# Vision-Based Global Localization and Mapping for Mobile Robots

Stephen Se, *Member, IEEE*, David G. Lowe, *Member, IEEE*, and James J. Little, *Member, IEEE*

***Abstract*—We have previously developed a mobile robot system which uses scale-invariant visual landmarks to localize and simultaneously build three-dimensional (3-D) maps of unmodified environments. In this paper, we examine global localization, where the robot localizes itself globally, without any prior location estimate. This is achieved by matching distinctive visual landmarks in the current frame to a database map. A Hough transform approach and a RANSAC approach for global localization are compared, showing that RANSAC is much more efficient for matching specific features, but much worse for matching nonspecific features. Moreover, robust global localization can be achieved by matching a small submap of the local region built from multiple frames. This submap alignment algorithm for global localization can be applied to map building, which can be regarded as alignment of multiple 3-D submaps. A global minimization procedure is carried out using the loop closure constraint to avoid the effects of slippage and drift accumulation. Landmark uncertainty is taken into account in the submap alignment and the global minimization process. Experiments show that global localization can be achieved accurately using the scale-invariant landmarks. Our approach of pairwise submap alignment with backward correction in a consistent manner produces a better global 3-D map.**

***Index Terms*—Global localization, map building, mobile robots, visual landmarks.**

## I. Introduction

WE HAVE previously proposed a vision-based SLAM (Simultaneous Localization And Mapping) algorithm [25] by tracking SIFT (Scale-Invariant Feature Transform) natural landmarks and building a three-dimensional (3-D) map simultaneously on our mobile robot equipped with Triclops[1], a trinocular stereo system.

Our previous algorithm builds a 3-D map continuously without maintaining the local image data, and hence does not allow backward correction. Therefore, it may have problems when long-term drifts occur and the robot closes the loop, i.e., returns to a previously mapped area.

In this paper, we consider global localization as a place recognition problem, by matching the SIFT features detected in the current frame to the pre-built SIFT database map. A Hough transform approach and a RANSAC approach are described and compared. Moreover, we improve the robustness of global localization by building submaps from multiple frames and using the submaps to align with the database map.

Instead of building a map continuously, we build multiple 3-D submaps which are subsequently merged together. We align them by applying our submap alignment approach used in global localization. In global localization, we are interested in the alignment only, but for map building, we also merge the input submaps. By aligning and merging submaps, we can improve the accuracy and efficiency of 3-D map construction.

When the robot returns to a previously mapped area, our framework of building and aligning multiple overlapping submaps allows backward correction between submaps in a consistent manner, even though we do not keep the local image data. We attribute the accumulated discrepancy to all the submap alignments, as errors have gathered over time. This loop closure constraint helps avoid the effects of error accumulation so that we can obtain a better global 3-D map.

The main contributions of this paper are: the use of distinctive visual SIFT features for localization and 3-D mapping, global localization using the Hough transform and RANSAC for matching groups of descriptors to a global map, and the backward correction of map alignment parameters taking into account uncertainty. The global localization and backward correction algorithms are also applicable to other methods that produce maps of point features with independent dense local descriptors.

Section II gives a brief literature survey on mobile robot localization and mapping. Section III overviews SIFT features, stereo matching and map building of our SLAM algorithm. The global localization algorithms based on the Hough transform and RANSAC are presented in Section IV with a comparison of computational costs. Map alignment for global localization using multiple frames is described in Section V. Map building by pairwise and incremental submap alignment is proposed in Section VI. Section VII describes the global minimization framework for backward correction taking into account the landmark uncertainty. Finally, we conclude and discuss some future work in Section VIII.

## II. Previous Work

### A. Localization

There are two types of localization: local and global. Local techniques aim at compensating for odometry errors during robot navigation. They require that the initial location of the

Manuscript received December 9, 2003; revised June 10, 2004. This paper was recommended for publication by Associate Editor N. Amato and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported by the Institute for Robotics and Intelligent System (IRIS III), a Canadian Network of Centres of Excellence, and by the Natural Sciences and Engineering Research Council of Canada.

S. Se is with MD Robotics, Brampton, ON L6S 4J3, Canada (e-mail: sse@mdrobotics.ca).

D. G. Lowe and J. J. Little are with the Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: lowe@cs.ubc.ca; little@cs.ubc.ca).

