Visual Teach and Repeat for Long-Range Rover Autonomy

Paul Furgale and Timothy D. Barfoot
Institute for Aerospace Studies, University of Toronto, Toronto, ON M3H 5T6, Canada
e-mail: paul.furgale@utoronto.ca, tim.barfoot@utoronto.ca

Received 30 September 2009; accepted 12 April 2010

This paper describes a system built to enable long-range rover autonomy using a stereo camera as the only sensor. During a learning phase, the system builds a manifold map of overlapping submaps as it is piloted along a route. The map is then used for localization as the rover repeats the route autonomously. The use of local submaps allows the rover to faithfully repeat long routes without the need for an accurate global reconstruction. Path following over nonplanar terrain is handled by performing localization in three dimensions and then projecting this down to a local ground plane associated with the current submap to perform path tracking. We have tested this system in an urban area and in a planetary analog setting in the Canadian High Arctic. More than 32 km was covered—99.6% autonomously—with autonomous runs ranging from 45 m to 3.2 km, all without the use of the global positioning system (GPS). Because it enables long-range autonomous behavior in a single command cycle, visual teach and repeat is well suited to planetary applications, such as Mars sample return, in which no GPS is available. © 2010 Wiley Periodicals, Inc.

1. INTRODUCTION

In environments lacking a global positioning system (GPS) or equivalent, long-range autonomous navigation for rovers becomes a very difficult problem. Relative motion estimation systems based on some combination of visual, inertial, and odometric sensing have become increasingly accurate. However, regardless of the level of accuracy, the error in the position estimate for any of these methods will grow without bound as the rover travels, unless periodic global corrections are made.

This paper describes a complete system for long-range, autonomous operation of a mobile robot in outdoor, unstructured environments. This is achieved using *only* a stereo camera for sensing and a teach-and-repeat operational strategy. During a learning phase—the *teach pass*—the rover is piloted over the desired route (either manually or using some external autonomous system) while the mapping system builds a manifold map composed of overlapping submaps. The topologically connected submaps are then used for localization during the autonomous traversal phase—the *repeat pass*. The hybrid topological/metric formulation alleviates the requirement for an accurate global reconstruction while avoiding the unbounded error growth of purely relative motion estimation. Furthermore, using a series of small submaps decouples the computational complexity of retracing a route from the path length. Similar systems have been proposed for rovers navigating indoors (Goedeme, Tuytelaars, Van Gool, Vanacker, & Nuttin, 2005), in mines (Marshall, Barfoot, & Larsson, 2008), or outdoors in planar environments (Royer, Lhuillier, Dhome, & Lavest, 2007; Šegvić, Remazeilles, Diosi, & Chaumette, 2009), but this is the first system shown to work over multikilometer autonomous traverses in highly three-dimensional (3D), outdoor, unstructured environments, without the use of GPS.

Multimedia files may be found in the online version of this article.

Our visual teach-and-repeat system is appropriate to many scenarios requiring rover autonomy in which GPS is not available. We envision such a system being used to support Mars sample return (Figure 1) or automated convoying of equipment between lunar landing sites and habitats. Hence we have tested our system at a Mars/Moon analog site on Devon Island in the Canadian High Arctic. For completeness, we have also performed tests in an urban setting, over grass, over challenging 3D terrain, and through extreme lighting changes (indoor to outdoor). This paper reports results for more than 32 km of evaluation of our algorithm with 99.6% of the distance traveled autonomously, all without the use of GPS.

2. RELATED WORKS

In an early paper on vision-based map building, Brooks (1985) outlined some basic principles for robotic mapping:

- The world is inherently 3D. Localization and mapping should reflect this.
- Uncertainty in sensing will lead to maps that are globally inconsistent. However, to enable robot autonomy, maps only need to be locally consistent.

To deal with this, Brooks proposed a map composed of an abstract graph of free-space primitives. Similar in concept, Howard, Sukhatme, and Mataric (2006) designed and implemented a multiagent system that represented the robot's

View this article online at wileyonlinelibrary.com • DOI: 10.1002/rob.20342

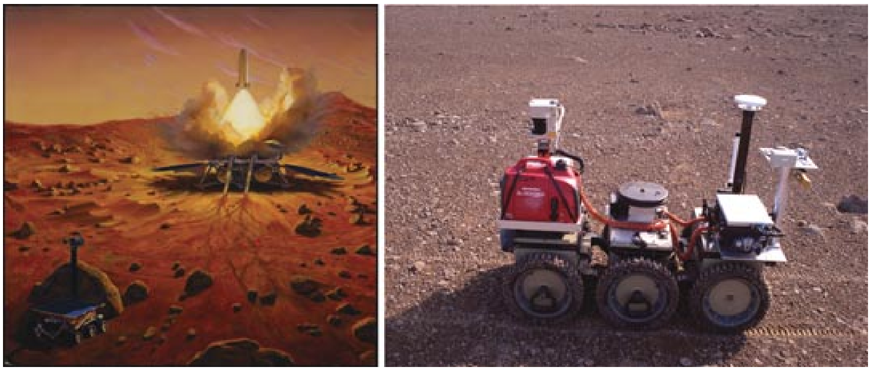


Figure 1. We envision our teach-and-repeat navigation framework being used to support Mars-sample-return mission operations (image credit: NASA/JPL). After sample acquisition, the rover would retrace its route, returning to the lander in a single command cycle. The image on the right shows our rover driving back along its outbound track in a planetary analog setting on Devon Island, Canada.

map as a manifold embedded in a higher dimensional space. Manifold mapping changes the way a map represents the world. A map becomes topological in the sense that it defines a sequence of connected spaces, but the spaces in the map may have a many-to-one correspondence with the world. This topology is represented by dividing the map into a graph of submaps (Bosse, Newman, Leonard, & Teller, 2004; Eade & Drummond, 2008; Howard et al., 2006; Marshall et al., 2008) or using a continuous relative representation (Mei, Sibley, Cummins, Newman, & Reid, 2009; Sibley, Mei, Reid, & Newman, 2009). Incremental errors that would cause inconsistencies in a purely metric map disappear within the manifold representation. As a result, loop-closing decisions may be delayed (Howard et al., 2006) and loops may be closed in constant time, regardless of the size of the map (Sibley et al., 2009). Manifold mapping removes the constraint that maps be globally consistent, but in order to be useful for localization, the neighborhood around the robot must still appear to be locally Euclidean.

To see where this local-Euclidean constraint expresses itself in the simultaneous localization and mapping (SLAM) problem, we examine the structure of the basic SLAM equations. The SLAM problem is formulated probabilistically as the task of estimating the joint posterior density of the map, $\mathbf{m}$, and vehicle state at time k, $\mathbf{v}_k$, given all previous measurements, $\mathbf{z}_{0:k}$, control inputs, $\mathbf{u}_{0:k}$, and prior knowledge, $\mathbf{x}_0$ (Durrant-Whyte & Bailey, 2006):

$$p(\mathbf{x}_k, \mathbf{m}|\mathbf{z}_{0:k}, \mathbf{u}_{0:k}, \mathbf{x}_0). \tag{1}$$

Most solutions to this problem involve computing $p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m})$, the likelihood of the measurement vector, $\mathbf{z}_k$, given the current state and map estimates. The likelihood is then expressed using an observation model, $\mathbf{h}(\cdot)$, such that

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{m}) + \mathbf{v}_k, \tag{2}$$

where $\mathbf{v}_k$ is observation noise. The properties of Eq. (2) determine the form of the constraint. Most navigation sensors discern something about the geometry in the robot's local neighborhood and, for a map to be useful, the neighborhood must appear to be Euclidean to the sensor suite; any deviation must be small enough to hide in $\mathbf{v}_k$. This is the motivation behind the adaptive optimization scheme in Sibley et al. (2009) and the choice of submap size in Marshall et al. (2008). If this constraint is satisfied, the map is still useful for localization, even if the global reconstruction is inaccurate.

Visual teach-and-repeat navigation systems have been built on this very concept, combining topologically connected key frames with a controller that attempts to drive the robot to the same viewpoints along the path. Our review of teach-and-repeat systems will focus on camera-based systems. Marshall et al. (2008) and Newman, Leonard, Tardos, and Neira (2002) both used planar laser-ranging devices to build teach-and-repeat systems in indoor corridors. The systems are well suited to these environments (an underground mine in the former, an office building in the latter), but in outdoor, unstructured environments, there is no guarantee that any walls will be within the range of a laser sensor. Cameras, on the other hand, are not dependent on scene topography. They require only ambient light and scene texture to return useful images. Wide field-of-view and omnidirectional cameras capture the large-scale geometry and appearance of a scene, which are generally unique to a particular viewpoint and somewhat robust to small-scale changes in the scene. In this way cameras are well suited to recognize places previously visited.

Early work in visual teach-and-repeat navigation recognized the key benefit of such a system: an accurate global reconstruction is not needed for a robot to retrace its path (Baumgartner & Skaar, 1994; Brooks, 1985).

Systems implementing teach and repeat span the continuum of map-based approaches, as categorized by Bonin-Font, Ortiz, and Oliver (2008). Different systems frame the problem as purely metric (Baumgartner & Skaar, 1994; Royer et al., 2007), combined topological/metric (Goedemé, Nuttin, Tuytelaars, & Van Gool, 2007; Šegvić et al. 2009), or purely topological (i.e., only track the position along the path) (Matsumoto, Inaba, & Inoue, 1996; Zhang & Kleeman, 2009). Our system can be described as topological/metric. Localization is performed in 3D space, path tracking is performed in a local planar projected space, and route management is topological.

Appearance-based systems compare large portions of the input image with prototype images captured during the teach pass. These algorithms are derived from the work of Matsumoto et al. (1996). They developed a system for autonomous navigation within corridors. During the route-learning phase, an ordered sequence of images is captured using a monocular camera. During route repeating, progress along the sequence is tracked. A template from the center of each new image is correlated with the two nearest images along the route. The steering angle is determined from the correlation peak offset, and the current image index is incremented when the next image returns a higher correlation score than the previous. Jones, Andresen, and Crowley (1997) extend this basic framework and introduce a second camera to more accurately track the position along the route, and Payá, Reinoso, Gil, Pedrero, and Ballesta (2007) try to make the process more efficient using principal component analysis. The most impressive demonstration of appearance-based path following was developed by Zhang and Kleeman (2009). They report more than 18 km of tests using an omnidirectional imaging system. Position along the route and steering angle determination are similar to those of Matsumoto et al. (1996), but significant image preprocessing is performed to make the system robust to changing lighting conditions. All of the appearance-based techniques rely on the assumption of planar camera motion. They are therefore not suitable for outdoor unstructured environments.

Another group of algorithms uses image features for mapping and navigation but relies on the planarity of the camera's motion to reduce the complexity of the problem. Ohno, Ohya, and Yuta (1996) use a monocular camera and a bearing-only, two-dimensional localization when navigating between prototype images. Tang and Yuta (2001) develop a similar system for a robot with an omnidirectional camera. They use color information to describe line features and planar localization based on the bearing of the line correspondences. The algorithm described by Bekris, Argyros, and Kavraki (2006) and Argyros, Bekris, Orphanoudakis, and Kavraki (2005) tracks point features between omnidirectional images. Instead of triangulating the features, they use only the bearing of the measurements and develop a control law to drive the robot between viewpoints. Jun, Miura, and Shirai (2000) describe an algorithm that uses range measurements from a stereo camera to detect obstacles, which are projected down to a plane and used for localization while repeating the route. Blanc, Mezouar, and Martinet (2005) developed a system that followed indoor visual routes. As the camera was facing the ceiling, the system could solve three-degree-of-freedom homographies using features tracked between exemplar images and images taken from the robot's current position. Courbon, Mezouar, and Martinet (2008) extended this work to use an omnidirectional camera. Chen and Birchfield (2006) developed a homing system that uses a KLT point tracker on images captured from a forward-facing monocular camera system. Stored points from the training run are matched with points on the repeat run, and all matched points contribute to a simple visual servoing scheme. Goedemé, Tuytelaars, and Van Gool (2005) improve the process of extracting line features by making every part of the algorithm invariant to changes in illumination and viewpoint (assuming that the camera is restricted to moving in the plane). They also move to use point features detected using the scale invariant feature transform (SIFT) (Lowe, 2004). During the map-building phase, features are triangulated between views and their 3D coordinates are stored in the map. Three-dimensional localization against the map is performed by observing features and estimating the essential matrix of the camera. Owing to the introduction of local metric 3D information derived from point features, this algorithm, and the similar one described by Booij, Terwijn, Zivkovic, and Krose (2007), could be adapted to work with nonplanar camera motion.

Developing a teach-and-repeat system for outdoor, unstructured environments requires the handling of nonplanar camera motions. Using point image features for localization removes the planarity constraint and enables localization in three dimensions as required by our system. There has been some work in this area using forward-facing monocular cameras. The work of Royer et al. (2007) represents one approach to this task. During the mapping phase, point features detected in a monocular image sequence are tracked between images, and data from the entire route are subject to a large, multilevel estimation routine to find the feature positions and the robot poses. The poses become a reference path, and the features are used as a map. To repeat the route, features in the current image are associated with features in the map and used to estimate the rover's position. In contrast to this global reconstruction approach, Šegvić et al. (2009) develop a system that performs many local reconstructions during the mapping phase, using two-view geometry to triangulate feature points seen in a monocular image sequence. During the repeat traverse, the rover's 3D pose is estimated using the triangulated features. Interestingly, the 3D pose is used only to localize the robot topologically; the steering angle is derived from a simple visual servoing rule similar to that used by Chen

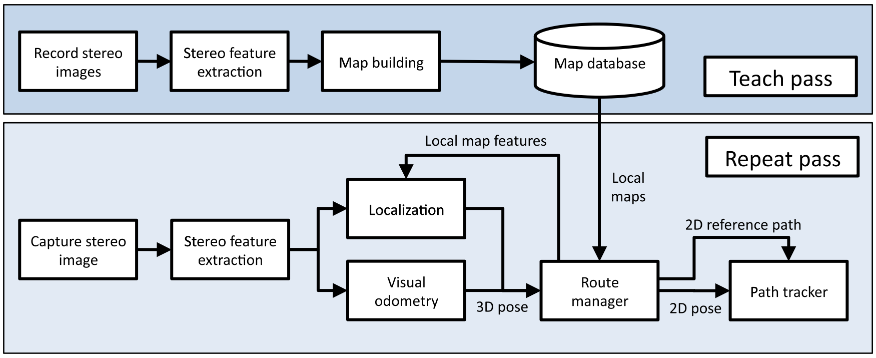


Figure 2. An overview of the major processing blocks in our system.

and Birchfield (2006). These algorithms are most similar to the one we propose. However, we use a stereo camera as the main sensor.

A stereo camera provides metric structure within each stereo pair of images, simplifying the reconstruction considerably. A very simple teach-and-repeat mode was built into the stereo navigation system described by Konolige, Agrawal, Blas, Bolles, Gerkey, et al. (2009). During a mapping phase, the rover's path is estimated using visual odometry (VO) (Konolige, Agrawal, & Solà, 2007). To repeat the route, they estimate the rover's position at the start of the route by matching the current view with the first image in the learning sequence. Then the route is repeated without relocalizing against the map. Although this method worked for the short paths in question (generally less than 200 m), longer routes would require localization corrections to maintain global consistency.

We show that it is possible to use stereo vision alone to retrace a long route with nonplanar camera motion in an outdoor, unstructured environment. Our work is based on VO (Konolige et al., 2007; Maimone, Johnson, Cheng, Willson, & Matthies, 2006; Moravec, 1980; Nistér, Naroditsky, & Bergen, 2004). Specifically, we start with the vision pipeline common to all of these papers. This involves tracking stereo features, using the random sample consensus (RANSAC) algorithm to reject outlier feature tracks, and using an iterative scheme to solve for the rover's pose. We transform the basic pipeline into a mapping and localization system and demonstrate that our algorithm can be used to drive multikilometer autonomous routes in a single command cycle.

