2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Stereo Mapping and Localization for Long-Range Path Following on Rough Terrain

Paul Furgale* and Tim Barfoot†

University of Toronto Institute for Aerospace Studies
Toronto, Ontario, Canada, M3H 5T6
*paul.furgale@utoronto.ca, †tim.barfoot@utoronto.ca

***Abstract*— Visual teach-and-repeat navigation enables long-range rover autonomy without solving the simultaneous localization and mapping problem or requiring an accurate global reconstruction. During a learning phase, the rover is piloted along a route, logging images. After post-processing, the rover is able to repeat the route in either direction any number of times. This paper describes and evaluates the localization algorithm at the core of a teach-and-repeat system that has been tested on over 32 kilometers of autonomous driving in an urban environment and at a planetary analog site in the High Arctic. We show how a stereo visual odometry pipeline can be extended to become a mapping and localization system, then evaluate the performance of the algorithm with respect to accuracy, robustness to path-tracking error, and the effects of lighting.**

I. Introduction

In environments lacking a Global Positioning System (GPS) or equivalent, long-range autonomous navigation for rovers becomes a very difficult problem. Relative localization 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.

We have developed 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 series of overlapping submaps. These submaps are then used for localization during the autonomous traversal phase—the *repeat pass*. Other teach-and-repeat systems have been proposed for rovers navigating indoors [1][2], in mines [3], or outdoors in planar environments [4][5][6], but this is the first system shown to work over multi-kilometer autonomous traverses, and in highly three-dimensional, outdoor, unstructured environments.

We have tested our teach-and-repeat system using the rover shown in Figure 1 on more than 32 kilometers of autonomous route following. This paper will focus on the localization algorithm used in our teach-and-repeat system. Related work will be discussed in Section II, the localization system will be described in Section III, and Section IV details our evaluation experiments.

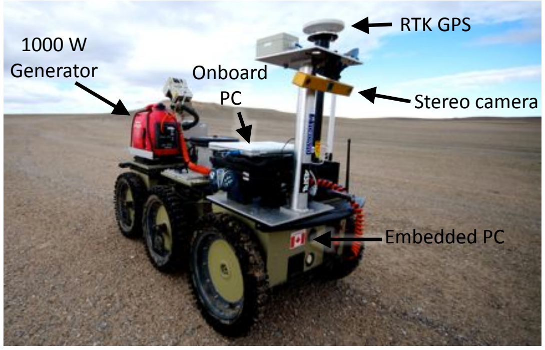

Fig. 1. The six-wheeled rover platform used in our visual path following experiments.

II. Related Works

Developing a teach-and-repeat system for outdoor, unstructured environments requires the handling of arbitrary camera motions and so localization within three-dimensional space is necessary. Many previous teach-and-repeat systems have relied on the constraint that the camera was moving in a plane. These systems have been built on planar laser scanners [2][3], omnidirectional cameras [7][1][8][6], and monocular cameras [9][4][5]. The camera-based systems can be loosely grouped into *appearance-based* and *feature-based* systems. Appearance-based systems compare large portions of the input image with prototype images captured during the teach pass [9][6]. Planar camera motion is necessary to ensure that the image templates will line up for correlation. The feature-based algorithms (like ours) match point features between the input image and prototypes captured during route learning. Using point features for localization can remove the planarity constraint, but many systems assume planarity and use point features for bearing-only navigation [7][8]. A handful of systems perform three-dimensional localization based on point-feature correspondences but have never been tested under significant three-dimensional camera motion [4][5].

We show that it is possible to use stereo vision alone to retrace a long route with arbitrary camera motion in an outdoor, unstructured environment. Our work extends the basic stereo odometry pipeline [10][11]: tracking stereo

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

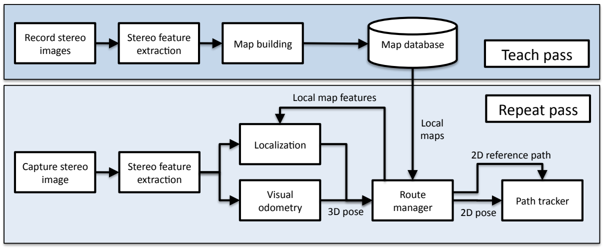

Fig. 2. An overview of the major processing blocks in our system.