Digital Object Identifier 10.1109/TRO.2004.839228

[1] www.ptgrey.com

robot be approximately known. Global techniques can localize a robot without any prior knowledge about its position, i.e., they can handle the kidnapped robot problem, in which a robot is transported to some unknown location without any information about the motion. Global localization techniques are more powerful than local ones and allow the robot to recover from serious positioning errors.

Thrun *et al.* [29] developed the museum tour-guide robot MINERVA that employs EM to learn its map and Markov localization with camera mosaics of the ceiling in addition to the laser scan occupancy map. The Monte Carlo Localization method based on the CONDENSATION algorithm was proposed in [8]. Given a visual map of the ceiling, it localizes the robot globally using a scalar brightness measurement. These probabilistic methods use sensor information of low feature specificity in a two-dimensional (2-D) plane and require the robot to move around for the probabilities to gradually converge toward a peak, whereas our approach makes use of highly distinctive visual information and allows instant global localization.

Thrun *et al.* [30] developed a real-time algorithm combining the strengths of the EM and incremental algorithms. Their approach computes the full posterior over robot poses to determine the most likely pose. When closing cycles, backward correction is computed from the difference between the incremental guess and the full posterior guess. Baltzakis and Trahanias [2] presented an EM-based iterative approach for building feature maps by extracting lines and corners from laser data. Their global localization algorithm assigns Kalman tracks to multiple hypotheses about the robot state while letting discrete dynamics handle probabilistic relations among them.

There are previous approaches in which appearance-based models are learnt from many images and then images are recognized subsequently for mobile robot navigation. Hayet *et al.* [12] extracted and recognized visual landmarks for mobile robot navigation. Planar quadrangular landmarks are extracted from images and homography rectification is applied to obtain an invariant representation for the principal component analysis (PCA) learning stage. Kosecka *et al.* [14] employed gradient orientation histograms to capture the essential appearance information. A Learning Vector Quantization technique is applied to obtain sparser representations by selecting prototype vectors which best cover the class. During the recognition phase, new images are classified using a nearest neighbor test.

Some appearance-based works also compute robot pose. Sim and Dudek [26] learnt natural visual features for pose estimation. PCA is used to match landmarks which are sets of image thumbnails detected in the learning phase, for each robot grid position. Cobzas and Zhang [7] used a panoramic image-based model for robot localization. The panoramic model is constructed with depth and 3-D planarity information, while the matching is based on planar patches. Krose *et al.* [15] used panoramic images for probabilistic appearance-based robot localization. PCA is applied to hundreds of training images to extract the 15-dimensional feature vectors for Markov localization. These appearance-based methods differ from our approach as they require many more training images from a complete sample of positions.

### *B. Map-Building*

The general approach of map building is to incrementally integrate new data into the map. When each new frame is obtained, it is aligned to a cumulative global map [1]. The resulting map may become inconsistent as different parts of the map are updated independently.

Smith *et al.* [27] developed the stochastic map, which contains estimates of the spatial relationships, their uncertainties and their inter-dependencies. The Kalman Filter is applied to the state vector consisting of the robot position as well as all the features in the map and the covariance matrix containing all the cross-covariances between the features. However, the computational complexity is $\mathbf{O}(n^2)$ where $n$ is the number of features in the environment.

This is similar to bundle adjustment [32] in the photogrammetry and computer vision literature which refines a visual reconstruction to produce jointly optimal structure and viewing parameters. This is a large sparse geometric parameter estimation problem and all the structure and camera parameters are adjusted together in one bundle.

### *C. Alignment*

Lu and Milios [20] presented a 2-D laser scan alignment batch algorithm which aligns frames of sensor data to obtain a consistent map. They maintain all the local data together with their estimated poses so that inconsistency can be resolved later. Spatial relationships between local frames are obtained by matching pairwise laser scans and then the maximum likelihood criterion is applied to optimally combine all the spatial relations.

There has been a considerable amount of recent work on submap decomposition as a computationally efficient approach to large-scale SLAM. These submap-based approaches to mapping vary in how map fusion and map transition are tackled.