3. SYSTEM OVERVIEW

This section will present a detailed description of our visual teach-and-repeat system. The major processing blocks of our system are depicted in Figure 2. Both the teaching and following systems are based on calibrated, parallel stereo vision, so fundamentals and notation will be presented first. Next, the route-learning system will be described. Finally, we outline the route-following algorithm and its handling of failures.

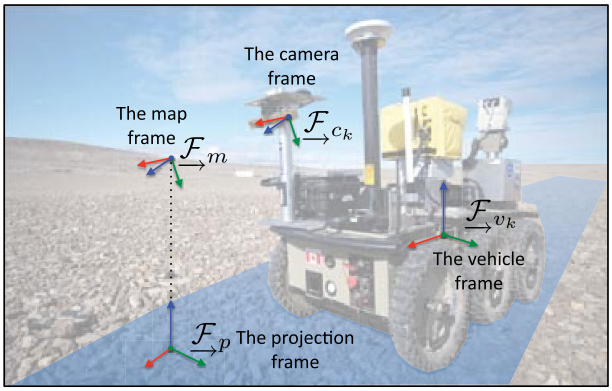

The camera frame
$\underrightarrow{\mathcal{F}}_{c_k}$
The map frame
$\underrightarrow{\mathcal{F}}_{m}$
$\underrightarrow{\mathcal{F}}_{v_k}$
The vehicle frame
$\underrightarrow{\mathcal{F}}_{p}$ The projection frame

Figure 3. Coordinate frames under consideration.

The coordinate frames used in our system are depicted in Figure 3. The map frame $\underrightarrow{\mathcal{F}}_{m}$ is the frame in which all 3D estimation occurs. We define $\underrightarrow{\mathcal{F}}_{c_k}$ to be a coordinate frame attached to the left camera of a stereo pair at time k. The attitude of the camera at this time may be described by $\mathbf{C}_{m,c_k}$, the rotation matrix that transforms vectors from $\underrightarrow{\mathcal{F}}_{c_k}$ to $\underrightarrow{\mathcal{F}}_{m}$. Similarly, we define the camera's position as $\boldsymbol{\rho}_m^{c_k,m}$, a vector from the origin of $\underrightarrow{\mathcal{F}}_{m}$ to the origin of $\underrightarrow{\mathcal{F}}_{c_k}$ (denoted by the superscript) and expressed in $\underrightarrow{\mathcal{F}}_{m}$ (denoted by the subscript). Using similar notation, we define a rotation, $\mathbf{C}_{c_k,v_k}$, and translation, $\boldsymbol{\rho}_{v_k}^{c_k,v_k}$, between the camera frame, $\underrightarrow{\mathcal{F}}_{c_k}$, and vehicle frame, $\underrightarrow{\mathcal{F}}_{v_k}$. This transformation is assumed to be static, but it could be time varying. Finally, the projection frame, $\underrightarrow{\mathcal{F}}_{p}$, defines the projection from three dimensions to two, as required by our path-tracking controller.

3.1. Stereo Pipeline

To enable this project and others, we have developed a sparse-stereo pipeline implemented entirely on the

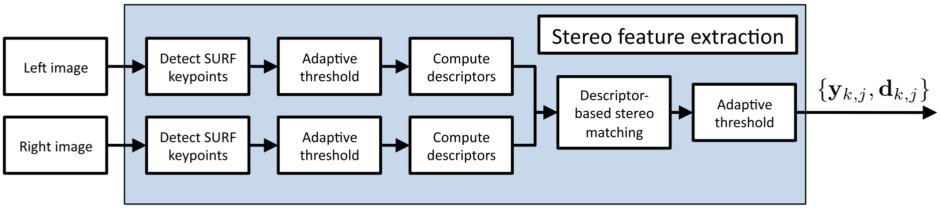


Figure 4. An overview of our GPU-accelerated stereo feature pipeline.

graphics processing unit (GPU) using NVIDIA's Compute Unified Device Architecture (CUDA) toolkit. As shown in Figure 4, the pipeline is based on the speeded-up robust features (SURF) algorithm (Bay, Ess, Tuytelaars, & Van Gool, 2008). As the camera is mounted on a rover that cannot roll arbitrarily, we use the upright descriptor that is not invariant to rotation.[1]

Adaptive thresholds are used to ensure constant computational complexity and robust performance across different scenes and lighting conditions. Each keypoint j coming out of the stereo pipeline at time k has image coordinates, $\mathbf{y}_{k,j}$, and a 64-dimensional description vector, $\mathbf{d}_{k,j}$.

When dealing with keypoints found in stereo image pairs, we use disparity coordinates (Demirdjian & Darrell, 2002). Keypoint $\mathbf{y}_{k,j}$ has the components

$$\mathbf{y}_{k,j} := \begin{bmatrix} u \\ v \\ d \end{bmatrix}.$$

The left camera is dominant in our formulation, so u and v are, respectively, the horizontal and vertical pixel coordinates in the left image, and d is the *disparity*—the difference between the left and right horizontal pixel locations.

The calibrated, parallel stereo camera model has the following parameters: c_u, c_v, the horizontal and vertical optical center in pixels (from the top left of the image); f_u, f_v, the horizontal and vertical focal length in pixels; and b, the camera baseline (i.e., distance between the two centers of projection) in meters.

The observation model, $\mathbf{h}(\cdot)$, is a nonlinear function that projects points expressed in the left camera frame into the disparity coordinates. Given a 3D feature location, $\mathbf{p}_{c_k}^{j,c_k}$, with its coordinates expressed in the left camera frame,

$$\mathbf{p}_{c_k}^{j,c_k} = \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

the image of that point, $\mathbf{y}_{k,j}$, is

$$\mathbf{y}_{k,j} = \mathbf{h}(\mathbf{p}_{c_k}^{j,c_k}) = \frac{1}{z}\begin{bmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ 0 & 0 & 0 & bf_u \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{3}$$

Because we are using a calibrated stereo camera, Eq. (3) is invertible. The inverse observation model, $\mathbf{g}(\cdot)$, triangulates points seen in a stereo pair:

$$\mathbf{p}_{c_k}^{j,c_k} = \mathbf{g}\left(\mathbf{y}_{k,j}\right) = \frac{b}{d}\begin{bmatrix} u - c_u \\ \dfrac{f_u}{f_v}(v - c_v) \\ f_u \end{bmatrix} \tag{4}$$

3.2. A Generic Localization Module

Throughout this work, we use a generic localization module based on the stereo VO algorithm pioneered by Moravec (1980) and refined by Matthies (1989) and others. The outline is shown in Figure 5. Stereo keypoints are tracked against a feature database, the tracks are subject to outlier detection, and the inlying tracks are used to solve for the current pose of the camera. By substituting different blocks for the feature database and numerical solution, we are able to build all of the different operating modes used for teach-and-repeat navigation: *map building*, *VO*, *submap selection*, and *localization*. We will refer back to this section as we specify the details used in these operating modes. Here we present the specific requirements of each block.

A feature database represents a map against which the robot can localize. To this end, it supplies information about the set of features available for this task: N, the number of features in the database; $\mathbf{q}_m^{i,m}$, the $[x_i \;\; y_i \;\; z_i]^T$ position of feature i with respect to and expressed in $\underrightarrow{\mathcal{F}}_m$; and $\mathbf{v}_i$, the SURF descriptor associated with feature i.

Data association is performed by looking for nearest neighbors in descriptor space. Using this notation and that of Section 3.1, the output of the first block in Figure 5 is a list of candidate feature tracks, each associating a feature i in the database to a feature j from the most recent stereo pair.

The candidate tracks are passed to the outlier detection block. We have implemented preemptive RANSAC (Nistér, 2005), as it will on average produce the best set of inliers given a fixed computational budget. Treating the feature

[1] Although we have used upright descriptors, the trajectory could still be truly 3D (as it could be for a helicopter or underwater vehicle) if the controller on the vehicle drove the robot back to the same views. Upright descriptors do not limit the trajectory; they only impede localization in the face of camera-frame roll error during route repeating.

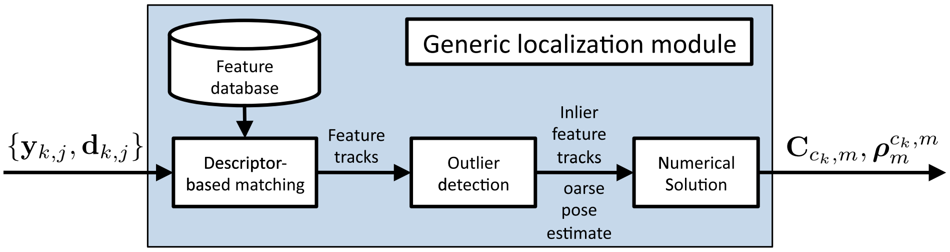

Generic localization module
Feature database
$\{\mathbf{y}_{k,j}, \mathbf{d}_{k,j}\}$
Descriptor-based matching
Feature tracks
Outlier detection
Inlier feature tracks
oarse pose estimate
Numerical Solution
$\mathbf{C}_{c_k,m}, \boldsymbol{\rho}_m^{c_k,m}$

Figure 5. An overview of our generic localization module.

database and the incoming stereo keypoints as 3D point clouds [using Eq. (4) to triangulate each feature], we use the three-point quaternion method of Horn (1987) as our hypothesis generator. Preemptive RANSAC generates a set of inlying feature tracks and a coarse estimate of the camera's pose in $\underrightarrow{\mathcal{F}}_m$.

Finally, the inlying feature tracks are passed to a pose solution method. The pose solution has access to the disparity coordinates of each incoming keypoint, the feature database, the pose estimate supplied by RANSAC, and the camera's pose from the last timestep. Using these data it produces an estimate of the camera's pose in $\underrightarrow{\mathcal{F}}_m$: $\mathbf{C}_{c_k,m}$, the camera's attitude with respect to $\underrightarrow{\mathcal{F}}_m$; and $\boldsymbol{\rho}_m^{c_k,m}$, the camera's position with respect to and expressed in $\underrightarrow{\mathcal{F}}_m$. Each solution method is iterative, based on Gauss–Newton minimization, but each operating mode uses a different mathematical formulation.

3.3. Route Teaching

The basic process for route teaching involves driving the path once while logging stereo images and then postprocessing the image sequence into a series of overlapping submaps. The postprocessing task is shown in Figure 6. At the front, a mapping loop incrementally builds the map and estimates the position of the rover within it. Periodically, the map is split, and the raw data are further processed into the format used in the repeat pass.

3.3.1. Teach Pass Localization and Mapping

The mapping loop seems to be solving the SLAM problem. However, the different requirements of this system dictate different design choices. Each submap must be locally consistent, and transformations between adjacent submaps must be reasonable. Outside of these constraints, the overall global consistency of the map sequence should not impact algorithm performance. Because of this, the system does not work toward global consistency. Figure 7 shows an example of a 5-km map sequence compared to GPS together with the robot's view of the map from either end. Although the reconstruction of the complete path is very inaccurate, locally it is sufficient to enable route following.

Submaps are constructed using a specialization of the generic localization module from Section 3.2. The system is initialized with the first keypoint list, $\{\mathbf{y}_{0,j}, \mathbf{d}_{0,j}\}$. The map frame $\underrightarrow{\mathcal{F}}_m$ is defined to be the same as $\underrightarrow{\mathcal{F}}_{c_0}$. All of the keypoints are triangulated using Eq. (4) and placed in the map. In each subsequent frame, incoming keypoints are tracked against the working database and subjected to outlier detection. Let us use n to index the inlying feature tracks. Each track associates feature i in the map to keypoint j. To estimate $\mathbf{C}_{c_k,m}$, and $\boldsymbol{\rho}_m^{c_k,m}$, we define the error term, $\mathbf{e}_n$:

$$\mathbf{e}_n := \mathbf{y}_{k,j} - \mathbf{h}\left[\mathbf{C}_{c_k,m}\left(\mathbf{q}_m^{i,m} - \boldsymbol{\rho}_m^{c_k,m}\right)\right].$$

Letting M_k be the number of feature tracks at time k, we define our objective function, J_k, to be

$$J_k := \frac{1}{2}\sum_{n=1}^{M_k} \mathbf{e}_n^T \mathbf{W}_n \mathbf{e}_n, \tag{5}$$

where $\mathbf{W}_n$ is a weighting matrix based on the inverse of the estimated measurement covariance of $\mathbf{y}_{k,j}$. We linearize Eq. (5) and minimize J_k using the Gauss–Newton method.

When the percentage of features tracked drops below a threshold, τ_f, the pose $(\mathbf{C}_{c_k,m}, \boldsymbol{\rho}_m^{c_k,m})$ is added to the

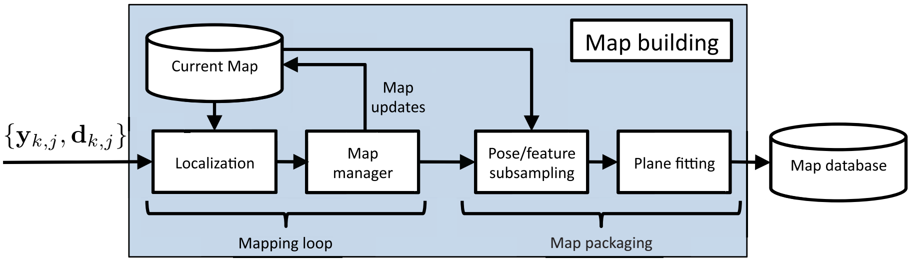


Figure 6. An overview of the mapping process.

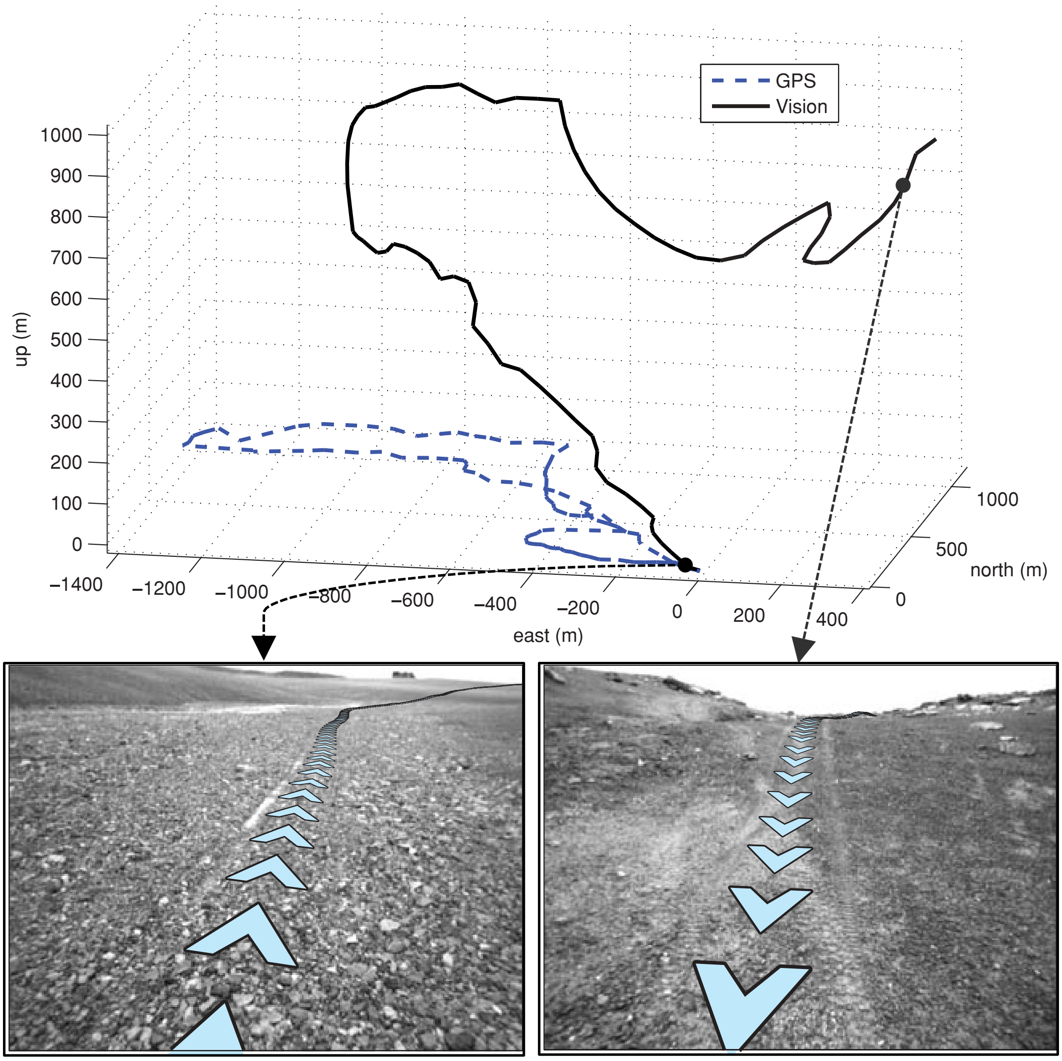


Figure 7. The visual reconstruction of a 5-km rover traverse plotted against GPS (top). Although the reconstruction is wildly inaccurate at this scale, locally it is good enough to enable retracing of the route. The bottom images show views from either end of the path, with the reference path plotted as a series of chevrons. To the rover, the map is locally Euclidean.