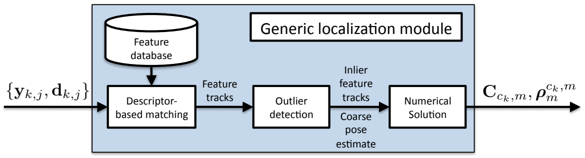
Feature database
Generic localization module
$\{\mathbf{y}_{k,j}, \mathbf{d}_{k,j}\}$
Feature tracks
Descriptor-based matching
Outlier detection
Inlier feature tracks
Coarse pose estimate
Numerical Solution
$\mathbf{C}_{c_k,m}, \boldsymbol{\rho}_m^{c_k,m}$

Fig. 3. An overview of our generic localization module.

features, rejecting 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 used to drive multi-kilometer autonomous routes in a single command cycle.

III. System Description

This section will present a detailed description of the localization algorithm used in our teach-and-repeat system. The major processing blocks of our system are depicted in Figure 2. Throughout this work, we use a generic localization module based on stereo visual odometry. The outline is shown in Figure 3.

The map frame, $\underrightarrow{\mathcal{F}}_m$, is the frame in which three-dimensional estimation occurs. We define $\underrightarrow{\mathcal{F}}_{r_k}$ 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 $\mathbf{C}_{m,r_k}$, the rotation matrix that transforms vectors from $\underrightarrow{\mathcal{F}}_{r_k}$ to $\underrightarrow{\mathcal{F}}_m$. Similarly, we define the camera's position as $\rho_m^{r_k,m}$, a vector from the origin of $\underrightarrow{\mathcal{F}}_m$ to the origin of $\underrightarrow{\mathcal{F}}_{r_k}$ (denoted by the superscript), and expressed in $\underrightarrow{\mathcal{F}}_m$ (denoted by the subscript). Together, $\mathbf{C}_{m,r_k}$ and $\rho_m^{r_k,m}$ define the camera's *pose* in $\underrightarrow{\mathcal{F}}_m$.

We use our own implementation of the Speeded Up Robust Features (SURF) algorithm [12] for feature detection and description. At time k, running the SURF algorithm on both images of a stereo pair and making associations between the pair results in a set of stereo keypoints. 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, $\mathbf{d}_{k,j}$. Our localization system requires an observation model, $\mathbf{h}(\cdot)$, a function that projects points expressed in the left camera frame, $\mathbf{p}_{r_k}^{j,r_k}$, into the image coordinates:

$$\mathbf{y}_{k,j} = \mathbf{h}\left(\mathbf{p}_{r_k}^{j,r_k}\right) \tag{1}$$

Because we are using a calibrated stereo camera, (1) is invertible. The inverse observation model, $\mathbf{g}(\cdot)$, triangulates points seen in a stereo pair:

$$\mathbf{p}_{r_k}^{j,r_k} = \mathbf{g}\left(\mathbf{y}_{k,j}\right) \tag{2}$$

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*, *initialization*, *relative localization*, and *global 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 $\begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T$ position of feature i
$\mathbf{v}_i$	:	The SURF descriptor associated with feature i

Descriptor-based tracking is done by looking for nearest neighbors in descriptor space. The output of the first block in Figure 3 is a list of candidate feature tracks, each mapping 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 Random Sample Consensus (RANSAC) [13], as it will on average produce the best set of inliers given a fixed computational budget. Treating the feature database and the incoming stereo keypoints as three-dimensional point clouds, we use the three-point quaternion method of [14] 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 image coordinates of each incoming keypoint, the feature database, the pose estimate supplied by RANSAC, and the camera's pose from the last timestep. Using this data it produces an estimate of the camera's pose in $\underrightarrow{\mathcal{F}}_m$. Each solution method is iterative, based on Gauss-Newton minimization, but each operating mode uses a different mathematical formulation.

