovers will be to explore sites hundreds of kilometers away from their landers [1]. 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 [2], descent trajectory modeling, and by comparing orbital to ground-camera imagery [3]. After 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 it can be inaccurate in high-slip terrain [4]. While computer vision techniques, such as VO and BA, can reduce the growth rate of error, they provide no bound on the absolute error as no measurements are linked to a global coordinate frame [5].

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.

A number of autonomous global localization algorithms have been proposed to address this problem. Radio localization was used produce the initial position estimate for the MERs [6]. In-situ orbiter tracking can reduce the uncertainty in the position estimate to below a kilometer after two passes and in the range of tens of meters after several days. However, this requires the rover to sit still and wait for the satellite to pass overhead. Attitude estimation based on celestial measurements has been successfully deployed on the MERs [7] and there have been several attempts to use the same measurements to perform position estimation [8], [9], [10]. While the results are promising, the technique has never been vetted on a significant dataset. The most successful work has involved matching imagery collected on the ground with satellite imagery [11], [12], [13]. The concept is similar to techniques used for precise spacecraft landing [2], but comparing maps built by orbiting sensors to data collected

P. Furgale, Pat Carle and T.D. Barfoot are with the University of Toronto Institute for Aerospace Studies, 4925 Dufferin Street, Toronto, Ontario, Canada M3H 5T6 {paul.furgale, pat.carle, tim.barfoot}@utoronto.ca

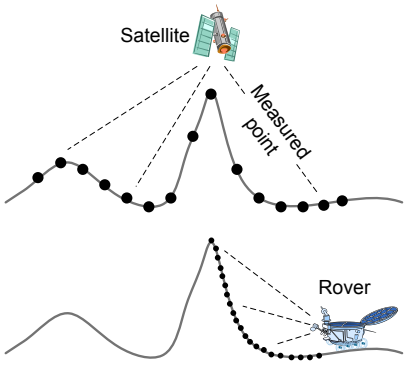

Fig. 1. Matching low-resolution orbital data (top) to high-resolution data collected on the surface is difficult because of differing viewpoints during data collection.

Fig. 2. Data for this study was collected with the Optech ILRIS3D-ER lidar. Long range lidar scans and panoramic images were collected in a planetary analogue setting on Devon Island, in the Canadian High Arctic.

from the ground presents significant challenges as the data is collected from nearly-perpendicular viewpoints (Figure 1).

The contribution of this paper is to compare two algorithms that have been subjected to significant field testing: Visual Position Estimation for Rovers (VIPER) [12] and Multi-frame Odometry-compensated Global Alignment (MOGA) [13]. The comparison is made based on a dataset containing measurements from 40 unique locations on Devon Island, a Mars/Moon analogue site in Nunavut, Canada. The most important of these measurements are long-range lidar scans used in MOGA, and image panoramas used in VIPER, collected simultaneously using the Optech ILRIS3D-ER lidar shown in Figure 2. Localization is accomplished by comparing these to a 100km^2 digital elevation map (DEM) acquired from satellites. Groundtruth positioning was obtained from GPS.

The basic premises of MOGA and VIPER are reviewed in Section II. Section III describes the test site and the data used in this study. Section IV presents the results and lessons learned, and finally Section V provides the conclusion.

II. GLOBAL POSITION ESTIMATION

This section briefly develops the MOGA and VIPER algorithms to be compared.

A. Multi-Frame Odometry compensated Global Alignment

The MOGA algorithm is designed to autonomously, globally localize a rover by matching features detected from a 3D orbital DEM and rover-based 3D lidar scans. The accuracy and efficiency of the algorithm can be enhanced with visual odometry, and inclinometer/sun-sensor orientation measurements. To provide a direct comparison to VIPER (which does not incorporate these extra measurements), a simplified, single-scan version of the algorithm will be presented here. For a more detailed derivation see [13].

1) Feature Detection: Features must first be detected from the DEM (the *global features*) and from the rover's lidar map (the *local features*). Generally, the most prominent features common to both maps are topographic peaks. These peaks are detected using a local maxima detector based on morphological dilation[1] [14], [15]. Morphological dilation replaces lower grid values with neighboring higher grid values, effectively blurring out low elevations. 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.

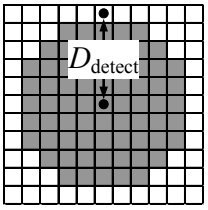

Fig. 3. Pixelated circle with $n = 5$.