Leonard and Feder [18] proposed decoupled stochastic mapping by representing the environment in terms of multiple globally-referenced submaps. Techniques are developed to transfer vehicle state estimate information from one submap to another as it transitions between map regions. Williams *et al.* [33] developed the constrained local submap filter that creates an independent local submap of the features in the immediate vicinity, which is periodically fused into the global map. A large number of observations can then be fused in a single step.

Tardos *et al.* [28] proposed sequential map joining that builds a sequence of independent limited-size stochastic maps and joins them in a globally consistent way; their method can handle loop closure. Leonard and Newman [17] developed a new efficient algorithm that achieves consistency, convergence, and constant-time update with multiple submaps, while assuming known data association.

One of the appealing aspects of a hybrid metrical/topological approach to mapping and localization [5], [6] is that uncertain state estimates need not be referenced to a single global reference frame. Gutmann and Konolige [10] proposed a real-time method to reconstruct consistent global maps from dense laser range data. The techniques of scan matching, consistent pose estimation and map correlation are integrated for incrementally building maps, finding topological relations and closing loops.

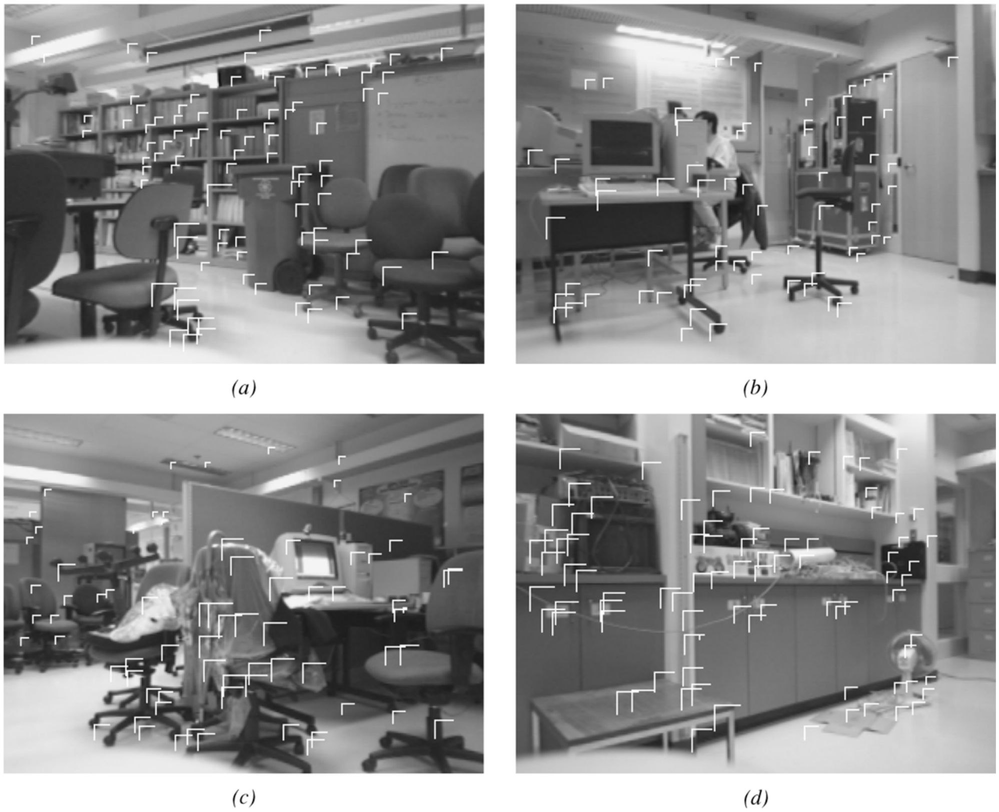

Fig. 1. Images for global localization experiments showing stereo matched SIFT landmarks, where the lines indicate the disparities (longer lines indicate closer features). (a) Case L2. (b) Case L4. (c) Case L6. (d) Case L8.

Bosse *et al.* [4] proposed a hybrid approach by using a graph where each vertex represents a local frame (a local environment map) and each edge represents the transformation between adjacent frames. Loop closing is achieved via an efficient map matching algorithm. Kuipers *et al.* [16] presented a hybrid extension to the spatial semantic hierarchy, using metrical SLAM methods to build local maps of small-scale space while topological methods are used to represent the structure of large-scale space. Their method creates a set of topological map hypotheses and can handle multiple nested large-scale loops.