A. Route Teaching

The basic process for route teaching involves driving the path once while logging stereo images, and then post-processing the image sequence into a series of overlapping submaps. A localization loop incrementally builds the map and estimates the position of the rover within it. Periodically, the map is split, and the raw data is further processed into the format used in the repeat pass.

Submaps are constructed using a specialization of the generic localization module. 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 $\underrightarrow{\mathcal{F}}_{r_0}$. All of the keypoints are triangulated using (2) 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 provides a mapping from feature i in the map, to keypoint j. To estimate $\mathbf{C}_{r_k,m}$, and $\rho_m^{r_k,m}$, we define the error term, $\mathbf{e}_n$:

$$\mathbf{e}_n := \mathbf{y}_{k,j} - \mathbf{h}(\mathbf{C}_{r_k,m}(\mathbf{q}_m^{i,m} - \rho_m^{r_k,m}))$$

Letting M_k be the number of feature tracks at time k, we define our objective function, J_{vis}, to be

$$J_{\text{vis}} := \frac{1}{2}\sum_{n=1}^{M_k} \mathbf{e}_n^T \mathbf{W}_n \mathbf{e}_n \ , \tag{3}$$

where $\mathbf{W}_n$ is a weighting matrix based on the inverse of the estimated measurement covariance of $\mathbf{y}_{k,j}$. We linearize (3) and minimize J_{vis} using the Gauss-Newton method.

When the percentage of features tracked drops below a threshold, τ_f, the pose, $\left\{\mathbf{C}_{r_k,m}, \rho_m^{r_k,m}\right\}$, is added to the 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 previous step, triangulated keypoints are placed into the map in a common frame, $\underrightarrow{\mathcal{F}}_m$:

$$\begin{aligned} \mathbf{q}_m^{i,m} &= \mathbf{C}_{r_k,m}^T(\mathbf{g}(\mathbf{y}_{j,k})) + \rho_m^{r_k,m} \\ &= \mathbf{C}_{r_k,m}^T \mathbf{p}_{r_k}^{j,r_k} + \rho_m^{r_k,m} \end{aligned}$$

Incoming keypoints that are not successfully tracked are added to the map as seen. Keypoints that are successfully tracked are discarded; the prototype feature in our system is based on the triangulated position and SURF descriptor of the first view only. Although there is enough information here to estimate the camera's pose *and* the feature positions—either on the entire map [4], or on some sliding window of poses [11][15]—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. While future work may involve some evaluation of the benefits of better reconstruction techniques, feature position estimation 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 map is packaged for the repeat pass and saved to disk. By changing this parameter, our system scales smoothly between a global map [4] and a view-sequenced route representation that matches against single images along the path [5]. Increasing the submap size increases the difficulty and computational complexity of the descriptor-based matching task, whereas reducing the map size decreases the likeliness of localizing globally during a path-tracking error. For all of the experiments in this paper, we have used $\tau_l = 5$ meters, which we feel is a good tradeoff between these two concerns. Packaged maps include a vehicle reference path with L poses (indexed by ℓ), $\{\rho_m^{\ell,m}\}$, and a set of N features (indexed by i), $\{\mathbf{q}_m^{i,m}, \mathbf{v}_i\}$. This satisfies the requirements needed to be used as a feature database in the generic localization module.

After saving the map to disk, older poses and features are removed from the database in memory. We build the submaps to overlap by 50% as this ensures overlapping data when transitioning between maps [3]. 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 localization 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. When the teach pass is complete, a database of maps is available for use in the repeat pass.

B. 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. Neither direction switching during path following nor local obstacle detection have been implemented, although both should be possible [3]. This section will describe the route following localization algorithm in detail.

Three specializations of the generic localization module are used during the repeat pass: *initialization*, *global localization*, and *relative localization*.