The dilation window is chosen to be a pixelated circle (Figure 3) to make the window's coverage more uniform 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. On a 100km^2 orbital map with resolution of 20m, several hundred global features could be detected.

A good uncertainty model is necessary to correctly assess the quality of individual features and global-local matches. Global feature positional uncertainty is assumed equal to the position uncertainty of a measured 3D point in the global map. The positional uncertainty of a measured 3D point from the lidar is comparatively very small (<1m) even for measurements far from the lidar's origin. However, the effect of occlusions will dominate the uncertainty in local feature measurements. Local feature uncertainty is therefore better estimated by the size of a local feature, D_{detect}.

2) Feature Matching: The feature matching methodology is based on the DARCES algorithm [16]. Hypothesized correspondences must first be generated between global and local features. A *hypothesis* is defined as a group of possible correspondences between three unique local features, called *control points*, and three unique global features. 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. A hypothesis is generated if the distances between three global features are similar to the distances between three local features. This test can be quantified knowing each feature's position uncertainty.

The validity of a hypothesis is evaluated based on the transformation it produces between the global and local frame. This transformation can be obtained using a least-squares point-alignment algorithm to align the three global and three local features that comprise a hypothesis [17].

To improve the efficiency and robustness of DARCES, hypotheses are first screened with a number of simple tests. For example, a hypothesis is discarded if the hypothesized rover orientation does not agree well with the measured orientation (i.e., from sun sensor, inclinometer [18]). 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. 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 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, and the corresponding points from the global map:

$$f_i := -\frac{1}{N_{\text{ref}}} \sum_{j=1}^{N_{\text{ref}}} \left| z_{i,j}^R - z_{i,j}^G \right|, \tag{1}$$

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 xy-position of the transformed reference point j. The negative is applied so that a low error corresponds to a high fitness.

Once a fitness is associated with each hypothesis, a search is made for a group of high-fitness hypotheses that have

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

similar position estimates. The hypothesis with the highest fitness in this group is then returned along with its associated transformation estimate and global-local feature correspondences. If no valid group is found after all combinations of control points have been exhausted, then DARCES returns no solution.

3) Pose Refinement: In the pose refinement stage, outlier feature correspondences are first rejected with RANSAC (*Ra*ndom *Sa*mple *C*onsensus) [19]. The remaining, inlier feature correspondences can then combined with odometry and orientation measurements into a Simultaneous Localization And Mapping (SLAM) problem to refine pose estimates. The Multi-frame Odometry-compensated Global Alignment (MOGA) algorithm, solves this problem by minimizing the errors between all available measurements and desired estimates. This is essentially a batch SLAM algorithm that fuses relative and absolute pose measurements over an entire rover traverse. In the single-scan version of the algorithm used in this study, MOGA estimates the pose of the sensor and the position of the features within the global map using all inlier feature correspondences. Further details are available in [13].

B. Visual Position Estimation for Rovers

The VIPER algorithm 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 DEM and the closest match is considered the most likely position. Various versions of the algorithm have been described [20], [12]. The latter of these is investigated in this paper to compare against MOGA.

The algorithm assumes that the heading of the platform is well known (from celestial and gravitational observations as in [18]) 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.

1) Skyline Segmentation: The measurement used in VIPER to estimate the rover's position is a set of elevation samples derived from the skyline profile in a panoramic image. Each time the lidar took a long-range scan (to be used in MOGA), it captured a series of 10 overlapping pictures at fixed azimuth angles, using an onboard digital camera. The commercial photo stitching software *AutoPano Pro*[2] was used to generate a panoramic image from these pictures. The resulting panorama is built using a spherical camera model; 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[3]. 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

[2]Available at `http://www.autopano.net`

[3]Skyline segmentation was automated in the original algorithm [12], but the automation was not implemented for this study.