reference path, and all of the keypoints are added to the map. Using a threshold avoids generating bloated maps while the robot is sitting still and automatically adjusts the number of features in the map based on the difficulty of the terrain. Using the pose estimated in the preceding step, triangulated keypoints are placed into the map in a common frame, $\underrightarrow{\mathcal{F}}_m$:

$$
\begin{aligned}
\mathbf{q}_m^{i,m} &= \mathbf{C}_{c_k,m}^T \mathbf{g}(\mathbf{y}_{j,k}) + \boldsymbol{\rho}_m^{c_k,m},\\
&= \mathbf{C}_{c_k,m}^T \left(\mathbf{p}_{c_k}^{j,c_k}\right) + \boldsymbol{\rho}_m^{c_k,m}.
\end{aligned}
$$

The prototype feature in our system is based on the triangulated position and SURF descriptor of the first view only. Incoming keypoints that are not successfully tracked are added to the map as seen. Keypoints that are successfully tracked are already present in the map, and so the new observations are discarded. Although there is enough information here to estimate the camera's pose *and* the feature positions—either on the entire map (Royer et al., 2007) or on some sliding window of poses (Konolige et al., 2007; Sibley, Matthies, & Sukhatme, 2008)—our system has no requirement to build a globally consistent map. Furthermore, our results show that this implementation works for the kind of local, metric localization needed in the repeat pass. Although future work may involve some evaluation of the benefits of better reconstruction techniques, local bundle adjustment is not necessary to build a working system.

As poses and features are added to the map, the length of the current reference path is tracked. When the length exceeds a threshold, τ_l, the submap is packaged for the repeat pass and saved to disk. By changing this parameter, our system scales smoothly between a complete global reconstruction (Royer et al., 2007) and view-sequenced route representations that match against single images along the path (Šegvić et al., 2009). We experimented with different values for τ_l early in the development of the algorithm and found that higher values (larger submaps) increased the algorithm's robustness to localization dropouts. However, this robustness came at the cost of increasing computational complexity as larger submaps contained more features. Eventually, we settled on the value $\tau_l = 5$ m; this was as high as we could set this value and still operate at the frame rate necessary to drive the robot at a reasonable speed to conduct long-range experiments.

The map-building step may fail if it is unable to succeed in tracking a minimum number of sparse feature points from the map to the latest image. We have seen this happen if the rover has moved too much between images, with image motion blur experienced in low-light conditions, and in areas with highly repetitive texture. The algorithm tries to deal with single-frame failures by storing the failed image and attempting to track features from the next image. If this fails, the algorithm has no way to associate the current images (and all future images) with the images that have come before. To deal with this, it sets a flag that the end of the current submap is broken, purges the existing map, and starts a new map as if it were processing the first image of a sequence. The *broken map* flag is used as a signal to the repeat pass that the algorithm should stop and look for the next section of the map. This will be described further below.

3.3.2. Teach Pass Map Packaging

When a split in the map is triggered, the current set of reference poses and features are packaged for use in the repeat pass. First, the poses are subsampled to satisfy a minimum-spacing constraint, τ_s. This smoothes the path and puts it in a format suitable for our path tracker. All experiments in this paper use $\tau_s = 0.5$ m. Note that this step subsamples only the poses used as the reference path; all features remain in the map unless they were observed in only one stereo pair. The frame-to-frame tracking process used to build the map is the best possible condition for tracking features; pose changes between images are small, and the lighting is consistent. If a feature was unable to be tracked in the teach pass, it is unlikely to be seen in the repeat pass.

The subsampled reference poses give the path of the camera in $\underrightarrow{\mathcal{F}}_m$, but our path tracker controls the position of the vehicle, not the camera. We compute the vehicle position using the rotation and translation between the camera and vehicle frames: $\mathbf{C}_{c_k,v_k}$ and $\boldsymbol{\rho}_{v_k}^{c_k,v_k}$. The reference path of the vehicle, $\boldsymbol{\rho}_m^{v_k,m}$, is

$$\boldsymbol{\rho}_m^{v_k,m} := \left(\boldsymbol{\rho}_m^{c_k,m} - \mathbf{C}_{m,c_k}\mathbf{C}_{c_k,v_k}\boldsymbol{\rho}_{v_k}^{c_k,v_k}\right), \tag{6}$$

and its attitude is

$$\mathbf{C}_{v_k,m} = \mathbf{C}_{c_k,v_k}^T\mathbf{C}_{c_k,m}. \tag{7}$$

The projection from three dimensions to two is defined by fitting a plane to the feature points in the current submap. The subsampled features have each passed the consistency test of outlier detection, and so they represent a reasonable, sparse reconstruction of the local area. For each feature i, at position $\mathbf{q}_m^{i,m}$ in the submap, we find d_i, the minimum distance between the feature and one of the vehicle reference poses:

$$d_i := \min_k \left\| \boldsymbol{\rho}_m^{v_k,m} - \mathbf{q}_m^{i,m} \right\|.$$

From this distance, we compute a weight, w_i, used in the plane fitting:

$$w_i = \begin{cases} \dfrac{1}{d_i + \sigma_p} & \text{if } d_i \leq \tau_d \\ 0 & \text{otherwise} \end{cases}. \tag{8}$$

This weighting term is designed to ensure that the plane fit captures the local ground plane directly along the path that the rover has already traversed. The threshold, τ_d, ensures that features outside of the vehicle corridor are not used for the plane fit, and σ_p controls the rolloff of weights as features approach the edge of the vehicle corridor. For all experiments in this paper, we use $\sigma_d = 0.01$ and $\tau_d = 1.5$. We parameterize the plane by a unit vector, $\mathbf{n}$, and offset, b, such that any point $\mathbf{x}$ on the plane satisfies

$$\mathbf{n}^T\mathbf{x} + b = 0.$$

From this equation, we define a weighted least-squares problem to solve for $\mathbf{n}$ and b by minimizing J_p:

$$J_p = \frac{1}{2}\sum_{i=1}^{N} w_i\left(\mathbf{n}^T\mathbf{q}_m^{i,m} + b\right)^2 - \frac{1}{2}\lambda(\mathbf{n}^T\mathbf{n} - 1), \tag{9}$$

where N is the number of features in the submap and λ is a Lagrange multiplier that ensures that $\mathbf{n}$ is a unit vector. Solving for the minimum of this equation results in the eigenvalue problem

$$A\mathbf{n}^\star = -\lambda\mathbf{n}^\star,$$

where

$$A := \sum_{i=1}^{N} w_i\left(\mathbf{q}_m^{i,m}\right)\left(\mathbf{q}_m^{i,m}\right)^T - \frac{1}{W}\left(\sum_{i=1}^{N} w_i\mathbf{q}_m^{i,m}\right)\left(\sum_{i=1}^{N} w_i\mathbf{q}_m^{i,m}\right)^T,$$

$$W := \sum_{i=1}^{N} w_i,$$

and $\mathbf{n}^\star$, the unit vector that minimizes J_p, is the eigenvector of $\mathbf{A}$ corresponding to its minimum eigenvalue. Figure 8

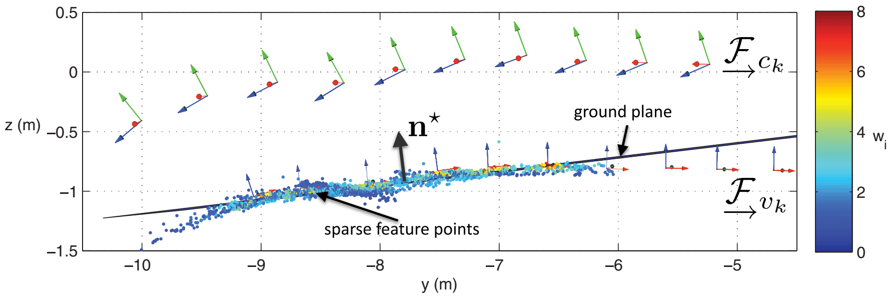


Figure 8. Side view of a submap showing the camera frames, $\underrightarrow{\mathcal{F}}_{c_k}$, the vehicle frames, $\underrightarrow{\mathcal{F}}_{v_k}$, the sparse feature points, and the ground plane fit to the features.

illustrates this process, showing the camera and vehicle poses, the weighted sparse feature points, and the resulting plane fit.

The unit vector $\mathbf{n}^\star$ is the normal of the xy plane of the projection frame, $\underrightarrow{\mathcal{F}}_p$, expressed in $\underrightarrow{\mathcal{F}}_m$. We now calculate the rotation, $\mathbf{C}_{m,p}$, that transforms vectors from $\underrightarrow{\mathcal{F}}_p$ to $\underrightarrow{\mathcal{F}}_m$. Using the shorthand $c_a := \cos(a)$ and $s_b := \sin(b)$, the rotation $\mathbf{C}_{m,p}$ may be parameterized by Euler angles, (α, β, γ), such that

$$\mathbf{C}_{m,p} = \begin{bmatrix} c_\alpha c_\beta & s_\alpha c_\beta & -s_\beta \\ c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & -s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & c_\beta s_\gamma \\ c_\alpha s_\beta c_\gamma + s_\alpha c_\gamma & -s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & c_\beta c_\gamma \end{bmatrix}. \quad (10)$$

We know that $\mathbf{n}^\star$ expressed in $\underrightarrow{\mathcal{F}}_p$ is $[0\ \ 0\ \ 1]^T$, which leads to the following constraint:

$$\mathbf{n}^\star = \mathbf{C}_{m,p} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -s_\beta \\ c_\beta s_\gamma \\ c_\beta c_\gamma \end{bmatrix}.$$

Defining the components of $\mathbf{n}^\star =: [n_1\ \ n_2\ \ n_3]^T$, we can solve for β and γ:

$$\beta = \operatorname{asin}(-n_1), \quad (11)$$

$$\gamma = \operatorname{atan2}(c_\beta n_2, c_\beta n_3). \quad (12)$$

The last Euler angle, α, is ambiguous (the plane normal is only a two-degree-of-freedom constraint), so we introduce a final constraint that the x axis of $\underrightarrow{\mathcal{F}}_{v_0}$ lies in the xz plane of $\underrightarrow{\mathcal{F}}_p$. Using $\mathbf{C}_{m,p}$ and the vehicle path from Eqs. (6) and (7), we can transform the reference path to the projection frame:

$$\boldsymbol{\rho}_m^{v_k,v_0} = \boldsymbol{\rho}_m^{v_k,m} + \mathbf{C}_{c_0,v_0}\boldsymbol{\rho}_{v_0}^{c_0,v_0}, \quad (13)$$

$$\boldsymbol{\rho}_p^{v_k,v_0} = \mathbf{C}_{m,p}^T \boldsymbol{\rho}_m^{v_k,v_0}, \quad (14)$$

$$\mathbf{C}_{v_k,p} = \mathbf{C}_{v_k,m}\mathbf{C}_{m,p}. \quad (15)$$

Finally, we compute a scalar difficulty score for the submap. During the repeat pass, the difficulty level is used to choose the robot's repeat speed to use on a given map. We compute a measure of curvature of the reference path as it captures two common forms of path difficulty: (1) tight turns and (2) rough terrain. To this end, we compute the incremental attitude changes of the camera, $\delta\mathbf{C}_k$:

$$\delta\mathbf{C}_k = \mathbf{C}_{c_{k-1},m}\mathbf{C}_{c_k,m}^T.$$

This attitude change is decomposed into an axis of rotation, $\hat{\mathbf{a}}_k$, and an angle of rotation, ω_k. The difficulty, h, is then computed as the root-mean-square attitude change:

$$h = \sqrt{\frac{1}{M}\sum_{k=1}^{M}\omega_k^2}, \quad (16)$$

where M is the number of reference poses.

When building reference trajectories with a fixed length and a fixed spacing of reference poses, M is very consistent across submaps. Furthermore, over these very short distances, the relative pose estimates are very accurate. Because of this, the values of h from Eq. (16) are comparable between submaps. The dependence of driving speed on terrain difficulty must be tuned for each vehicle/application combination. Table I lists the speed schedule used for all experiments in this paper.

Table I. Driving speed used as a function of the difficulty (RMS attitude change).

Difficulty range (deg)	Speed (m/s)
$h < 1.0$	1.00
$1.0 \le h < 2.0$	0.75
$3.5 \le h < 8.5$	0.50
$8.5 \le h$	0.35

The driving-speed schedule for a deployment of this algorithm would have to be tuned for each vehicle/application combination.

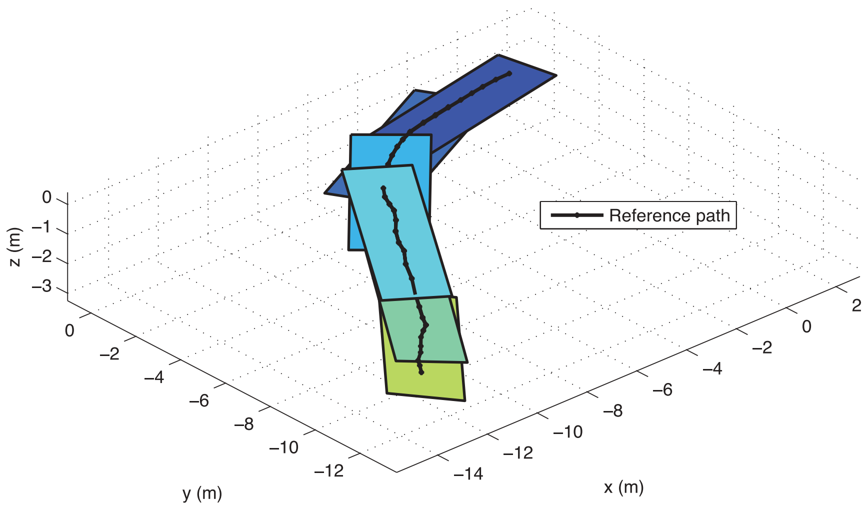


Figure 9. A view of six overlapping submaps with the reference path plotted above.

At this point, the submap is saved to disk with the following information:

- a vehicle reference path with L poses (indexed by ℓ), $\{\boldsymbol{\rho}_p^{\ell,p}\}$, expressed in $\underrightarrow{\mathcal{F}}_p$, calculated from Eqs. (6), (13), and (14)
- a rotation $\mathbf{C}_{p,m}$ that defines the projection to a local ground plane, calculated from Eqs. (11), (12), and (10)
- a set of N features (indexed by i), $\{\mathbf{q}_m^{i,m}, \mathbf{v}_i\}$, expressed in $\underrightarrow{\mathcal{F}}_m$
- a scalar difficulty score, h, computed from Eq. (16)
- flags that indicate whether the beginning or end of the map is broken

This satisfies the requirements needed to be used as a feature database in the generic localization module described in Section 3.2.