Initialization is performed at the start of a route or when the robot is lost. One of the maps built in the teaching 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 (9 for all experiments in this paper), the objective function used in the route-teaching phase (3) is used to solve for the pose of the camera. If the initialization is successful, the rover begins to repeat the route, interleaving relative and global localization as the route is retraced.

The interleaving of global and relative localization 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 [4] or [5]. In both of these projects, all localization is global. This worked well on pavement, and in urban environments, but when we tested our system in grass and rough terrain, the system failed too easily under changing lighting conditions. Realizing that our localization module was based on visual odometry (a purely relative localization method), we implemented a system that would switch back and forth between relative and global localization. Visual odometry is accurate enough to keep the rover near the path through difficult areas, and periodic global localizations maintain the global (topological) accuracy that allows long routes to be repeated. This is similar to the method used by [6] who use wheel odometry in between their global corrections. We perform relative localization every frame, but given our current hardware, there are not enough computational resources to also perform global localization every frame.

Hence, we introduce an integer parameter G and only attempt global localization when $\mathrm{mod}(k,G)=0$ (every Gth frame). In these experiments we have used $G=4$.

For relative localization, we have implemented frame-to-frame visual odometry. Keypoints from the previous pose are triangulated and transformed into $\underrightarrow{\mathcal{F}}_m$ using $\mathbf{C}_{r_k,m}$ and $\rho_m^{r_k,m}$, then used as the feature database. After outlier rejection, the pose of the rover is estimated using the Gauss-Newton method described by [16].

Global localization is handled similarly to the initialization phase. Using only the localization method described for initialization, 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 [17]. To account for this, a prior information term is added to the error term used to estimate the pose. Starting with J_{vis} from (3), we add prior information error terms for the position, J_{pos}, and attitude, J_{att}, so that the error term we minimize, J, is

$$J = J_{\text{vis}} + J_{\text{pos}} + J_{\text{att}} \quad . \tag{4}$$

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

$$J_{\text{pos}} := \frac{1}{2}\left(\hat{\rho}_m^{r_k,m} - \rho_m^{r_k,m}\right)^T \mathbf{W}_{\text{pos}} \left(\hat{\rho}_m^{r_k,m} - \rho_m^{r_k,m}\right) \quad .$$

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

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

The weighting matrices were chosen to be

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

where $\mathbf{1}$ is the identity matrix. All experiments in this work use $\sigma_{\text{pos}}^2 = 0.1$ and $\sigma_{\text{att}}^2 = 1.0$. As with the previous localization methods, (4) is linearized and solved using the Gauss-Newton method.

IV. System Evaluation

We have tested our teach-and-repeat system on 27 learned routes ranging in length from 47 meters to nearly 5 kilometers. Field trails were conducted in the urban environment surrounding the University of Toronto Institute for Aerospace Studies (UTIAS) and in a planetary analog setting near the Haughton-Mars Project on Devon Island, Nunavut. GPS tracks of our large-scale routes on Devon Island are shown in Figure 4. The 27 routes were used to perform 60 autonomous runs. During the most extreme routes, the rover experienced up to 118.5 meters of elevation change, as well as pitch and roll deviation from vertical of up to 28° and 22°, respectively. The longest fully autonomous run was 3.2 kilometers. There were two autonomous runs of approximately two kilometers and ten autonomous runs approximately one kilometer long. Out of the 32.919 kilometers traveled, only 0.128 kilometers were piloted manually. This is an autonomy rate of 99.6%. In all cases where the rover needed an intervention, it stopped along the path and signaled the operator.

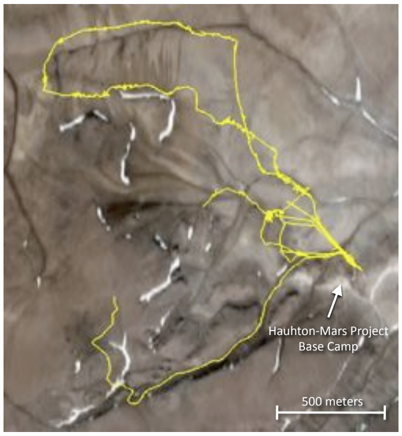

Fig. 4. GPS tracks of our path-following experiments on Devon Island.