Our approach also makes use of submaps, but differs from these works as we build 3-D submaps and our map also allows global localization to recover from localization failure.

## III. Simultaneous Localization and Mapping

Our vision-based mobile robot localization and mapping system uses SIFT visual landmarks in unmodified environments [25]. By keeping the SIFT landmarks in a database, we track the landmarks over time and build a 3-D map of the environment, and use these 3-D landmarks for localization at the same time.

### A. SIFT Features

SIFT was developed by Lowe [19] for image feature generation in object recognition. The features are invariant to image translation, scaling, and rotation, and are not sensitive to illumination changes and affine/perspective projection. These characteristics make them suitable landmarks for robust SLAM, since landmarks are observed over time from different angles, distances, or under different illumination when mobile robots move around in an environment. A subpixel image location, scale, and orientation are associated with each SIFT feature.

Previous approaches to feature detection, such as the widely used Harris corner detector [11], are sensitive to the scale of an image and therefore are not suited to building a map that can be matched from a range of robot positions. There has been considerable recent research on developing affine-invariant features [21], [24], but they have reduced stability to nonextreme affine changes compared to our features and have a much higher computational cost for detection. Recently a performance evaluation of various local descriptors [22] showed that SIFT feature descriptors perform best among them.

### B. SIFT Stereo

In our Triclops system, we have three images at each frame. In addition to the epipolar constraint and disparity constraint, we also employ the SIFT scale and orientation constraints for matching the right and left images. These resulting matches are then matched with the top image in the same manner. We can then compute the 3-D world coordinates relative to the robot for each feature. They can subsequently serve as landmarks for map building and tracking. Fig. 1 shows the stereo matched SIFT landmarks in some typical lab images of resolution 320 × 240.

### C. Map-Building

To build a map, we need to know how the robot has moved between frames in order to put the landmarks together coherently. The robot odometry can only give a rough estimate and it

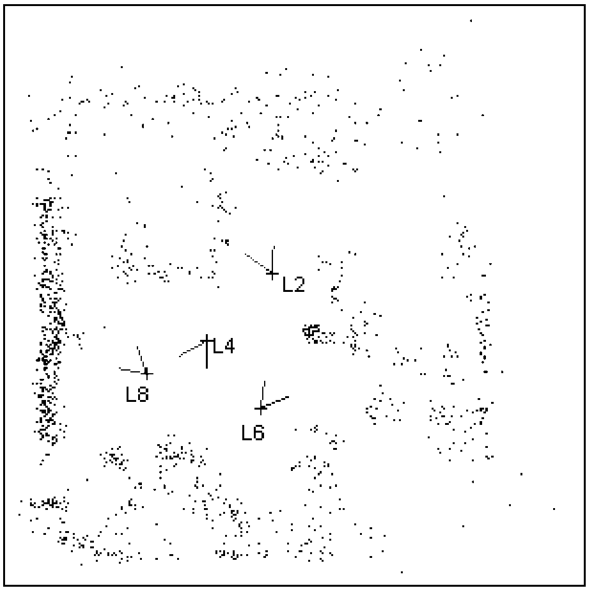

Fig. 2. Bird's eye view of the SIFT 3-D map showing the global localization results with the RANSAC approach. The "V"s indicate the robot field of view for the images in Fig. 1.

is prone to error. To find matches in the second view, the odometry allows us to predict the region to search for each match more efficiently.

Once the SIFT features are matched, we can use the matches in a least-squares procedure to compute a more accurate ego-motion and hence better localization. This will also help adjust the SIFT landmark coordinates for map building.

We track the SIFT landmarks and build a 3-D map while the robot moves around in our lab environment. Fig. 2 shows the bird's eye view of the map after 435 frames and there are 2783 SIFT landmarks in the database. The system currently runs at 2 Hz on a Pentium III 700 MHz processor. Readers are referred to [25] for further details.

## IV. Global Localization

For the kidnapped robot problem, the robot needs to detect that it has been kidnapped and then carries out global localization. Failing to track sufficient features indicates that the robot may be kidnapped or the environment has changed significantly and that global localization is required.

Global localization is similar to a recognition problem where the robot tries to match the current view to a previously built map. The SIFT features used here were originally designed for object recognition purposes, therefore these visual landmarks are very suitable for global localization.