Each submap is between 500 KB and 2 MB, depending on the number of features tracked (which is scene dependent). This size includes extra data that are used solely for algorithm evaluation and not to repeat the route. Averaged over all teach passes, this amounts to 348 MB per kilometer. The teach pass processes an image approximately every 0.2 m, 5,000 images per kilometer. An appearance-based approach using the rectified stereo images would occupy 2.9 GB per kilometer, and saving all of the keypoints and descriptors would take up 1.3 GB per kilometer (assuming 500 stereo keypoints per frame). By aggregating data, our system offers an order-of-magnitude savings in storage over a pure appearance-based approach.

After saving the submap to disk, older poses and features are removed from the database in memory. We build the submaps to overlap by 50% as this ensures data overlap during transitions (Marshall et al., 2008). Poses are removed from the reference path until it is half of the length saved to disk. Any feature not seen by the remaining poses is then removed from the feature database. After this step, the mapping loop continues, processing new keypoint lists, localizing against the feature database, and adding features to the map, until another split is triggered or the image sequence ends. Figure 9 shows a short section of a map database, the ground plane of each submap, and the reference path. When the teach pass is complete, a database of maps is available for use in the repeat pass.

3.4. Route Repeating

During the repeat pass, the robot uses the database of submaps to repeat the route. The system we have implemented can start at any place along the path and repeat the route in either direction, provided that the camera is facing the same direction it was facing during the teach pass. Neither direction switching during path following nor local obstacle detection has been implemented, although both should be possible (Marshall et al., 2008). This section will describe the route-following algorithm in detail: localization, route management, and failure handling.

3.4.1. Repeat Pass Localization

Three specializations of the generic localization module are used during the repeat pass: *submap selection*, *localization*, and *VO*.

Submap selection is performed at the start of a route or when the robot is lost. One of the submaps built in the teach pass is loaded into memory and used as a feature database. Features are tracked and subjected to outlier detection. If there are enough inlying feature tracks (nine for all experiments in this paper), the objective function used in the route-teaching phase (5) is used to solve for the pose of the camera. If this process is successful, the rover begins to repeat the route, interleaving localization and VO as the route is retraced.

The interleaving of localization and VO during the repeat pass is one of the key strategies that makes this system robust to lighting changes, scene changes, and occlusions. Our first iteration of this project used a formulation similar to that of Royer et al. (2007) or Šegvić et al. (2009)—all estimation of the robot's position was based on localization against the map, and no form of dead reckoning was used. This worked well on pavement and in urban environments, but when we tested our system on grass and rough terrain, the system failed too easily under changing lighting conditions. Realizing that our localization module was based on VO (a purely relative motion estimation method), we implemented a system that would switch back and forth between VO and localization. VO is accurate enough to keep the rover near the path through difficult areas, and periodic localization maintains the global (topological) accuracy that allows long routes to be repeated. This is similar to the method used by Zhang and Kleeman (2009), who use wheel odometry between their global corrections. We process VO every frame, but given our current hardware, there are not enough computational resources to also perform localization every frame. Hence, we introduce an integer parameter G and attempt localization only when $\mathrm{mod}(k, G) = 0$ (every Gth frame). In these experiments we have used $G = 3$. We use the frame-to-frame VO method described by Maimone, Cheng, and Matthies (2007).

Localization is similar to submap selection, but our prior knowledge of the rover's position (from VO) allows us to improve on the position estimate. Using only the process described for submap selection, our system would periodically localize only using distant features. In these cases, the orientation was estimated quite well, but the position of the rover would experience huge jumps. Similar behavior is described by Diosi, Remazeilles, Segvic, and Chaumette (2007). To account for this, a prior information term is added to the error term used to estimate the pose. Let J_{vis} be the error term from Eq. (5). We add prior information error terms for the position, J_{pos}, and attitude, J_{att}, so that the error term we minimize, J, is

$$J = J_{\mathrm{vis}} + J_{\mathrm{pos}} + J_{\mathrm{att}}. \tag{17}$$

Let $\hat{\boldsymbol{\rho}}_m^{c_k,m}$ and $\hat{C}_{c_k,m}$ be the position and attitude estimated by VO, and let $\boldsymbol{\rho}_m^{c_k,m}$ and $\mathbf{C}_{c_k,m}$ be the position and attitude we are estimating. In this notation,

$$J_{\mathrm{pos}} := \frac{1}{2}\left(\hat{\boldsymbol{\rho}}_m^{c_k,m} - \boldsymbol{\rho}_m^{c_k,m}\right)^T \mathbf{W}_{\mathrm{pos}}\left(\hat{\boldsymbol{\rho}}_m^{c_k,m} - \boldsymbol{\rho}_m^{c_k,m}\right).$$

Expressing $\hat{\mathbf{C}}_{c_k,m}$ and $\mathbf{C}_{c_k,m}$ as yaw–pitch–roll Euler-angle vectors, $\hat{\boldsymbol{\alpha}}_k$ and $\boldsymbol{\alpha}_k$, respectively, results in a similar error term for attitude:

$$J_{\mathrm{att}} := \frac{1}{2}\left(\hat{\boldsymbol{\alpha}}_k - \boldsymbol{\alpha}_k\right)^T \mathbf{W}_{\mathrm{att}}\left(\hat{\boldsymbol{\alpha}}_k - \boldsymbol{\alpha}_k\right).$$

The weighting matrices were chosen to be

$$\mathbf{W}_{\mathrm{pos}} := \frac{1}{\sigma_{\mathrm{pos}}^2}\mathbf{1}, \quad \mathbf{W}_{\mathrm{att}} := \frac{\mathbf{1}}{\sigma_{\mathrm{att}}^2}\mathbf{1}, \tag{18}$$

where $\mathbf{1}$ is the identity matrix. All experiments in this work use $\sigma_{\mathrm{pos}} = 1.0$ and $\sigma_{\mathrm{att}} = 0.1$, which results in a very weak prior. Finally, Eq. (17) is linearized and solved using the Gauss–Newton method.

The output of the localization block is an estimate of the camera's position, $\boldsymbol{\rho}_m^{c_k,m}$, and attitude, $\mathbf{C}_{c_k,m}$. Equations (6), (13), and (14) are then used to produce $\boldsymbol{\rho}_p^{v_k,p}$, the position of the vehicle in the projection frame. The attitude of the vehicle in the projection frame, $\mathbf{C}_{p,v_k}$ is computed using Eq. (15), then decomposed into a yaw–pitch–roll Euler-angle sequence. The yaw value of this sequence is the vehicle's heading in the projection frame, θ_k. Defining the components, $\boldsymbol{\rho}_p^{v_k,p} =: [x_k \;\; y_k \;\; z_k]^T$, we can express the two-dimensional robot pose, $\boldsymbol{\varrho}_k$:

$$\boldsymbol{\varrho}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}. \tag{19}$$

This planar pose of the robot and the projected reference path are passed to a unicycle-model version of the planar path-tracking algorithm described by Marshall et al. (2008).

3.4.2. Repeat Pass Route Management

The localization module feeds into a route management system that triggers map handoffs, schedules the robot's speed based on the path difficulty, and monitors the route-following system for errors. The route manager tracks the closest point on the current reference path. When the vehicle reaches the middle of a reference path, a map handoff is triggered. This involves the following steps:

– loading the next submap from disk
– updating the feature database used for localization
– updating the reference path used by the path tracker
– updating the transformation from $\underrightarrow{\mathcal{F}}_m$ to $\underrightarrow{\mathcal{F}}_p$
– setting the robot's speed based on the submap difficulty

3.4.3. Repeat Pass Failure Handling

Route-following failures are detected by monitoring the distance traveled since the last successful localization. When this distance reaches a threshold, τ_g, the rover stops and the system attempts to recover from the failure. To recover, the system signals the operator that there has been a failure and then searches nearby (topologically) submaps using the submap selection mode to perform wide-baseline matching. If this reinitialization is successful, the rover continues the route. During the search phase, an operator may also reposition the rover on the path (using images from the teach pass images to identify the correct position). Any

Table II. The major parameters of our teach-and-repeat algorithm, a description of their functions, and the intuition behind the choice of value.

Parameter	Value	Description
τ_f	45%	(Section 3.3.1) The current set of features is added to the map when the percentage of features tracked from the preceding frame drops below τ_f. Using this threshold avoids generating bloated maps when the robot is sitting still and automatically adjusts the number of features per map based on the difficulty of feature tracking on specific terrain.
τ_l	5 m	(Section 3.3.1) The length of the reference path for each submap. Larger values increase the computational complexity of searching the map for feature correspondences. Smaller values make the system more prone to failure in the face of localization errors.
τ_s	0.5 m	(Section 3.3.2) The spacing of poses in the reference path. This controls the fidelity of the path used by the path tracker and the fidelity of the difficulty metric (16).
τ_d	1.5 m	(Section 3.3.2) Controls the width of the vehicle corridor used to build the local ground plane for each submap in Eq. (8). This was tuned for the width of the vehicle.
σ_p	0.01	(Section 3.3.2) Controls the weighting rolloff as features approach the edge of the vehicle corridor in the plane fit (8). Larger values weight features near the edge of the corridor more. This parameter must be tuned for the expected clutter of the environment.
G	3	(Section 3.4.1) During the repeat pass, localization is processed every G frames (VO is processed every frame). This value would be 1 if possible, causing localization every frame. Higher values reduce the computational complexity of the algorithm.
$\{\sigma_{\text{pos}}, \sigma_{\text{att}}\}$	{1.0, 0.1}	(Section 3.4.1) These terms determine how much to rely on the prior pose estimate (from VO) during localization (18). The values chosen result in a weak prior that significantly changes the solution only when localization against the map is uncertain (e.g., when localizing with a small number of distant features).
τ_g	50 m	(Section 3.4.3) The distance to travel without successful localization before stopping, signaling the operator, and searching nearby submaps in an attempt to relocalize. This is based on the estimated accuracy of our VO implementation.

failures or repositioning of the rover will be noted in our experiments below.

When the algorithm encounters a break in the map as described in Section 3.3.1, the system drives to the end of the current submap, stops, loads the next submap into memory, and attempts to localize. If this is successful, the algorithm restarts the rover and continues repeating the route. If this fails, the rover will signal the operator for intervention. The algorithm will then start searching nearby maps (topologically) until submap selection is successful. The operator can then choose to reposition the rover or command it to continue using VO.

3.5. Parameter Choices

As in many robotics applications, a number of parameters must be tuned for each deployment. The parameters of our teach-and-repeat system were tuned during algorithm development and then fixed for the experiments reported in this paper. Throughout the algorithm description above, we have tried to elaborate on the intuition behind each of our parameter choices. For clarity, we summarize the main algorithm parameters in Table II, along with a description of their functions and some notes about the intuition used to select the parameter value.

3.6. Hardware

The experiments described in this paper were performed using the six-wheeled articulated rover shown in Figure 10. Motor control on the rover was performed by a pair of microcontrollers. Vehicle-level motion commands and path

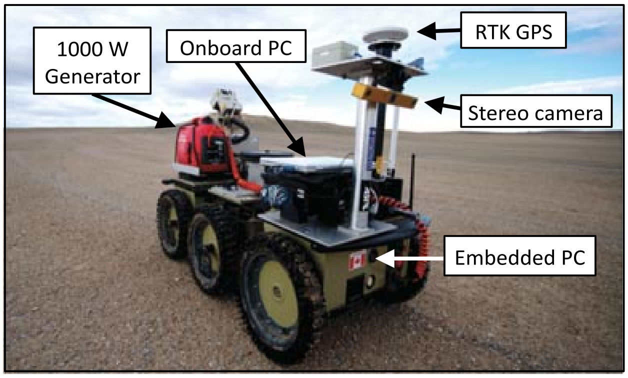


Figure 10. The six-wheeled rover platform used in these experiments was outfitted with a stereo camera, an onboard PC to run the teach-and-repeat algorithm, an embedded PC to process path tracking, an RTK GPS for benchmarking localization performance, and a 1,000-W gas generator to provide power during multiple-kilometer traverses.

tracking were handled by a single embedded PC with a 1.2-GHz Pentium 4 processor and 1 GB of ram. The base was powered by three lithium-ion battery packs, but in order to perform the long-range experiments in this paper, the onboard power supply was augmented with a Honda 1,000-W generator, which supplied power to the base and all of the onboard computers. The computer running the localization and route management was a MacBook Pro with a 2.4-GHz Intel Core 2 Duo processor, 4 GB of RAM, and an NVIDIA GeForce 8600M GT graphics card capable of supporting CUDA 1.1. The stereo camera was a Point Grey Research Bumblebee XB3 with a 24-cm baseline and 70-deg field of view, mounted approximately 1 m above the surface pointing downward by approximately 20 deg. Each image of the stereo pair was captured at 640 × 480 pixel resolution. When possible, we used a pair of Thales DG-16 real-time kinematic (RTK) GPS units for ground-truth evaluation. These units are rated at 0.4-m circular error probable (50% of the data should be within a circular area of this radius around the true value). Unfortunately, our radio link was not robust to occlusions; for long routes and near buildings, it was not possible to receive the real-time corrections, and only regular GPS was available. The rover was able to track a path while driving forward or backward. This allowed us to repeat routes in either direction by keeping the camera facing the same direction as it was during route learning.

4. FIELD TESTING

We conducted a number of field trials to test the capabilities of the full teach-and-repeat system and characterize the performance of the localization system. This section will summarize the results of our route-following tests. Our preliminary tests were performed at the University of Toronto Institute for Aerospace Studies (UTIAS). Because of the applicability of our algorithm to planetary exploration, we conducted trials at the Haughton-Mars Project Research Station (HMP-RS) on Devon Island in the Canadian High Arctic (Lee, Braham, Boucher, Schutt, Glass, et al., 2007). The HMP-RS is located within a polar desert, which offers an unusually wide variety of geological features of strong planetary-analog value. Because of this, it has been used for rover testing in the past (Fong, Deans, Lee, & Bualat, 2007; Fong, Allan, Bouyssounouse, Bualat, Deans, et al., 2008; Wettergreen, Dias, Shamah, Teza, Tompkins, et al., 2002; Wettergreen, Tompkins, Urmson, Wagner, & Whittaker, 2005). Additionally, the lack of vegetation, low angle of the sun in the sky, and wide range of terrain types make it an ideal site for testing of vision-based algorithms for planetary exploration.

4.1. Route Following

Our teach-and-repeat system has been tested on 27 routes and more than 32 km of autonomous driving. Results reported in this paper are for the algorithm described in Section 3. Earlier route-following results during the algorithm's development are not included. All tests described here were performed using the same code and parameters. Individual teach passes are named according to the convention shown in Figure 11. Experiments performed at UTIAS are marked with a u, and those from Devon Island are marked with a d. We used two methods to teach routes. The h tag indicates that the rover was piloted by a human, and the a tag indicates that the rover was driving autonomously. The autonomous teach passes were recorded during trials of a terrain assessment and path-planning algorithm. When the terrain assessment algorithm signaled that its run was complete, the route was taught from logged images, and the rover autonomously returned along its path. Some routes were taught with the camera facing forward, and others with the camera facing backward (as required by other concurrently running experiments). However, during route repeating, the rover drove forward or backward as necessary to keep the camera pointed in the same direction as it was during route learning. Complete statistics for all teach-and-repeat passes are given in Appendix B, but the overall results will be summarized here.

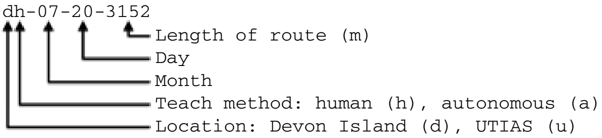

dh-07-20-3152
Length of route (m)
Day
Month
Teach method: human (h), autonomous (a)
Location: Devon Island (d), UTIAS (u)

Figure 11. The naming convention used for teach passes.

Learned routes ranged in length from 47 m to nearly 5 km. Of the 27 teach passes, 21 successfully built maps without failure. The teach pass failures will be discussed in greater detail in Section 5.5. The difficulty of the routes was assessed using an inclinometer to measure the vehicle-frame pitch and roll and GPS to measure the relative elevation change. During the most extreme routes, the rover experienced up to 118.5 m of elevation change, as well as pitch and roll deviation from vertical of up to 28 and 22 deg, respectively.