A. Localization Performance During Path Following

To characterize the performance of our localization system during path following, we compared the lateral path-tracking error estimated by the localization algorithm to that measured by GPS. Our GPS unit required line-of-site between the base station and the rover to send the real-time corrections and so we only have Real-Time Kinematic GPS (RTK-GPS) data for a small subset of routes. Figure 5 shows results from a 450-meter-long path that had RTK-GPS for both the teach pass and the repeat pass. A blue background highlights the portions of the repeat pass where global localization has failed.

These plots show two important characteristics of our algorithm. First, when the global localizations are successful, the estimated lateral path-tracking error has good agreement with the same quantity measured by GPS. We used a pair of Thales DG-16 GPS units rated at 0.4 meter circular error probable (50% of the data should be within 0.4 m of the true value). When globally localized, none of the differences are larger than 0.2 meters, 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 global localization is recovered. This is shown at around 360 meters traveled where the global localization drops out for nearly 15 meters. The speed of divergence is a function of the accuracy of our relative localization algorithm. During path following, we have seen the algorithm recover from lateral path-tracking errors of 1.5 meters and global localization dropouts of up to 40 meters. In each case, successful global localizations pull the estimate back toward global consistency and allow our algorithm to faithfully repeat long routes.

B. Initialization Convergence

To test the convergence properties of the *initialization* localization algorithm, we taught a single map on charac-

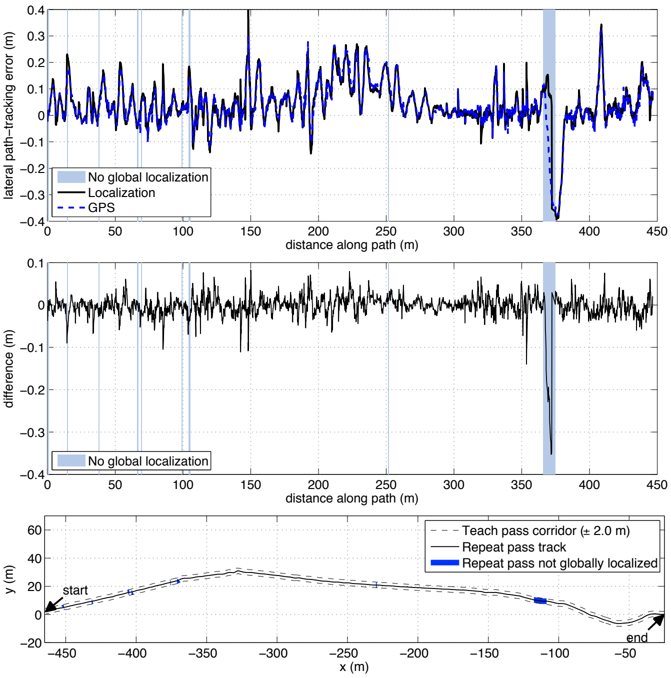

Fig. 5. From top to bottom: Lateral path-tracking error during a repeat pass as estimated by the localization algorithm and measured by RTK-GPS, the difference between these curves, and the track of the rover during this segment. The blue background highlights areas where the global localization step failed.

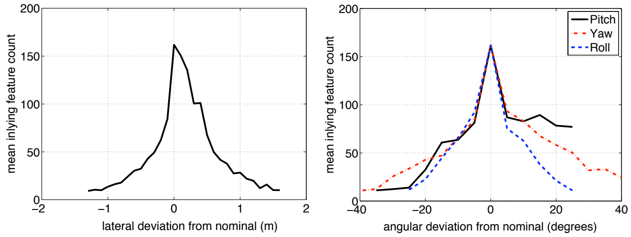

Fig. 6. Feature counts of the global localization algorithm when the camera is displaced laterally from the path, or rotated in place.

teristic terrain 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 localizations, and perturbed from this nominal position until the localization failed. Perturbations were introduced four ways: as lateral displacements from the path center (0.1 meter increments), and along vehicle-frame yaw, pitch, and roll axes (5° increments). At each increment, 200 localizations were processed.