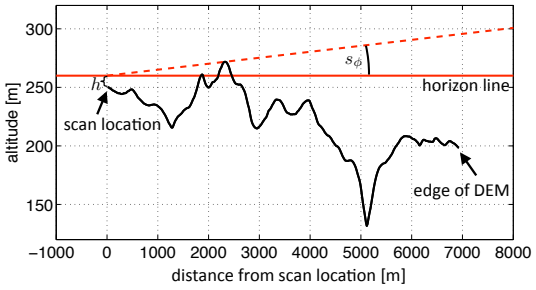

Fig. 4. The geometry of a single elevation sample in VIPER's skyline rendering algorithm. The altitude of points along a line are looked up in the DEM and corrected for dip due to curvature of the planet. For each azimuth angle, ϕ, the maximum elevation angle between the horizon line an altitude sample is returned as s_ϕ.

is sampled at $\phi = 0^\circ \ldots 359^\circ$ and these samples are stacked into a column, $\mathbf{m}$:

$$\mathbf{m} := \begin{bmatrix} m_0 \\ \vdots \\ m_{359} \end{bmatrix} \tag{2}$$

2) Skyline Rendering: A predicted skyline is rendered at each position, $\mathbf{p}$,

$$\mathbf{p} := \begin{bmatrix} x \\ y \end{bmatrix} , \tag{3}$$

in the DEM, 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 degree 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,

$$S(\phi, \mathbf{p}, \delta p, N) = \left\{ a_j \middle| \, j = 0 \ldots N \right\} , \tag{4}$$

where

$$a_j = z\left(\mathbf{p} + j \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix} \delta p \right) - d(j\,\delta p) . \tag{5}$$

The step size is set based on the resolution of the DEM, the number of steps is set to ensure the DEM is sampled to its edge, and the function $d(\cdot)$ is a dip correction for the curvature of the planet (further details are available in [13]). Finally, using the height of the sensor above the ground, h, each altitude sample, a_j, is converted to an elevation angle, e_j:

$$e_j = \operatorname{atan2}\left(a_j - (a_0 + h), j\,\delta p\right) \tag{6}$$

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

$$s_\phi = \max_j (e_j) \tag{7}$$

Figure 4 depicts the geometry of the elevation calculation. A

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

$$\mathbf{s}(\mathbf{p}) = \begin{bmatrix} s_0 \\ \vdots \\ s_{359} \end{bmatrix} \tag{8}$$

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 only has to be performed once, and it speeds up the position estimation algorithm by several orders of magnitude.

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:

$$p(\mathbf{p}|\mathbf{m}) = \frac{p(\mathbf{m}|\mathbf{p})p(\mathbf{p})}{p(\mathbf{m})} \tag{9}$$

The measurement model is

$$\mathbf{m} = \mathbf{s}(\overline{\mathbf{p}}) + \delta\mathbf{m}\,, \tag{10}$$

where $\overline{\mathbf{p}}$ 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:

$$\delta\mathbf{m} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{11}$$

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

$$\mathbf{R} = \sigma^2\mathbf{1}\,, \tag{12}$$

where σ is the measurement's standard deviation.

Given a particular measurement, $\mathbf{m}$, the error, $\mathbf{e}$, at a particular place in the map, $\mathbf{p}$, is

$$\mathbf{e}(\mathbf{m}, \mathbf{p}) := \mathbf{m} - \mathbf{s}(\mathbf{p})\ . \tag{13}$$

Hence, the likelihood of $\mathbf{m}$ given $\mathbf{p}$ is[4]

$$p(\mathbf{m}|\mathbf{p}) = \frac{1}{\sqrt{\det(2\pi\mathbf{R})}} \exp\left(-\frac{1}{2}\mathbf{e}(\mathbf{m}, \mathbf{p})^T \mathbf{R}^{-1}\mathbf{e}(\mathbf{m}, \mathbf{p})\right). \tag{14}$$

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 ([12] 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

$$p(\mathbf{m}) = \sum_{\rho \in K} p(\mathbf{m}|\rho)p(\rho)\,, \tag{15}$$

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

[4]This observation likelihood equation is different than the one published in [12], 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 (14) has been used in this implementation as it is true to the noise model.

4) The VIPER Algorithm: 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 one degree increments to produce a vector of elevation measurements, $\mathbf{m}$.
3) Find $\mathbf{p}^\star = \max_{\mathbf{p}}(p(\mathbf{p}|\mathbf{m}))$.

III. FIELD TESTING

To compare the two algorithms, a realistic dataset was collected from Devon Island, Nunavut just North of the Haughton Crater at 75°22'N and 89°41'W. The area's geological features and lack of vegetation make it a unique Mars/Moon analogue site [21]. In these comparison tests, MOGA and VIPER were provided the same orbital maps and heading measurements. The similarity of the input data ensures a fair comparison of the two approaches.