The 27 teach passes were used to perform 60 repeat passes. Only four of the routes required manual interventions. Four repeat passes were not completed to the end. The repeat pass failures will be discussed in greater detail in Section 5.6. The longest autonomous repeat pass was 3.2 km (dh-07-23-4963). There were 2 autonomous runs of approximately 2 km (dh-07-20-2120) and 10 autonomous runs approximately 1 km long (uh-05-20-1152, uh-05-21-1170, and dh-07-22-1091). Of the 32.919 km traveled, only 0.128 km was piloted manually. This is an autonomy rate of 99.6%. In all cases in which the rover needed an intervention, it stopped along the path and signaled the operator.

4.2. Route with Large 3D Motion and Extreme Lighting Changes

To test the operational limits of our algorithm, we built a route at UTIAS where the rover experienced large 3D motion and extreme lighting changes. Figure 12 shows an overhead view of the route, the vehicle-frame pitch and roll (as measured by an inclinometer), and some representative views from the left camera. The rover started inside our indoor test facility on a raised platform. It descended a slope, climbed two hills, ascended a ramp, and then drove through a narrow corridor leading outdoors. There, it traversed an obstacle course, crossed a road, and finished the route by parking in our laboratory. The platform experienced pitch and roll up to 27 deg and moved from an indoor, low-light environment to outdoors. Multimedia Extension 1 (Appendix A) is a video of the rover repeating this route.

We taught this route twice, once during development of the obstacles (uh-07-22-0120), and once after they were complete (uh-07-23-0120). The routes were repeated seven and five times, respectively. Every repeat pass was successful, despite the 3D motion of the camera. Figure 13 shows the teach pass corridor (the track of the teach pass laterally extended ±2 m for illustration) with the tracks of the seven repeat passes overlaid. The background is shaded (blue) behind sections of the route where the algorithm experienced localization dropouts. Steep hills in the indoor section caused localization and VO failures due to significant motion blur. At the end of the route, computers and chairs were moved around, changing the appearance of the scene significantly. The experiment highlights the interplay of localization and VO. Where VO fails, localization corrects the error and keeps the rover on the correct path. Where localization fails, VO is accurate enough to carry the rover through to a place it recognizes.

This experiment was performed before our field trials on Devon Island to prove that the teach-and-repeat system would work on 3D terrain. During our field trials we tested the algorithm over many 3D routes (see the list of the most extreme routes in Table III). In all cases, 3D motion of the camera was not a limiting factor for route following.

5. EVALUATION

In this section we offer some evaluation of our algorithm to try to describe why it works and what its strengths and shortcomings are. We examine the convergence properties of the localization algorithm and the properties of the algorithm under changing lighting conditions. We compare the estimated lateral path-tracking error to that measured by GPS and look at which features are used for localization. Finally, we examine the failure modes experienced by the algorithm.

5.1. Convergence Properties

To test the convergence properties of the localization algorithm, we taught a single map on characteristic terrain (from the Devon Island experiments) using a camera on a tripod. After processing the teach pass, the camera was placed in a nominal position in the middle of the map, set to process localization, and perturbed from this nominal position until the localization failed. Perturbations were introduced four ways: as lateral displacements from the path center (0.1-m increments) and along vehicle-frame yaw, pitch, and roll axes (5-deg increments). At each increment, 200 localizations were processed.

Figure 14 shows the mean inlying feature count for lateral and angular deviations. The curves end when the localization algorithm fails. This experiment shows that the feature count decreases rapidly from the camera's nominal placement. Any curve of this type will be scene dependent, and we believe that the slower drop in feature count with positive lateral displacement may be due to prominent rocks to the right of the path. The experiment shows that localization is possible with up to ±1-m lateral displacement from the path and over ±20-deg angular deviation in all of yaw, pitch, and roll.

5.2. Lighting Dependence

We also designed an experiment to show the properties of our algorithm under changing lighting. The SURF feature description algorithm accounts for contrast changes by normalizing the description vector. However, in our experience, descriptor-based matching is very difficult under extreme lighting changes. To illustrate this, we taught a short route and set up a camera to capture an image and perform localization every 30 s. The inlying feature count is plotted against time passed in Figure 15.

Ten hours after the teach pass, the localization module fails to find enough inlying features. This confirms the lighting dependence that we have seen in our experiments. Strong lighting with a low angle of incidence is particularly problematic in this regard. Similarly, routes taught on overcast days and repeated on sunny days (or the other way around) cause problems. On overcast days, SURF's blob detector finds points based mainly on surface albedo, whereas during periods of strong lighting, shadowing creates areas of intensified image contrast based on scene structure. Different sets of point features are returned in each case.

5.3. Localization Performance during Path Following

This section will characterize the performance of our localization system during path following. Although the reconstruction of the route may not be globally consistent, each small section of the path should have a small

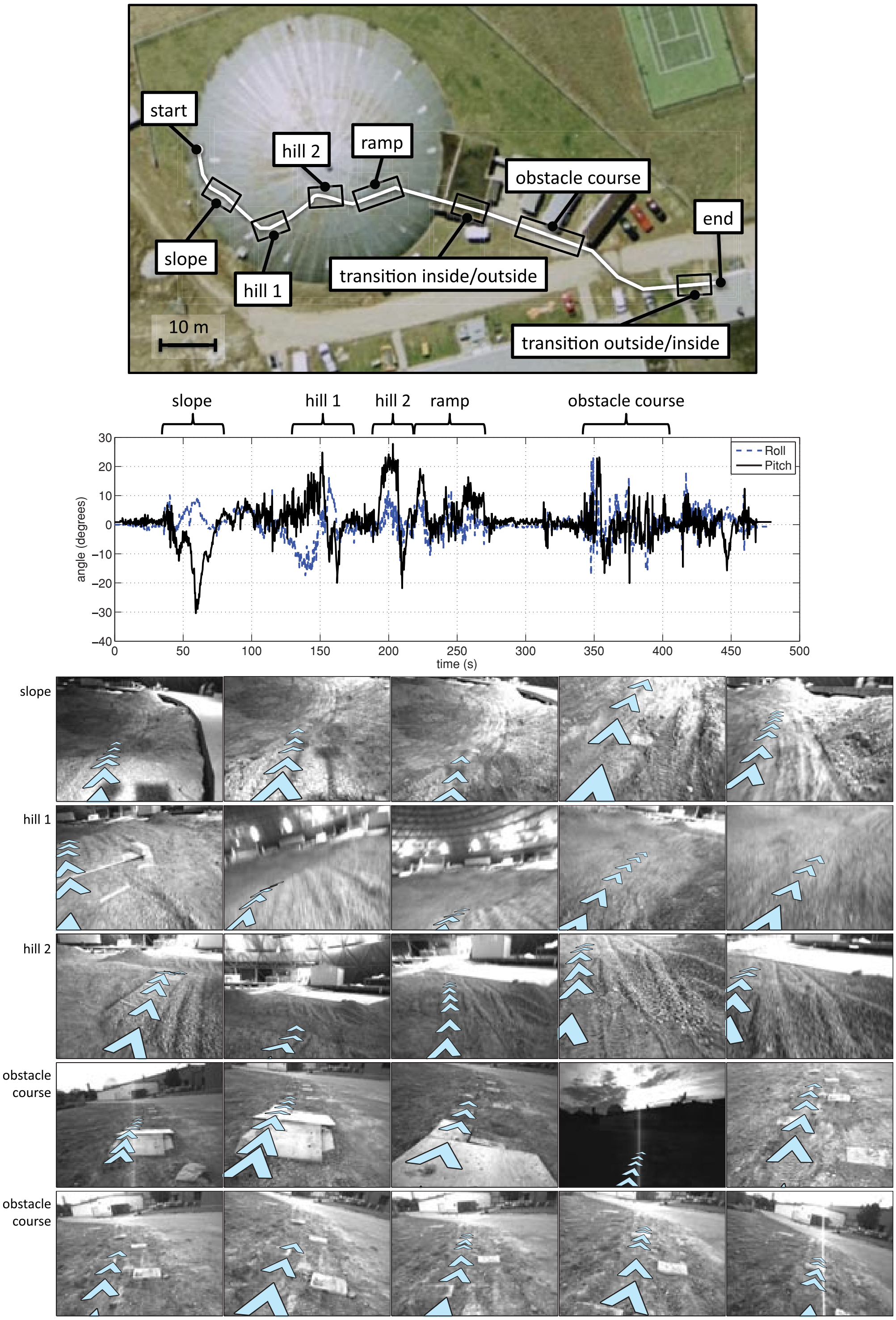


Figure 12. Top to bottom: An overhead view of the route built to test nonplanar camera motion and extreme lighting changes (uh-07-22-0120, uh-07-23-0120), the pitch and roll of the rover during the teach pass of route uh-07-22-0120, and short image sequences (left camera) from one repeat run of the route. The path, plotted as chevrons, confirms that localization is indeed performed in three dimensions.

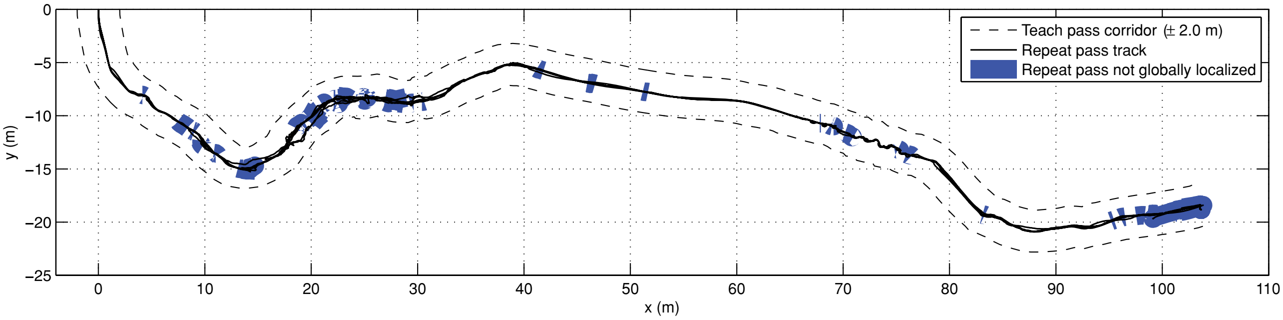


Figure 13. The seven repeat passes of route uh-07-22-0120 with the reference path from the teach pass extended laterally ±2 m. Localization dropouts, highlighted in blue, occurred mainly due to scene changes and motion blur.

reconstruction error. Because of this, we may compare the lateral path-tracking error estimated by the localization algorithm to that measured by GPS. Our GPS unit required line of sight between the base station and the rover to send the real-time corrections, and so we do not have RTK–GPS data for all routes. Figure 16 shows the lateral path-tracking error estimated by localization and measured by GPS over a 450-m-long segment of route da-07-29-0486. This segment of the route had RTK GPS for both the teach pass and the repeat pass. A shaded background highlights the portions of the repeat pass where localization has failed.

These figures show two important characteristics of our algorithm. First, when localization is successful, the estimated lateral path-tracking error has good agreement with the same quantity measured by GPS. When localized, none of the differences is larger than 0.2 m, agreement well within what we can discern with this GPS. Second, it shows that, when the algorithm is unable to globally localize, the estimate may diverge and then reconverge when localization is recovered. This is shown on Figure 16 at around 360 m traveled, where the localization drops out for nearly 15 m. The speed of divergence is a function of the accuracy of our VO algorithm. We have seen the algorithm recover from lateral path-tracking errors of 1.5 m and localization dropouts of up to 40 m. In each case, successful localization pulls the estimate back toward global consistency and allows our algorithm to faithfully repeat long routes.

5.4. Keypoint and Feature Usage

This section tries to shed some light on which keypoints and features are used by the algorithm to perform localization. To this end, we used data collected during the nine repeat passes of route uh-05-26-0202. This route was taught midday when it was overcast, and the first seven repeats were performed on a sunny day, every hour starting at 7:45 a.m. After the sixth repeat, cloud cover moved in and the additional runs were performed on a different day. The large number of repeats and varying lighting conditions make this route a good candidate for an examination of feature usage.

Figure 17(a) shows a histogram of track length (number of observations) for the 132,781 features stored in the map. The figure shows that maps are predominantly populated by features seen in only two images. From there, the track length decreases quite steeply but a small number of features are still seen many more times. The long tail of this curve has been truncated. The longest track length was 102 frames.

Table III. Difficulty metrics for the teach passes where the rover experienced the most extreme 3D camera motion.

Tag	Elevation change	Min roll	Max roll	Min pitch	Max pitch
uh-05-20-1152	4.9	−18.4	8.2	−12.0	6.3
uh-05-21-1170	4.9	−17.8	11.9	−14.6	10.3
uh-05-22-0120	3.6	−14.7	11.7	−21.5	27.2
uh-07-23-0120	4.9	−13.3	12.1	−18.5	26.5
dh-07-20-2120	69.2	−22.0	15.9	−28.3	16.9
dh-07-23-4963	118.5	−12.8	13.6	−15.5	12.0
dh-07-30-0347	2.1	−12.2	13.1	−12.1	10.8
dh-07-31-0192	1.1	−9.8	10.9	−17.9	19.1

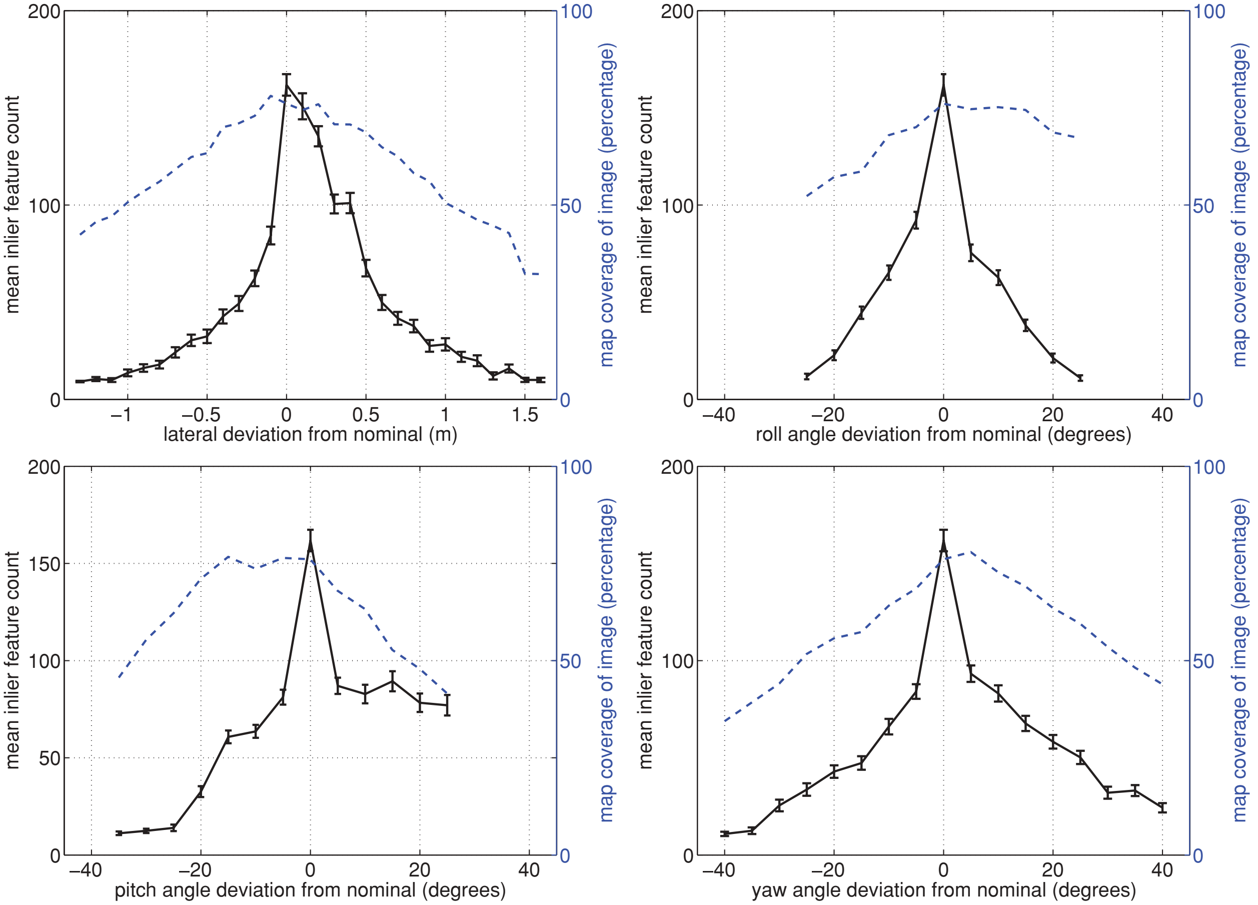


Figure 14. Average feature count of the localization algorithm (black, solid line with 1σ bounds) as the camera was displaced laterally from the path, or rotated in place. Each data point is averaged over 200 trials. Angular perturbations were made along the vehicle-frame yaw, pitch, and roll axes. The dotted curve shows the percentage of the image covered by the features in the map. Whereas the feature count is correlated to coverage, changes in viewpoint also reduce the ability of the system to associate features.