Figure 6 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 global localization is possible with up to ±1 meter lateral displacement from the path, and over ±20° angular deviation in all of yaw, pitch, and roll.

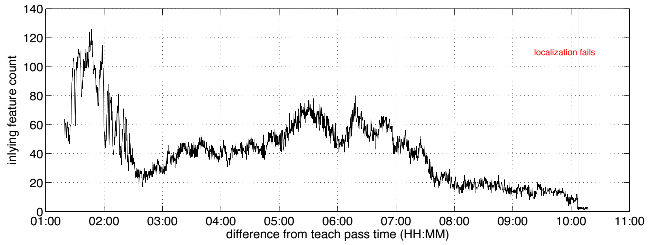

Fig. 7. Results of testing the global localization algorithm performance under changing lighting.

C. 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, feature matching is very difficult under extreme global lighting changes. To illustrate this, we taught a short route and set up a camera to capture an image and process a localization every 30 seconds. The inlying feature count is plotted against time passed in Figure 7. Ten hours after the teach pass, the localization module fails to find enough inlying features to localize. This confirms the lighting dependence that we have seen in during path-following 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.

D. Keypoint and Feature Usage

This section tries to shed some light on which keypoints are used by the algorithm to perform global localizations. To this end, we used data collected during the nine repeat passes of a single route 202 meters long. 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 am. 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 8 shows a histogram of track length during the learning phase (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 there are still a small number of features seen many more times. The long tail of this plot has been truncated. The longest track length was 102 frames.

During the repeat pass, we logged which features were used for global localization. Figure 8 shows the relationship

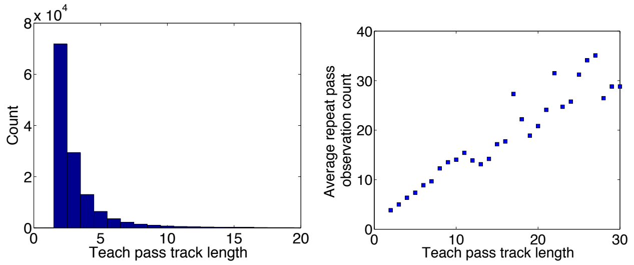

Fig. 8. Histogram of track lengths during the teach pass (left) and the mean repeat pass observation count plotted against teach pass track length (right).

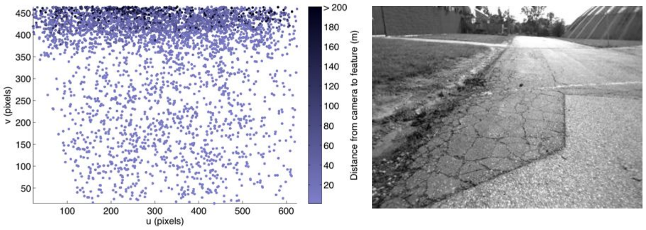

Fig. 9. A plot of the image locations of repeat pass observations (left) and a typical image from this sequence (right).

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 the global localizations, we plotted the global feature observations in image space. Figure 9 shows a clustering of features around the top of the image. When compared to a typical image from this route, it clearly shows that the majority of features used during the repeat pass are distant from the camera—horizon features. The appearance of a feature close to the camera changes drastically as the rover moves. Because the non-rotation-invariant SURF algorithm only accounts for translation and scaling *in image space*, it is not well tuned to track features on the ground plane as a rover drives toward them.

Horizon features are good for estimating rover orientation but, as stereo-based range accuracy decreases with distance from the camera, they are not good for estimating the rover position. In this sense, our algorithm works a lot like visual odometry with globally-consistent orientation updates. This also suggests a way forward for future work; it may be possible to reduce the map size by only using features that have been tracked for multiple frames. This would reduce the computational complexity of matching against the map and enable map matching to be performed more often. However, using only distant features would most likely require more accurate feature position and covariance estimates, or a separation of the rover orientation and position estimates as described in [18].