### A. Local Image Characteristics

In order to recognize where the robot is, sufficiently distinctive features are required to identify scenes in the map. We use the highly specific feature vector obtained from the local image region [19]. The local and multi-scale nature of the features makes them insensitive to noise, clutter, and occlusion, while the detailed local image properties make them highly selective for matching to large databases.

Lowe's object recognition application used a feature vector computed from 8 orientations, each sampled over $4 \times 4$ grid of locations, so the total number of samples for each SIFT key is 128. For our application our experimental comparison of different sample sizes showed that a smaller vector is sufficiently discriminating in our environment. We use 4 orientations, each sampled over a $2 \times 2$ grid of locations, to reduce computation time. The total number of samples in each SIFT key vector is now $4 \times 2 \times 2$ or 16 elements.

Using this local image vector metric, we can simply compute the Euclidean distance measure between the vectors of two features to check whether they are below a matching threshold.

Stereo matching and frame-to-frame matching are based on the scale and orientation only to avoid extra computational burden, as consistent results can be obtained without further information.

The following sections describe two alternative methods for finding consistent sets of matches: the Hough transform and RANSAC. Their properties will be compared in later sections.

### B. Hough Transform Approach

Given a set of current SIFT features and a set of SIFT landmarks in the database, we search for the robot position that would have brought the largest number of landmarks into close alignment, treating global localization as a search problem.

The Hough transform [13] with a 3-D discretized search space $(X, Z, \theta)$ is used,where $X$ is the sideways translation, $Z$ is the forward translation and $\theta$ is the orientation. The algorithm is as follows:

- For each SIFT feature in the current frame, find the set of $N$ potential SIFT landmarks in the database that match, using the local image vector and the height above the ground plane as the preliminary constraints.
- For each of the potential matches, compute all the possible poses and place a vote in the corresponding Hough bins. Votes are placed in multiple bins as robot pose cannot be uniquely determined from just one match.
- Votes are also placed in the neighboring bins within the uncertainty region based on the landmark covariance.
- Select the top $K$ poses and carry out least-squares minimization with outlier removal to obtain pose estimates.
- Select the pose with maximum number of matches. This corresponds to a robot pose which can best match the most features to the database.

### C. RANSAC Approach

Global localization is performed by finding the robot pose supported by the most landmarks. This can be alternatively formulated as a hypothesis testing problem, where multiple pose hypotheses are considered and the best one corresponds to the pose which can match the most features in the current frame to the database.

RANSAC [9] has been used in many applications for model fitting, hypothesis testing and outlier removal. We employ RANSAC for global localization to test the pose hypotheses and find the inlier landmarks.

*1) Tentative Matches:* First, we create a list of tentative matches from the features in the current frame to the landmarks in the database. For each feature in the current frame, we find the landmark in the database which is closest in terms of the local image vector and has similar height.

*2) Computing the Alignment:* Next, we randomly select two tentative matches from the list and compute the alignment parameters $(X, Z, \theta)$ from them. Two tentative matches are required in this case, since for each match, we can obtain 2 equations with 3 unknowns

$$X = X_i - X_i' \cos\theta - Z_i' \sin\theta \tag{1}$$

$$Z = Z_i - Z_i' \cos\theta + X_i' \sin\theta \tag{2}$$

where $(X_i, Y_i, Z_i)$ is the landmark position in the database and $(X_i', Y_i', Z_i')$ is the feature position in the current frame in camera coordinates. With two matches, $i$ and $j$, we have

$$A\cos\theta + B\sin\theta = C \tag{3}$$

$$B\cos\theta - A\sin\theta = D \tag{4}$$

where $A = X_i' - X_j', B = Z_i' - Z_j', C = X_i - X_j, D = Z_i - Z_j$. If the two tentative matches are correct, the distance between two landmarks is invariant for this Euclidean transformation, so the following constraint is applied to each sample selection: $A^2 + B^2 \approx C^2 + D^2$. This efficiently eliminates many samples containing wrong matches from further consideration.

Solving (3) and (4), we obtain

$$\theta = \tan^{-1} \frac{BC - AD}{AC + BD}$$

and substituting this into (1) and (2) gives an alignment.

*3) Seeking Support:* Now we check all the tentative matches which support this particular pose $(X, Z, \theta)$.