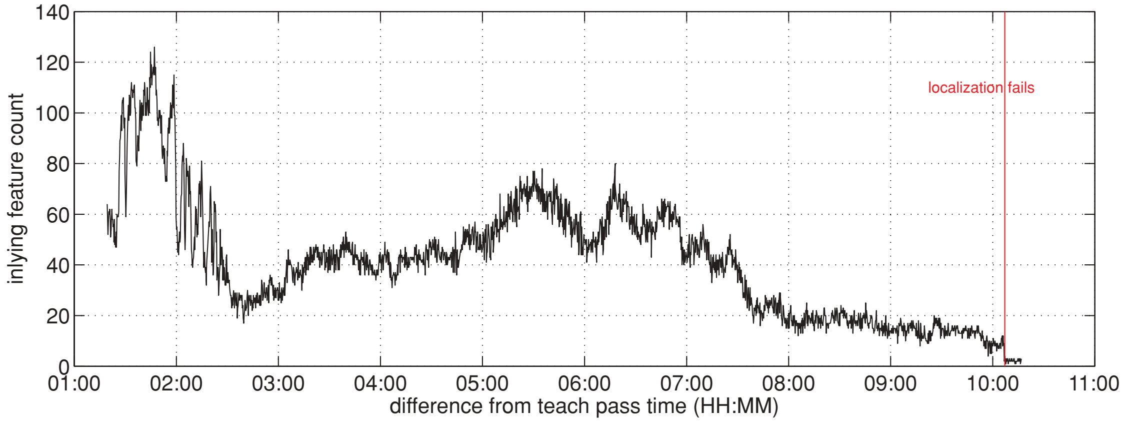


Figure 15. Results of testing the localization algorithm performance under changing lighting. Owing to time constraints during our field campaign, we were able to perform only a single trial. However, the result here fits very well with the results of our path-following experiments; the SURF feature matching is not robust to extreme lighting changes.

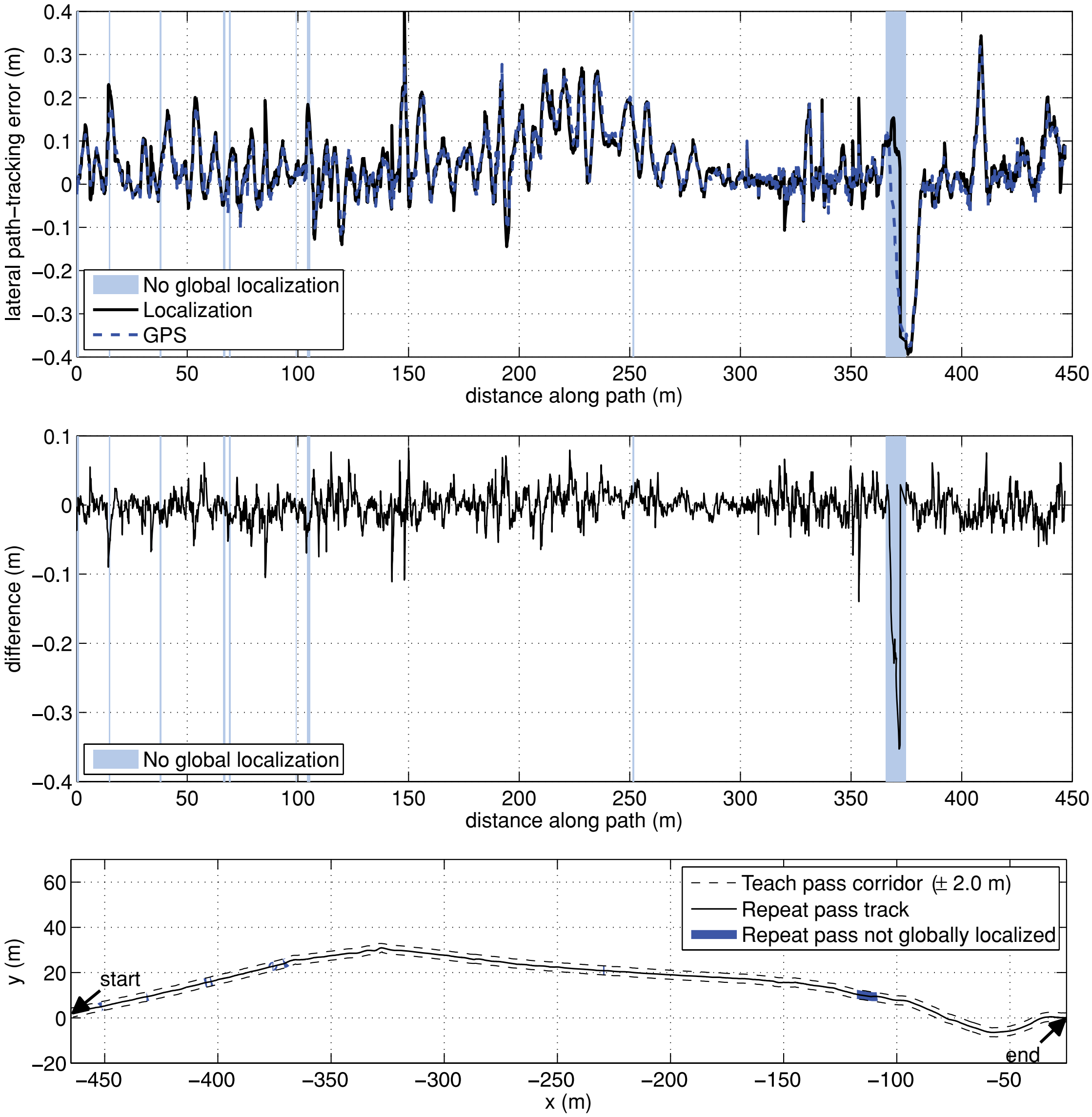


Figure 16. Top to bottom: Lateral path-tracking error during a repeat pass as estimated by the localization algorithm and measured by RTK GPS, the difference between these curves, and the track of the rover during this segment. The blue shaded background highlights areas where the localization step failed. When localization is successful, the pose estimate agrees well with GPS.

During the repeat pass, we logged which features were used for localization. Figure 17(b) shows the relationship of the track length during the teach pass to feature use during the repeat pass. Plotting the mean over all samples shows a strong linear relationship with slope 1. This confirms what intuition would suggest: that unique features seen for a long time during route learning are easily found during route repeating.

To determine which features contribute most to localization, we plotted the feature observations in image space. Figure 18(a) shows a clustering of features around the top of the image. When compared to a typical image from this route shown in Figure 18(b), it clearly shows that the majority of features used during the repeat pass are distant from the camera—horizon features. Horizon features are good for correcting for rover orientation but, as stereo-based range accuracy decreases with distance from the camera, they are not great for estimating the rover position. In this sense, our algorithm works a lot like VO with globally consistent orientation updates. This also suggests a way forward for future work; it may be possible to reduce the submap size by using only features that have been tracked

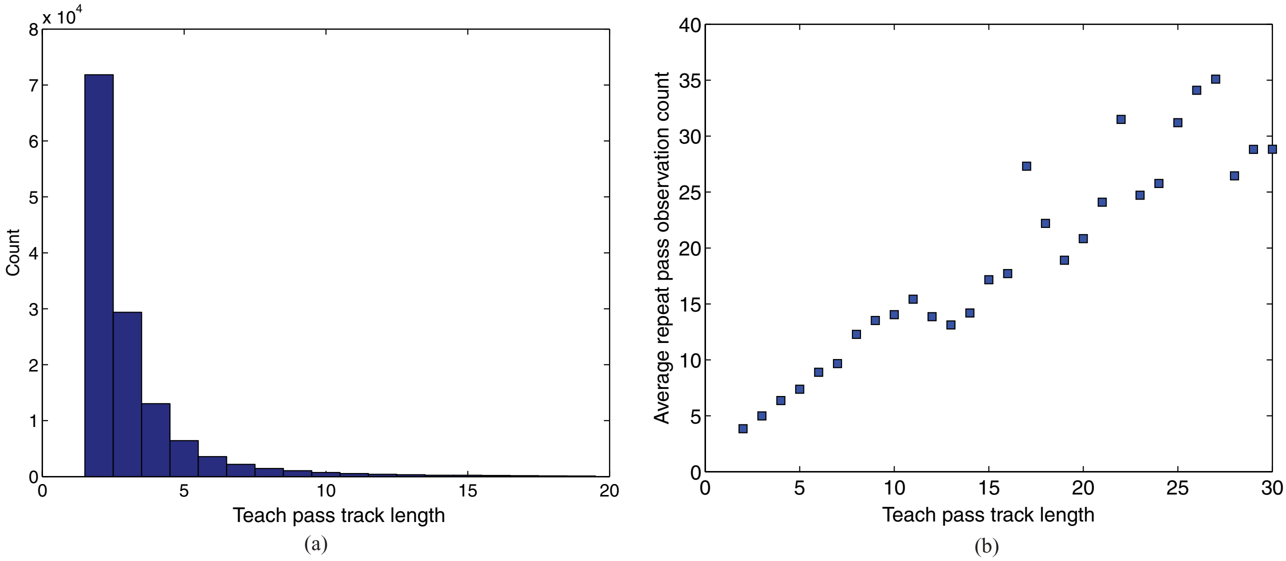


Figure 17. Histogram of feature track length for the 132,781 features tracked in the teach pass (a). The mean observation count during route repeating for each track length is shown in (b). Feature track length during route learning correlates strongly with the observation count during route repeating.

for multiple frames. This would reduce the computational complexity of finding feature correspondences in the map and enable localization to be performed more often. However, using only distant features would most likely require more accurate feature position and covariance estimates, suggesting the use of a multiframe reconstruction method during the teach pass. Alternately, the two-stage estimation algorithm described by Kaess, Ni, and Dellaert (2009) could be used to decouple the orientation and position estimation problems.

5.5. Teach Pass Failures

All of the teach pass failures listed in Table IV were due to large displacement of the camera between images. Sometimes a processing backup would cause our data-logging

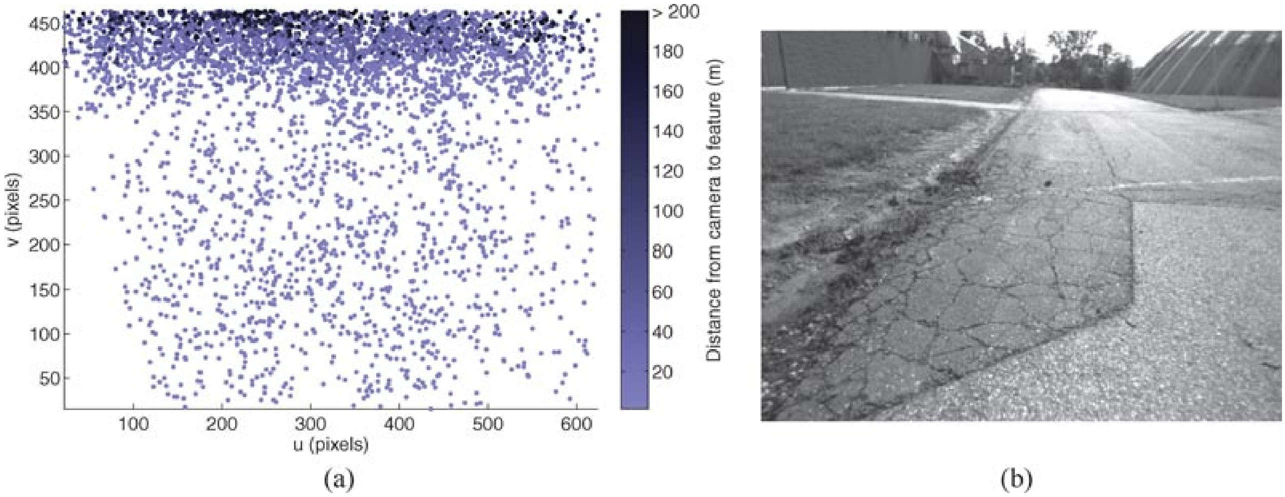


Figure 18. A plot of (a) the image locations of 240,534 feature observations from nine repeat passes and (b) a typical image from this sequence. The features used for localization cluster in a band at the horizon.

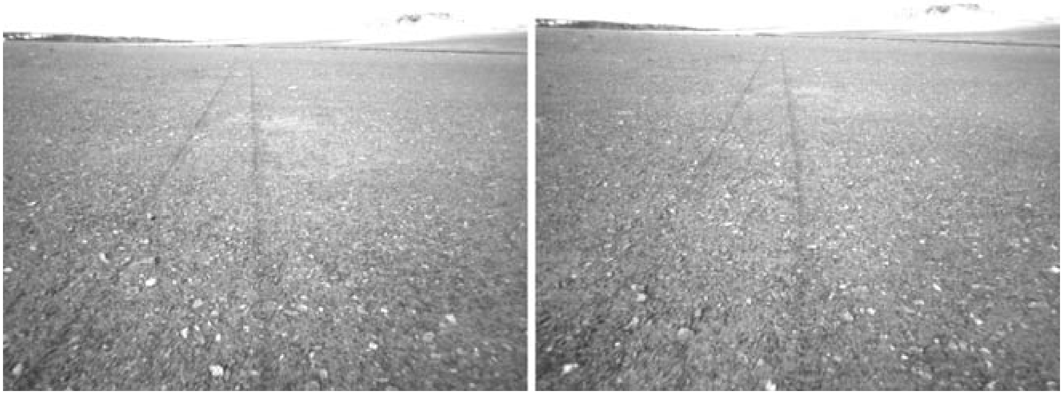


Figure 19. All teach pass failures were the result of large interimage spacing due to data logging dropouts. This figure shows two consecutive images from uh-07-23-4963 that caused a failure.

Table IV. Teach passes with failures.

Tag	Length (m)	Maps	Features per map	Failures
ua-06-04-0097	96.8	35	1,581	3
ua-06-06-0186	185.8	65	1,975	1
dh-07-20-2120	2,120.0	740	3,680	1
da-07-20-0464	464.0	166	1,662	1
da-07-21-0453	453.5	161	1,960	1
dh-07-23-4963	4,962.5	1,732	3,993	5

system to drop images. This was not a problem on many types of terrain, especially where there were strong horizon features or large objects out of the ground plane. However, on flat, repetitive terrain such as that seen in a long section of uh-07-23-4963, even short dropouts caused teaching failures. This is illustrated in Figure 19, which shows a pair of consecutive images from this route that caused a failure.

Three of the five routes with teach pass failures required no operator intervention. The rover simply drove to the end of the broken map, loaded the next map, relocalized, and continued.

5.6. Repeat Pass Failures

Table V lists all of the repeat pass failures and incomplete routes. Repeat pass failures had two distinct causes. The first was integration with an autonomous terrain assessment and path-planning algorithm, and the second was changing scene appearance.