In total, 40 separate lidar scans/image panoramas were collected over two weeks using the Optech ILRIS3D-ER lidar mounted on a pan-tilt unit as seen in Figure 2. Designed as a accurate, long-range mapping sensor, this lidar has a maximum range of about 1.5km in Extended Range mode, a beam divergence of 0.00974° and a range accuracy of 7mm at a distance of 50m. The vertical and horizontal scan resolutions were respectively set to 0.03° and 0.06°. The vertical resolution was smaller to compensate for the oblique scanning angle. With these settings, a scan with a field of view of 360° in the horizontal and 20° in the vertical finished in about 30 minutes. The lidar also captured images at 10 equally-spaced azimuth angles (used to build the image panoramas) from an on-board, 6-megapixel digital camera.

A. Maps

The DEM 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 map's x and y resolutions were respectively 13m and 24m. In order to be more compatible with these algorithms, the global map was interpolated at the smaller resolution to form a uniform grid.

B. Orientation Measurements

Orientation measurements were a required input for the VIPER algorithm and they improve the efficiency of MOGA. Heading measurements for A01-A25 were obtained from a sun sensor/inclinometer pair [18], while those from A26-A37 were computed knowing the GPS and local positions of a distant target [13]. Roll and pitch were effectively measured by leveling the lidar with its built-in, two-axis bubble level.

C. Ground-Truth

Ground-truth xy-position measurements were obtained from a Garmin GPSMAP 76CSx by averaging measurements for several minutes.

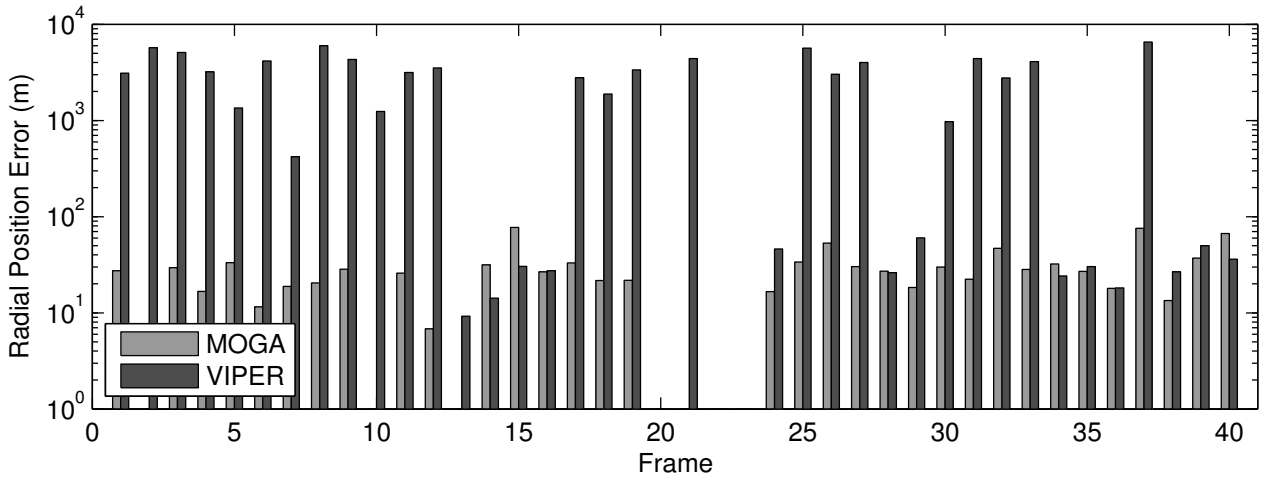

Fig. 5. VIPER and MOGA position errors. The vertical axis is on a log scale.

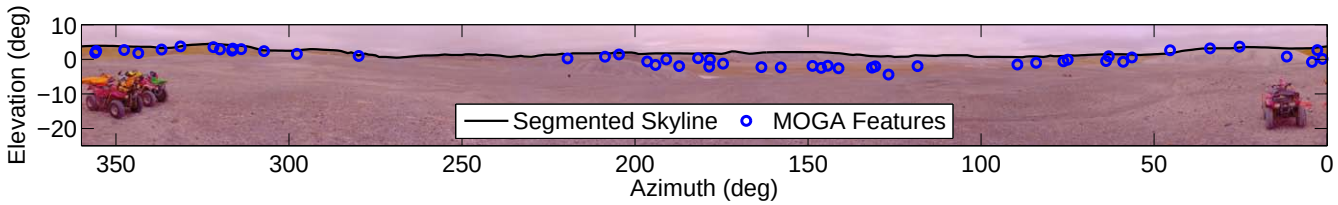

Fig. 6. VIPER's segmented skyline and MOGA's features plotted on image panorama. MOGA is able to use topographic information below the skyline whereas VIPER is restricted to using information on the skyline.