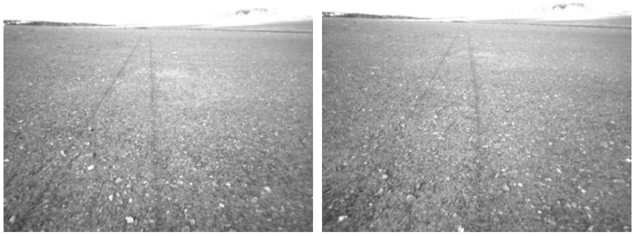

Fig. 10. Two consecutive images that caused a teach pass failure.

E. Teach Pass Failures

Occasionally, our mapping system was unable to create a cohesive motion estimate for a subsequence of images—an event we call teach pass failure. All of these failures were due to large displacement of the camera between images. Sometimes a processing backlog would cause our data-logging system to drop some 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, even short dropouts caused teaching failures. This is illustrated in Figure 10, which shows a pair of consecutive images that caused a failure. Three out of the five routes with teach-pass failures required no operator intervention while repeating the route; the rover simply drove to the end of the broken map, loaded the next map, relocalized, and continued.

F. Repeat Pass Failures

Failures during repeat passes were due to the changing appearance of the scene, mostly because of changing lighting conditions. When the rover was unable to localize globally for 50 meters, it would stop, signal the operator and then attempt to search nearby maps in an attempt to reset the localization. We encountered several situations where a route required manual interventions or failed to complete at one time of day but was autonomously repeated successfully when the lighting changed. For example, a 187 meter route at Lake Orbiter on Devon Island was in an area made up entirely of fist-sized rocks (Figure 11). Four hours after learning the route, the lighting had changed enough that our algorithm was unable to repeat the path. However, the next morning the path was retraced successfully. Flat areas with repetitive texture were particularly difficult under changing lighting conditions. The route already shown in Figure 10 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 while the second was attempted backward. Both failed at either end of the same stretch of terrain. Figure 12 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 meters laterally displaced from the path after 50 meters without global localization. Still, although the viewpoint was nearly the same, the combination of repetitive texture, different lighting, lack of horizon features, and lack

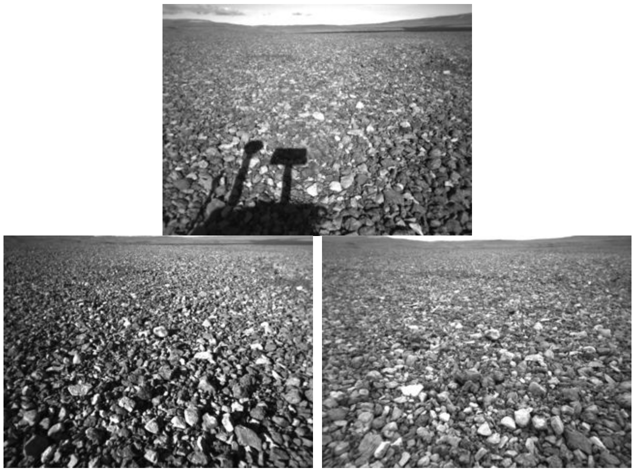

Fig. 11. Top: An image from the teach pass at Lake Orbiter. Bottom Left: An image from the failed repeat pass four hours after route learning. Bottom Right: an image from the successful repeat pass the next morning.

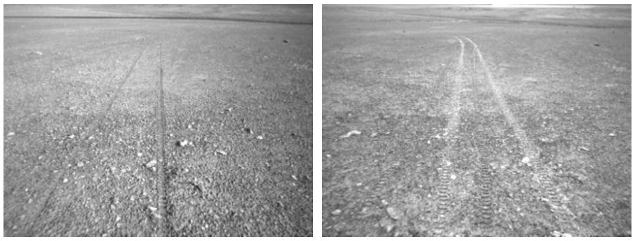

Fig. 12. Images from the teach (left) and repeat (right) passes of a route on Devon Island. The rover was unable to localize for 50 meters even though it was clearly on the path.

of unique three-dimensional objects in the scene caused the localization to fail.