The route-learning algorithm described in this paper had no problem learning an image sequence with direction switches, but paths that doubled back on themselves were not amenable to our path-tracking algorithm. Early in development, we decided not to implement direction switches. However, the autonomous terrain assessment and path-planning algorithm used to build some of the routes sometimes backed up along its own path to get out of a cul-de-sac. When faced with a knot in the path, the path tracker would command the robot to perform a wide U-turn, ending up 180 deg to the desired orientation on the path. To deal with this, we developed a preprocessing step that used the motion estimate from the terrain assessment run to automatically identify path knots and remove the images making up those knots from the sequence. We developed this preprocessing step during some tests at the beginning of June 2009 (ua-06-04-∗ and ua-06-06-∗). Failures during this time informed this development process, and the knot-removal step worked without fail after that.

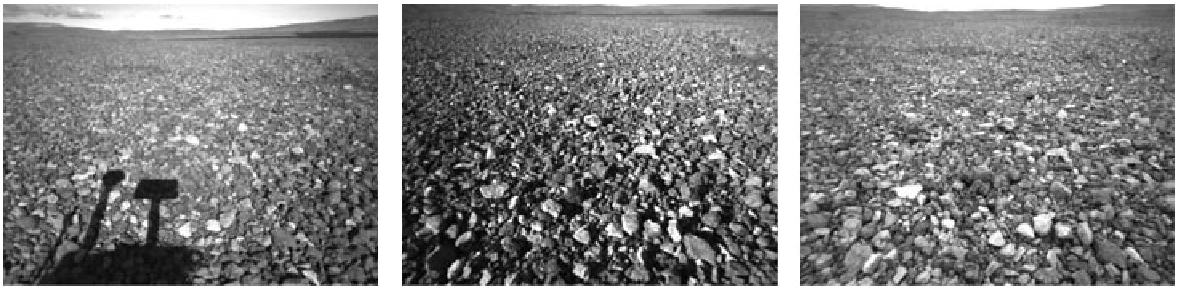
Other failures during repeat passes were due to the changing appearance of the scene, mostly because of changing lighting conditions. We encountered several situations in which a route required manual interventions (uh-05-21-1170) or failed to complete (dh-07-30-0187) at one time of day but was autonomously repeated successfully when the lighting changed. These results agree well with the lighting test in Section 5.2. Route uh-05-21-1170, taught at midday in direct sunlight, was repeated six times and had trouble only late in the evening or early in the morning. Route dh-07-30-0187 was in an area made up entirely of fist-sized rocks.[2] The complex shadows created by these rocks were difficult for our algorithm under time changes. Figure 20 shows an image from the teach pass of this route along with images from the failed and successful repeat passes.

Flat areas with repetitive texture were particularly difficult under changing lighting conditions. The section of route uh-07-23-4963 already shown in Figure 19 was taught when it was partly cloudy with some periods of strong direct sunlight, and both repeat passes were attempted when it was overcast. The first repeat pass was attempted forward along the route, and the second was attempted

[2] A video of the rover repeating route dh-07-31-0192 at the same site is available as Multimedia Extension 2 (Appendix A).

Table V. Repeat passes with failures.

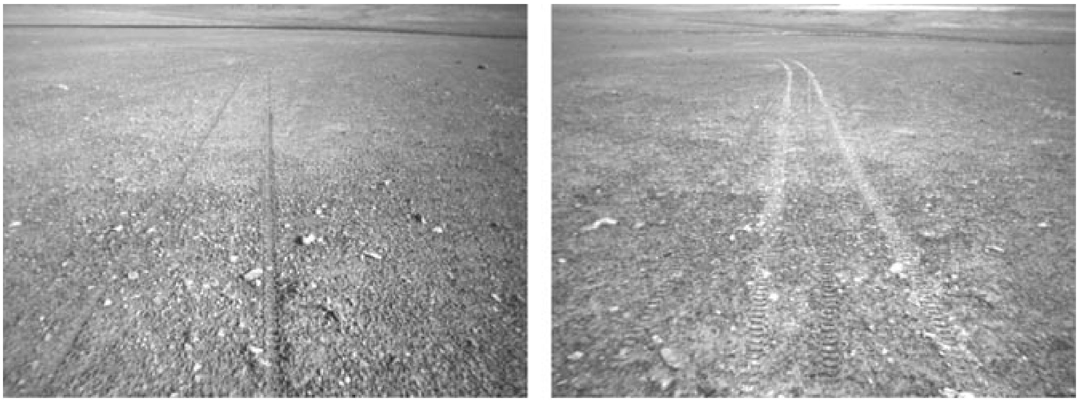
Teach pass tag	Teach pass start time	Repeat pass start time	Completed (%)	Autonomous (%)	Operator interventions	Globally localized (%)
uh-05-21-1170	12:16:02	20:26:28	100.0	92.0	1	49.6
		08:03:44	100.0	98.4	3	45.0
ua-06-04-0097	14:48:50	15:10:59	87.4	95.5	2	93.4
ua-06-06-0186	13:11:57	13:55:08	100.0	98.4	1	81.5
dh-07-23-4963	08:50:49	08:49:24	64.9	100.0	0	96.3
		15:07:49	32.8	100.0	0	76.2
dh-07-30-0187	14:35:09	18:37:18	78.4	100.0	0	9.9



(a) An image from the start of the teach pass (b) An image from the failed repeat pass started 4h later (c) An image from the successful repeat pass started the next morning

Figure 20. Images from the start of route dh-07-30-0187 that show the scene changes due to lighting.

backward. Both failed at either end of the same stretch of terrain. Figure 21 shows an image from the rover where it stopped on the first repeat pass and a corresponding image from the teach pass. It is clear from the image that the rover was no more than 0.5 m laterally displaced from the path after 50 m without localization. Still, although the viewpoint was nearly the same, the repetitive texture, different lighting, lack of horizon features, and lack of unique 3D objects in the scene were too much. At this point, we would have repositioned the rover onto the path or piloted it through this section manually but it started to rain. After an hour, we decided the rain would not let up, so we secured a tarp over the rover and piloted it home manually. As stated earlier, the second repeat pass came from the other direction.



(a) An image from the teach pass (b) The stopping position of the robot on the first repeat pass

Figure 21. Images from the teach-and-repeat passes of route uh-07-23-4963. The rover was unable to localize for 50 m even though it was clearly on the path for most of the way. Repetitive texture, different lighting, lack of horizon features, and lack of unique 3D objects in the scene were the major causes of localization dropouts.

This time the rover stopped at the teach pass failure shown in Figure 19 and was unable to relocalize. We commanded the robot to continue using VO, but it was unable to localize anywhere along the path. We were unable to find time in our test schedule to test the route under weather conditions similar to the teach pass.

6. DISCUSSION

Through our extensive field testing and evaluation of this algorithm, we have learned a number of lessons that apply generally to the field of camera-based localization and mapping. First and foremost, we have shown that long-range autonomous navigation in unstructured, 3D terrain is possible using a stereo camera as the only sensor and using the SURF algorithm to detect and describe visual landmarks. Recent work has shown that it is possible to perform more accurate mapping using sparse bundle adjustment (Konolige et al., 2007; Sibley et al., 2009) and optimization over large-scale loops in the trajectory (Konolige & Agrawal, 2008; Newman, Sibley, Smith, Cummins, Harrison, et al., 2009). However, mapping accuracy did not limit the performance of our algorithm, and we feel that these advances, while desirable, are not necessary to build a robust long-range navigation system. The rest of this section will outline the major lessons learned throughout this project.

The limitations of the feature detection and description pipeline: The SURF algorithm had a lot of trouble dealing with lighting changes. This was particularly evident on terrain with 3D structure (the rocks in Figure 20 or on grass) and less of an issue in urban environments (on concrete and near buildings). Although the SURF descriptor is normalized to provide some invariance to the effects of lighting, the detector will return different sets of points when shadowing produces areas of high contrast in the image. It is possible that performance could be improved somewhat by preprocessing the images [e.g., the patch normalization in Zhang & Kleeman (2009)], but strong shadows on 3D terrain would continue to cause problems. We have shown this for the SURF algorithm, but the results hold for any image-space blob detector. To deal with lighting in the current framework, it would be possible to learn a route several times under different lighting conditions and then dynamically select the "best" map sequence for route repeating (based on matching score or time of day). However, this does not address the main problem that the current feature detection and description paradigm does not deal well with lighting changes.

The utility of dead reckoning: Interleaving VO and localization was one of the keys to making this algorithm work in practice. This is clearly demonstrated in the multimedia extensions (Appendix A). VO carries the algorithm through areas with moderate appearance changes, and localization keeps the estimate consistent over long distances and corrects for VO failures. Although we found VO to be very effective, some form of dead reckoning *not* based on the camera could be very useful. A planetary exploration rover with power and computational constraints could use wheel odometry between stereo images in low-slip environments. The combination of local submaps and wheel odometry was already used by Marshall et al. (2008) for navigation in underground mines. For terrestrial applications, we advocate the use of an inertial measurement unit (IMU). As described by Corke, Lobo, and Dias (2007), cameras and IMUs are complementary sensors. The use of an IMU in this work would have eliminated teach pass failures and compensated for VO failures due to motion blur. Our next iteration will incorporate an IMU directly into the mapping and localization framework.

The importance of map update: The performance of our system degrades as the environment surrounding a route changes over time. Solutions to this problem—sometimes called persistent mapping (Milford & Wyeth, 2009) or life-long learning (Konolige & Bowman, 2009)—must be developed before robots can be broadly deployed in service roles. Although our system could be patched to remap while the path is being followed, this would not address the underlying structure of the problem, which includes difficult issues such as (i) differentiating static and dynamic scene elements, (ii) periodic environmental changes (e.g., daily lighting changes or seasonal changes), or (iii) joining disparate maps in the event of a loop closure. We envision the next iteration of our system becoming much like the one described by Konolige and Bowman (2009), mapping and localizing continuously online while retaining the ability to retrace a known path at any time.

The utility of loop detection: Although it was not the focus of this work, the ability to handle loops and networks of paths would increase the number of possible applications of the algorithm. Loop detection in visual SLAM is an active research area [see the recent review by Williams, Cummins, Neira, Newman, Reid, et al. (2009)], and the incorporation of a fast, accurate loop-detection technique [such as FAB-MAP (Cummins & Newman, 2008)] along with further geometric consistency checking (Eade & Drummond, 2008) would provide two immediate benefits. First, the signal from a dedicated loop detection algorithm could totally replace the submap selection component of our system. Reliable automatic submap selection would make our algorithm more robust to path-tracking errors or VO failures. Second, this would allow the system to build a graph of connected route segments. The graph representation could be used to plan routes between places on the map. Within the current framework, it should be possible to stack submaps at intersections (one submap per branch). However, navigating on a graph of routes would be more elegantly handled using a continuous relative representation (Mei et al., 2009; Sibley et al., 2009).

7. CONCLUSION

We have presented a complete algorithm for performing long-range rover navigation using a stereo camera as the only sensor. Our system produces a combined topological/metric map consisting of a sequence of small overlapping submaps. As the rover progresses along a path, the nearest submap is swapped into memory. The rover interleaves VO and localization, using VO to carry the algorithm through areas with moderate appearance changes and using localization to ensure that the rover ends up in the same physical place at the end of a long path. We have tested our algorithm in an urban setting, over extreme terrain, through indoor-to-outdoor lighting changes, and in a planetary analog setting in the High Arctic that offered many types of vegetation-free terrain. Of the 32.919 km traveled, 99.6% was traversed autonomously, and in all situations requiring an intervention, the rover stopped and signaled the operator.

8. APPENDIX A: INDEX OF MULTIMEDIA EXTENSIONS

Two multimedia extensions have been prepared to accompany this work. The extensions show different views of the algorithm performing on challenging routes. The videos are available as Supporting Information files in the online version of this article or at http://asrl.utias.utoronto.ca/~ptf/JFR_VTnR.

Extension	Media type	Description
1	Video	A video of route uh-05-22-0120, repeat pass 6. The video shows an external view, the robot's view during the repeat pass, the nearest teach pass image, inlier feature counts, two-dimensional localization, and 3D localization.
2	Video	A video of route dh-07-31-0192, repeat pass 2. The video shows an external view, the robot's view during the repeat pass, the nearest teach pass image, inlier feature counts, two-dimensional localization, and 3D localization.

Table BI. Teach passes.

Tag	Length (m)	Maps	Features per map	Failures
ua-05-17-0726	725.6	253	2,937	0
uh-05-20-1152	1,151.7	405	3,484	0
uh-05-21-1170	1,169.7	410	3,424	0
uh-05-22-0120	120.0	42	5,218	0
uh-07-23-0120	119.7	43	5,323	0
uh-05-26-0202	201.8	71	3,714	0
ua-06-04-0086	85.6	30	1,924	0
ua-06-04-0048	47.7	17	1,857	0
ua-06-04-0097	96.8	35	1,581	3
ua-06-04-0091	91.1	31	1,924	0
ua-06-06-0081	81.5	28	1,718	0
ua-06-06-0184	184.4	65	1,563	0
ua-06-06-0186	185.8	65	1,975	1
ua-06-06-0167	166.9	60	1,540	0
dh-07-20-2120	2,120.0	740	3,680	1
da-07-20-0464	464.0	166	1,662	1
da-07-21-0453	453.5	161	1,960	1
dh-07-22-1091	1,090.9	382	5,110	0
dh-07-23-4963	4,962.5	1,732	3,993	5
da-07-23-0557	557.3	200	2,126	0
da-07-24-0425	424.6	151	2,225	0
da-07-29-0487	486.9	172	2,318	0
da-07-29-0486	486.2	176	2,415	0
dh-07-30-0347	347.3	124	3,147	0
dh-07-30-0187	187.5	66	3,322	0
dh-07-30-0153	152.8	54	3,145	0
dh-07-31-0192	191.8	68	5,431	0

Table BII. Teach passes with difficulty metrics.

Tag	Elevation change	Min roll	Max roll	Min pitch	Max pitch
ua-05-17-0726	7.1	−7.8	12.2	−6.4	5.6
uh-05-20-1152	4.9	−18.4	8.2	−12.0	6.3
uh-05-21-1170	4.9	−17.8	11.9	−14.6	10.3
uh-05-22-0120	3.6	−14.7	11.7	−21.5	27.2
uh-07-23-0120	4.9	−13.3	12.1	−18.5	26.5
uh-05-26-0202	3.7	−5.5	3.5	−8.4	2.4
ua-06-04-0086	1.6	−8.0	2.9	−5.1	1.6
ua-06-04-0048	1.0	−6.7	5.5	−5.1	6.1
ua-06-04-0097	1.3	−6.6	3.3	−7.3	1.9
ua-06-04-0091	1.1	−5.8	3.1	−5.7	4.3
ua-06-06-0081	0.9	−9.4	4.0	−5.9	2.3
ua-06-06-0184	1.3	−7.9	3.7	−5.7	2.3
ua-06-06-0186	1.5	−10.4	4.9	−6.8	7.4
ua-06-06-0167	3.1	−8.8	3.1	−4.0	2.4
dh-07-20-2120	69.2	−22.0	15.9	−28.3	16.9
da-07-20-0464	8.7	−12.0	4.5	−9.7	6.9
da-07-21-0453	9.4	−13.0	4.3	−10.0	11.6
dh-07-22-1091	32.4	−7.4	10.0	−11.0	11.2
dh-07-23-4963	118.5	−12.8	13.6	−15.5	12.0
da-07-23-0557	36.7	−8.7	12.0	−11.7	13.4
da-07-24-0425	7.4	−6.7	5.9	−7.4	18.3
da-07-29-0487	9.1	−9.0	4.1	−13.0	7.5
da-07-29-0486	9.1	−6.6	3.4	−11.0	6.8
dh-07-30-0347	2.1	−12.2	13.1	−12.1	10.8
dh-07-30-0187	1.5	−6.1	8.5	−12.3	7.7
dh-07-30-0153	1.5	−6.9	5.9	−10.6	3.6
dh-07-31-0192	1.1	−9.8	10.9	−17.9	19.1

9. APPENDIX B: CATALOG OF ROUTE-FOLLOWING EXPERIMENTS

This appendix includes the complete listing of our route-following experiments. Table BI lists each teach pass with some basic information such as the length of the route, the number of features per map, and the number of failures encountered teaching the route. Table BII lists some difficulty metrics for each teach pass: the elevation change reported by GPS and the roll and pitch variation measured by an inclinometer mounted on the sensor head. The repeat passes are listed in Tables BIII and BIV. For each repeat pass we report the start time of both the teach-and-repeat passes (relevant for lighting effects), the percentage of the route completed, the percentage of run completed autonomously, the number of operator interventions, and the percentage of time the algorithm was globally localized.