IV. RESULTS

Figure 5 shows the main quantitative output of this study: a comparison of the radial position error of the estimates of MOGA and VIPER on the common dataset. The error in the MOGA estimate was less than 100 meters for all 33 scans where it was able to produce a solution. On the other seven scans (no error reported for MOGA on Figure 5), the DARCES component of the algorithm failed and no answer was returned. Viper produced an estimate with an error of less than 100 meters on 13 out of 40 scans. The results show that, while VIPER was computationally less demanding, MOGA's localization performance was generally superior. The difference can primarily be attributed to the use of 3D information in MOGA, since VIPER is unable to distinguish between areas that produce similar 2D skylines. Figure 6 plots the local features available to MOGA and the segmented skyline used by VIPER on the same panorama. In this scan, MOGA is able to use the topographic relief below the skyline whereas VIPER is limited to the horizonal boundary.

VIPER outperformed MOGA in a small number of cases, notably for scan A13. MOGA found no solution for this scan since the lidar was located in a relatively flat region where good features were out of lidar range. Meanwhile, the lack of occlusions nearby allowed the built-in camera to observe the horizon many kilometers in the distance. This result suggests that there are situations where VIPER should be used instead of MOGA, particularly where the nearby terrain is flat but distant features are visible in the panoramic image. Figure 7 compares the horizon of data used by VIPER and MOGA on scan A13. The rendered skyline used by VIPER includes data up to 7 kilometers away whereas MOGA is limited to the 1.5 kilometer range of the lidar.

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 DEM, and therefore not accurately represented in the global map.

In the end, there are a number of common factors limiting 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 [12]; the panoramas gathered in the Atacama desert as part of the that work[5] show large-scale relief along the horizon—much greater relief than exhibited on Devon Island.

[5] http://www.cs.cmu.edu/~VIPER/AtacamaMission/

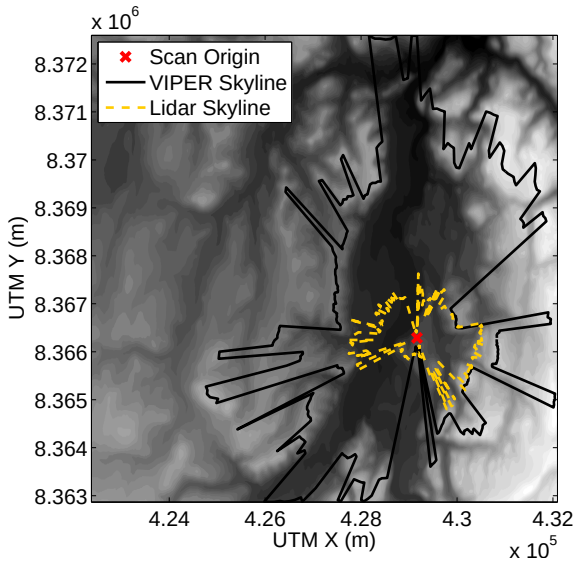

Fig. 7. Horizon of the lidar scan compared to the data used to render the horizon in VIPER with the DEM plotted in the background. VIPER can use topographic information that is much further away from the sensor. This allows VIPER to succeed in situations where the major topographic relief in the area is far away from the sensor.

- **The amount of topographic relief within the range of the sensor**—VIPER is limited to using the topographic relief visible on the horizonal 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. While this may be mitigated through careful attention during data collection, future work in this area should address the problem of some uncertainty in orientation.

V. CONCLUSION

This paper has presented an experimental comparison of two algorithms for performing surface-based global localization in the absence of GPS. The comparison was performed on a common dataset collected at a planetary analog site in the Canadian High Arctic. The work shows that the algorithms share a common set of factors limiting performance. Both algorithms are limited by the quality of the DEM and require good topographic relief within the range of the sensor. On this dataset, MOGA performed much better than VIPER, producing estimates with radial position error under 100 meters on 33 out of 40 scans. VIPER was not as successful (13/40 scans) but was able to obtain a solution is some situations where MOGA failed.