V. Conclusion

We have described a complete localization system that can be used by a robot to map a path and then autonomously retrace the route any number of times in either direction. The system extends the basic visual odometry pipeline by abstracting its basic operating blocks and substituting in different parts for the feature database and numerical solution method. In doing this, we transform the pipeline into a complete mapping and localization system that may be used for over-the-horizon autonomous navigation.

Our evaluation shows that our localization algorithm produces estimates on par with RTK GPS when the system is globally localized. Furthermore, global localization is possible with up to ±1 meter lateral path tracking error and as much as 20° yaw, pitch, or roll. The single greatest difficulty in using our algorithm for path following outdoors is the impact of lighting on our implementation of the SURF algorithm. Future work should include some examination of mitigating the problem of lighting changes as well as some method of updating maps each time they are retraced.

VI. Acknowledgements

Funding for our field trials 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. Thanks to Pascal Lee and the entire Mars Institute/Haughton Mars Project team for support during our field trials on Devon Island.

References

[1] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. Van Gool, "Omnidirectional vision based topological navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 219–236, 2007.

[2] P. Newman, J. Leonard, J. D. Tardos, and J. Neira, "Explore and return: experimental validation of real-time concurrent mapping and localization," in *IEEE International Conference on Robotics and Automation, 2002.*, vol. 2, 2002, pp. 1802–1809.

[3] J. Marshall, T. Barfoot, and J. Larsson, "Autonomous underground tramming for center-articulated vehicles," *J. Field Robot.*, vol. 25, no. 6-7, pp. 400–421, 2008.

[4] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," *Int. J. Comput. Vision*, vol. 74, no. 3, pp. 237–260, 2007.

[5] S. Šegvić, A. Remazeilles, A. Diosi, and F. Chaumette, "A mapping and localization framework for scalable appearance-based navigation," *Comput. Vis. Image Underst.*, vol. 113, no. 2, pp. 172–187, 2009.

[6] A. M. Zhang and L. Kleeman, "Robust appearance based visual route following for navigation in large-scale outdoor environments," *Int. J. Rob. Res.*, vol. 28, no. 3, pp. 331–356, 2009.

[7] A. A. Argyros, K. E. Bekris, S. C. Orphanoudakis, and L. E. Kavraki, "Robot homing by exploiting panoramic vision," *Auton. Robots*, vol. 19, no. 1, pp. 7–25, 2005.

[8] J. Courbon, Y. Mezouar, and P. Martinet, "Indoor navigation of a non-holonomic mobile robot using a visual memory," *Autonomous Robots*, vol. 25, no. 3, pp. 253–266, 10 2008.

[9] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using view-sequenced route representation," *1996 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 83–88 vol.1, April 1996.

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

[11] K. Konolige, M. Agrawal, and J. Solà, "Large scale visual odometry for rough terrain," in *Proceedings of the International Symposium on Research in Robotics (ISRR)*, November 2007.

[12] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.

[13] D. Nistér, "Preemptive ransac for live structure and motion estimation," *Machine Vision and Applications*, vol. 16, no. 5, pp. 321–329, 2005.

[14] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Am. A*, vol. 4, no. 4, pp. 629–642, 1987.

[15] G. Sibley, L. Matthies, and G. Sukhatme, "A sliding window filter for incremental slam," in *Unifying Perspectives in Computational and Robot Vision*, ser. Lecture Notes in Electrical Engineering. Springer US, 2008, vol. 8, pp. 103–112.

[16] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[17] A. Diosi, A. Remazeilles, S. Segvic, and F. Chaumette, "Outdoor visual path following experiments," *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 4265–4270, November 2007.

[18] M. Kaess, K. Ni, and F. Dellaert, "Flow separation for fast and robust stereo odometry," in *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Kobe, Japan, May 2009.