ACKNOWLEDGMENTS

This paper would not have been possible without the support of many people, and the authors would like to thank some of them here. Braden Stenning did an excellent job developing most of the middleware for our robot. Colin McManus implemented the path-tracking module. Chi Hay Tong and Gaetan Kenway were of great help developing the GPU-SURF pipeline for a course run by Andreas Moshovos. Thanks to Pascal Lee and the entire Mars Institute/Haughton Mars Project team for support in the field. Funding for our field trials on Devon Island was provided by the Canadian Space Agency's Canadian Analogue Research Network (CARN) program, and the Natural Sciences and Engineering Research Council of Canada (NSERC) funded the remaining work. Finally, the authors would like to thank the

Table BIII. Repeat passes (1/2).

Teach pass tag	Teach pass start time	Repeat pass start time	Completed (%)	Autonomous (%)	Operator interventions	Globally localized (%)
ua-05-17-0726	12:27:12	14:27:02	100.0	100.0	0	41.1
		12:50:02	100.0	100.0	0	91.8
uh-05-20-1152	12:04:45	09:25:48	100.0	100.0	0	88.8
uh-05-21-1170	12:16:02	20:26:28	100.0	92.0	1	49.6
		12:54:14	100.0	100.0	0	99.2
		15:28:08	100.0	100.0	0	97.4
		16:14:01	100.0	100.0	0	99.1
		08:03:44	100.0	98.4	3	45.0
		09:04:15	100.0	100.0	0	76.3
uh-05-22-0120	18:13:22	10:23:40	100.0	100.0	0	96.8
		10:56:05	100.0	100.0	0	94.1
		11:18:14	100.0	100.0	0	96.0
		11:41:32	100.0	100.0	0	91.7
		11:59:30	100.0	100.0	0	95.8
		13:04:15	100.0	100.0	0	89.1
		11:17:39	100.0	100.0	0	90.8
uh-07-23-0120	16:53:43	18:39:23	100.0	100.0	0	93.6
		19:00:53	100.0	100.0	0	97.3
		19:12:07	100.0	100.0	0	95.8
		19:22:49	100.0	100.0	0	97.8
		19:33:36	100.0	100.0	0	95.5
uh-05-26-0202	11:14:39	07:44:53	100.0	100.0	0	95.9
		08:53:08	100.0	100.0	0	95.5
		09:41:09	100.0	100.0	0	88.7
		10:41:07	100.0	100.0	0	87.5
		11:39:57	100.0	100.0	0	95.0
		12:38:06	100.0	100.0	0	94.7
		13:34:20	100.0	100.0	0	98.8
		14:38:29	100.0	100.0	0	86.0
		14:58:39	100.0	100.0	0	88.8
ua-06-04-0086	12:57:06	13:12:38	100.0	100.0	0	99.4
ua-06-04-0048	14:41:09	14:51:15	100.0	100.0	0	97.2
ua-06-04-0097	14:48:50	15:10:59	87.4	95.5	2	93.4

Table BIV. Repeat passes (2/2).

Teach pass tag	Teach pass start time	Repeat pass start time	Completed (%)	Autonomous (%)	Operator interventions	Globally localized (%)
ua-06-04-0091	16:26:46	16:37:13	100.0	100.0	0	84.5
ua-06-06-0081	10:01:59	10:08:55	100.0	100.0	0	100.0
ua-06-06-0184	10:25:07	11:35:48	100.0	100.0	0	99.7
ua-06-06-0186	13:11:57	13:55:08	100.0	98.4	1	81.5
ua-06-06-0167	21:09:38	21:45:16	100.0	100.0	0	100.0
dh-07-20-2120	10:18:18	09:45:12	100.0	100.0	0	97.4
		11:05:46	100.0	100.0	0	89.9
da-07-20-0464	16:09:42	16:46:56	100.0	100.0	0	94.4
da-07-21-0453	17:26:10	18:07:05	100.0	100.0	0	98.4
dh-07-22-1091	08:55:08	10:35:23	100.0	100.0	0	99.5
		11:14:58	100.0	100.0	0	100.0
		12:00:56	100.0	100.0	0	99.2
		13:40:39	100.0	100.0	0	99.9
		14:13:58	100.0	100.0	0	97.5
dh-07-23-4963	08:50:49	08:49:24	64.9	100.0	0	96.3
		15:07:49	32.8	100.0	0	76.2
da-07-23-0557	17:19:32	18:21:22	100.0	100.0	0	99.2
da-07-24-0425	16:24:23	17:20:45	100.0	100.0	0	98.4
da-07-29-0487	11:41:31	15:28:45	100.0	100.0	0	96.7
da-07-29-0486	16:04:04	15:28:45	100.0	100.0	0	96.7
		17:00:29	100.0	100.0	0	96.0
dh-07-30-0347	12:03:40	12:49:47	100.0	100.0	0	89.5
dh-07-30-0187	14:35:09	18:37:18	78.4	100.0	0	9.9
		10:02:35	100.0	100.0	0	75.7
dh-07-30-0153	16:33:18	16:59:04	100.0	100.0	0	93.9
dh-07-31-0192	11:40:01	12:27:23	100.0	100.0	0	100.0
		12:50:01	100.0	100.0	0	92.7

reviewers for their insightful comments and constructive criticism.

REFERENCES

Argyros, A. A., Bekris, K. E., Orphanoudakis, S. C., & Kavraki, L. E. (2005). Robot homing by exploiting panoramic vision. Autonomous Robots, 19(1), 7–25.

Baumgartner, E. T., & Skaar, S. B. (1994). An autonomous vision-based mobile robot. IEEE Transactions on Automatic Control, 39(3), 493–502.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). Computer Vision Image Understanding, 110(3), 346–359.

Bekris, K. E., Argyros, A. A., & Kavraki, L. E. (2006). Exploiting panoramic vision for bearing-only robot homing. In K. Daniilidis & R. Kleete (Eds.), Imaging beyond the pinhole camera (pp. 229–251). Lecture Notes in Computer Science. Dordrecht, The Netherlands: Springer.

Blanc, G., Mezouar, Y., & Martinet, P. (2005, April). Indoor navigation of a wheeled mobile robot along visual routes. In ICRA 2005. Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005, Barcelona, Spain (pp. 3354–3359).

Bonin-Font, F., Ortiz, A., & Oliver, G. (2008). Visual navigation for mobile robots: A survey. Journal of Intelligent and Robotic Systems, 53(3), 263–296.

Booij, O., Terwijn, B., Zivkovic, Z., & Krose, B. (2007, April). Navigation using an appearance based topological map. In 2007 IEEE International Conference on Robotics and Automation, Rome (pp. 3927–3932).

Bosse, M., Newman, P., Leonard, J., & Teller, S. (2004). Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. International Journal of Robotics Research, 23(12), 1113–1139.

Brooks, R. (1985, March). Visual map making for a mobile robot. In 1985 IEEE International Conference on Robotics and Automation Proceedings, Durham, England (vol. 2, pp. 824–829).

Chen, Z., & Birchfield, S. (2006, May). Qualitative vision-based mobile robot navigation. In ICRA 2006. Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006, Orlando, FL (pp. 2686–2692).

Corke, P., Lobo, J., & Dias, J. (2007). An introduction to inertial and visual sensing. International Journal of Robotics Research, 26(6), 519–535.

Courbon, J., Mezouar, Y., & Martinet, P. (2008). Indoor navigation of a non-holonomic mobile robot using a visual memory. Autonomous Robots, 25(3), 253–266.

Cummins, M., & Newman, P. (2008). FAB-MAP: Probabilistic localization and mapping in the space of appearance. International Journal of Robotics Research, 27(6), 647–665.

Demirdjian, D., & Darrell, T. (2002). Using multiple-hypothesis disparity maps and image velocity for 3-D motion estimation. International Journal of Computer Vision, 47(1–3), 219–228.

Diosi, A., Remazeilles, A., Segvic, S., & Chaumette, F. (2007, November). Outdoor visual path following experiments. In IROS 2007. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, San Diego, CA (pp. 4265–4270).

Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: Part I. IEEE Robotics & Automation Magazine, 13(2), 99–110.

Eade, E., & Drummond, T. (2008, September). Unified loop closing and recovery for real time monocular SLAM. In 19th British Conference on Machine Vision, Leeds, UK.

Fong, T., Allan, M., Bouyssounouse, X., Bualat, M., Deans, M., Edwards, L., Fluckiger, L., Keely, L., Lee, S., Lees, D., To, V., & Utz, H. (2008, February). Robotics site survey at Haughton Crater. In Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS), Los Angeles, CA.

Fong, T., Deans, M., Lee, P., & Bualat, M. (2007, March). Simulated lunar robotic survey at terrestrial analog sites. In Proceedings of the 38th Lunar and Planetary Science Conference, League City, TX.

Goedemé, T., Nuttin, M., Tuytelaars, T., & Van Gool, L. (2007). Omnidirectional vision based topological navigation. International Journal of Computer Vision, 74(3), 219–236.

Goedemé, T., Tuytelaars, T., & Van Gool, L. (2005, October). Omnidirectional sparse visual path following with occlusion-robust feature tracking. In 6th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras, OMNIVIS05, in Conjunction with ICCV 2005, Beijing, China.

Goedeme, T., Tuytelaars, T., Van Gool, L., Vanacker, G., & Nuttin, M. (2005, August). Feature based omnidirectional sparse visual path following. In 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005 (IROS 2005), Edmondton, Canada (pp. 1806–1811).

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A, 4(4), 629–642.

Howard, A., Sukhatme, G. S., & Mataric, M. J. (2006). Multirobot simultaneous localization and mapping using manifold representations. Proceedings of the IEEE, 94(7), 1360–1369.

Jones, S., Andresen, C., & Crowley, J. (1997, September). Appearance based process for visual navigation. In IROS '97, Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1997, Grenoble, France (vol. 2, pp. 551–557).

Jun, K. K., Miura, J., & Shirai, Y. (2000). Autonomous visual navigation of a mobile robot using a human-guided experience. Robotics and Autonomous Systems, 40, 2–3.

Kaess, M., Ni, K., & Dellaert, F. (2009, May). Flow separation for fast and robust stereo odometry. In IEEE International Conference on Robotics and Automation, ICRA, Kobe, Japan.

Konolige, K., & Agrawal, M. (2008). FrameSLAM: From bundle adjustment to real-time visual mapping. IEEE Transactions on Robotics, 24(5), 1066–1077.

Konolige, K., Agrawal, M., Blas, M. R., Bolles, R. C., Gerkey, B., Solà, J., & Sundaresan, A. (2009). Mapping, navigation, and learning for off-road traversal. Journal of Field Robotics, 26(1), 88–113.

Konolige, K., Agrawal, M., & Solà, J. (2007, November). Large scale visual odometry for rough terrain. In Proceedings of the International Symposium on Research in Robotics (ISRR), Hiroshima, Japan.

Konolige, K., & Bowman, J. (2009, October). Towards lifelong visual maps. In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009, St. Louis, MO.

Lee, P., Braham, S., Boucher, M., Schutt, J., Glass, B., Gross, A., Hine, B., McKay, C., Hoffman, S., Jones, J., Berinstain, A., Comptois, J.-M., Hodgson, E., & Wilkinson, N. (2007, March). Haughton-Mars project: 10 years of science operations and exploration systems development at a moon/Mars analog site on Devon Island, High Arctic. In Proceedings of the 38th Lunar and Planetary Science Conference, League City, TX (pp. 2426–2427).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91–110.

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.

Maimone, M., Johnson, A., Cheng, Y., Willson, R., & Matthies, L. (2006). Autonomous navigation results from the Mars Exploration Rover (MER) mission. In Experimental robotics IX, vol. 21 of Springer Tracts in Advanced Robotics (pp. 3–13). Berlin: Springer.

Marshall, J., Barfoot, T., & Larsson, J. (2008). Autonomous underground tramming for center-articulated vehicles. Journal of Field Robotics, 25(6–7), 400–421.

Matsumoto, Y., Inaba, M., & Inoue, H. (1996, April). Visual navigation using view-sequenced route representation. In 1996 IEEE International Conference on Robotics and Automation, Minneapolis, MN (vol. 1, pp. 83–88).

Matthies, L. (1989). Dynamic stereo vision. Ph.D. thesis, Carnegie Mellon University Computer Science Department.

Mei, C., Sibley, G., Cummins, M., Newman, P., & Reid, I. (2009, September). A constant time efficient stereo SLAM system. In British Machine Vision Conference, London, England.

Milford, M., & Wyeth, G. (2009). Persistent navigation and mapping using a biologically inspired SLAM system. International Journal of Robotics Research, in press.

Moravec, H. (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover. Ph.D. thesis, Stanford University. Available as Stanford AIM-340, CS-80-813, and republished as a Carnegie Mellon University Robotics Institute Tech. Rep. CMU-RI-TR-80-03.

Newman, P., Leonard, J., Tardos, J. D., & Neira, J. (2002, May) Explore and return: Experimental validation of real-time concurrent mapping and localization. In Proceedings ICRA '02. IEEE International Conference on Robotics and Automation, 2002, Washington, DC (vol. 2, pp. 1802–1809).

Newman, P., Sibley, G., Smith, M., Cummins, M., Harrison, A., Mei, C., Posner, I., Shade, R., Schroeter, D., Murphy, L., Churchill, W., Cole, D., & Reid, I. (2009). Navigating, recognizing and describing urban spaces with vision and lasers. International Journal of Robotics Research, 28(11–12), 1406–1433.

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. (2004, June). Visual odometry. In CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, Washington, DC (vol. 1, pp. 652–659).

Ohno, T., Ohya, A., & Yuta, S. (1996, November). Autonomous navigation for mobile robots referring prerecorded image sequence. In IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems '96, Osaka, Japan (vol. 2, pp. 672–679).

Payá, L., Reinoso, O., Gil, A., Pedrero, J., & Ballesta, M. (2007). Appearance-based multirobot following routes using incremental PCA. Knowledge-Based Intelligent Information and Engineering Systems (pp. 1170–1178). Berlin: Springer.

Royer, E., Lhuillier, M., Dhome, M., & Lavest, J.-M. (2007). Monocular vision for mobile robot localization and autonomous navigation. International Journal of Computer Vision, 74(3), 237–260.

Šegvić, S., Remazeilles, A., Diosi, A., & Chaumette, F. (2009). A mapping and localization framework for scalable appearance-based navigation. Computer Vision Image Understanding, 113(2), 172–187.

Sibley, G., Matthies, L., & Sukhatme, G. (2008). A sliding window filter for incremental SLAM. In Unifying perspectives in computational and robot vision, vol. 8 of Lecture Notes in Electrical Engineering (pp. 103–112). New York: Springer US.

Sibley, G., Mei, C., Reid, I., & Newman, P. (2009, June). Adaptive relative bundle adjustment. In Robotics Science and Systems (RSS), Seattle, WA.

Tang, L., & Yuta, S. (2001, May). Vision based navigation for mobile robots in indoor environment by teaching and playing-back scheme. In IEEE International Conference on Robotics and Automation, 2001. Proceedings 2001 ICRA, Seoul, Korea (vol. 3, pp. 3072–3077).

Wettergreen, D., Dias, M., Shamah, B., Teza, J., Tompkins, P., Urmson, C., Wagner, M., & Whittaker, W. (2002, May). First experiment in sun-synchronous exploration. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Washington, DC (pp. 3501–3507).

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.

Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I., & Tardós, J. (2009). A comparison of loop closing techniques in monocular SLAM. Robotics and Autonomous Systems, 57(12), 1188–1197.

Zhang, A. M., & Kleeman, L. (2009). Robust appearance based visual route following for navigation in large-scale outdoor environments. International Journal of Robotics Research, 28(3), 331–356.