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the Mars Institute, the Canadian Space Agency, the Natural Sciences and Engineering Research Council of Canada, and the Canada Foundation for Innovation for funding this work and making our field trials on Devon Island possible.

REFERENCES

[1] A. Behar, J. Matthews, F. Carsey, and J. Jones, "NASA/JPL Tumbleweed polar rover," in *Proceedings of IEEE Aerospace Conference*, vol. 1, 2004.

[2] J. Guinn, "Mars surface asset positioning using in-situ radio tracking," in *Proceedings of the 11 th Annual AAS/AIAA Space Flight Mechanics Meeting, Santa Barbara, CA*, 2001, pp. 45–53.

[3] R. Li, R. E. Arvidson, K. Di *et al.*, "Opportunity rover localization and topographic mapping at the landing site of Meridiani Planum, Mars," *Journal of Geophysical Research*, vol. 112, no. E2, February 2007.

[4] R. Li, S. W. Squyres, and R. E. Arvidson, "Initial results of rover localization and topographic mapping for the 2003 Mars Exploration Rover mission," *Photogrammetric Engineering & Remote Sensing*, vol. 71, no. 10, pp. 1129–1142, October 2005.

[5] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Rover navigation using stereo ego-motion," *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 215–229, 2003.

[6] T. Parker, M. Malin, M. Golombek, T. Duxbury, A. Johnson, J. Guinn, T. McElrath, R. Kirk, B. Archinal, L. Soderblom, R. Li, the MER Navigation Team, and A. S. Team, "Localization, localization, localization," in *Lunar and Planetary Science XXXV*, 2004.

[7] M. W. Maimone, P. C. Leger, and J. J. Biesiadecki, "Overview of the mars exploration rovers' autonomous mobility and vision capabilities," in *IEEE International Conference on Robotics and Automation (ICRA) Space Robotics Workshop*, April 2007.

[8] J. Enright, P. Furgale, and T. Barfoot, "Sun sensing for planetary rover navigation," in *Proceedings of the IEEE Aerospace Conference. Big Sky, MT*, March 2009.

[9] Y. Kuroda, T. Kurosawa, A. Tsuchiya, and T. Kubota, "Accurate localization in combination with planet observation and dead reckoning for lunar rover," *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 2, pp. 2092–2097 Vol.2, 26-May 1, 2004.

[10] F. Cozman and E. Krotkov, "Robot localization using a computer vision sextant," in *IEEE International Conference on Robotics and Automation*, vol. 1, May 1995, pp. 106–111.

[11] R. Li, K. Di, and A. B. Howard, "Rock modeling and matching for autonomous long-range Mars rover localization," *Journal of Field Robotics*, vol. 24, no. 3, pp. 187–203, March 2007.

[12] F. Cozman, E. Krotkov, and C. Guestrin, "Outdoor visual position estimation for planetary rovers," *Auton. Robots*, vol. 9, no. 2, pp. 135–150, 2000.

[13] P. Carle, P. T. Furgale, and T. D. Barfoot, "Long-range rover localization by matching lidar scans to orbital elevation maps," *Journal of Field Robotics*, vol. 27, no. 3, pp. 344–370, 2010.

[14] van den Boomgaard, R. and van Balen, R., "Methods for fast morphological image transforms using bitmapped binary images," *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 3, pp. 252–258, 1992.

[15] R. Haralick and L. Shapiro, *Computer and Robot Vision. Vol. 1*. Addison-Wesley, 1992.

[16] C.-S. Chen, Y.-P. Hung, and J.-B. Cheng, "RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1229–1234, 1999.

[17] K. Arun, T. Huang, and S. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 698–700, 1987.

[18] P. T. Furgale, J. Enright, and T. D. Barfoot, "Sun sensor navigation for planetary rovers: Theory and field testing," *IEEE Transactions on Aerospace and Electronic Systems*, 2010, manuscript # TAES-2009-00219, accepted on April 13, 2010.

[19] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[20] F. Cozman and E. Krotkov, "Automatic mountain detection and pose estimation for teleoperation of lunar rovers," in *International Conference on Robotics and Automation*, vol. 3, April 1997, pp. 2452–2457.

[21] P. Lee, "Mars on Earth: The NASA Haughton-Mars Project," *Ad Astra: The Magazine of the National Space Society*, vol. 14, no. 3, 2002.