First, we compute the landmark position for each match $k$ relative to this pose

$$X_p = (X_k - X)\cos\theta - (Z_k - Z)\sin\theta$$
$$Y_p = Y_k$$
$$Z_p = (X_k - X)\sin\theta + (Z_k - Z)\cos\theta$$

and then compute the image position $(r_p, c_p)$ and disparity $d_p$ for this landmark at this pose.

Match $k$ supports this pose if $(r_p, c_p)$ and $d_p$ are close to the measured image position $(r_m, c_m)$ and disparity $d_m$ for the feature in the current frame, i.e., $|r_p - r_m| < \Delta_r$ and $|c_p - c_m| < \Delta_c$ and $|d_p - d_m| < \Delta_d$ (currently $\Delta_r = 5, \Delta_c = 5, \Delta_d = 2$).

*4) Hypothesis With Most Support:* The random selection, alignment computation and support seeking steps are repeated $m$ times. The pose with most support is our hypothesis. We then proceed with least-squares minimization for the inliers that support this hypothesis and obtain a better estimate for the pose. Using this new pose estimate, we proceed with another least-squares minimization if more matches or lower least-squares error can be obtained.

The probability of a good sample $\tau$ for RANSAC [23] is given by

$$\tau = 1 - (1 - (1 - \epsilon)^p)^m \tag{5}$$

where $\epsilon$ is the contamination ratio (ratio of false matches to total matches), $p$ is the sample size and $m$ is the number of samples required. Recent work by Tordoff and Murray [31] showed that, in practice, this stopping condition usually underestimates the number of iterations required unless a guided sampling approach is employed. We will utilize the random sampling approach for the following analysis.

### D. Experimental Results

Using the database map built earlier covering a 10 m by 10 m area, we test the robot by placing it at various positions and let it localize itself globally. Both approaches give similarly good results. The following results $(X\text{cm}, Z\text{cm}, \theta^\circ)$ are obtained using the RANSAC approach $(\tau = 99\%, \epsilon = 0.7, p = 2, m = 50)$:

| Case | Measured Pose | Estimated Pose | Matches |
|---|---|---|---|
| L1 | (-10,120,-60) | (-13.3,127.6,-60.5) | 35 |
| L2 | (50,210,-25) | (54.3,208.9,-25.6) | 17 |
| L3 | (-15,130,-140) | (-16.0,134.9,-140.5) | 32 |
| L4 | (-80,60,-150) | (-75.7,68.8,-148.6) | 23 |
| L5 | (-100,0,130) | (-105.0,7.6,130.9) | 50 |
| L6 | (30,-70,40) | (31.3,-64.9,38.5) | 11 |
| L7 | (-170,20,-125) | (-175.2,21.8,-124.4) | 52 |
| L8 | (-210,0,-50) | (-207.6,8.3,-49.0) | 18 |

Measured pose is the ground truth measured manually. The average Euclidean translation error is 7 cm and the average rotation error is around 1° for these 8 cases. These errors could be further reduced by using higher image resolution but they are sufficient for our navigation requirement.

We currently set a minimum of 10 matches for a reliable estimation. Fig. 1 shows some of the test images for these cases while Fig. 2 shows these results visually, indicating the robot location and orientation relative to the database map.

Global localization can fail when the robot is facing some landmarks which were previously viewed from very different directions during map building. Therefore, landmarks all over the environment should be observed from multiple views during map building, to obtain a richer database map.

### E. Comparison

We would like to compare the computational efficiency of these two approaches of global localization using SIFT features, i.e., the Hough transform versus RANSAC. Moreover, we also compare with the cost of the Hough transform and RANSAC approaches using nonspecific features. The following run-time results are based on a Pentium III 700 MHz processor.

*1) Hough Transform With SIFT:* In this approach, for each of the $N$ potential matches, we need to vote for multiple robot poses that could have observed this landmark. Let $t_1$ be the pose computation time for one potential match for all features in the current frame at all orientations. As we find the best $N$ matches for each feature, the pose computation takes $Nt_1$.

Assume that it takes $t_2$ to find the highest peak in the Hough space and do least-squares fitting with outlier removal. We can simply go through the bins $K$ times, so the time required is $Kt_2$. Part of this time can be saved by maintaining the top $K$ bins during the voting process. There is some overhead of $t_3$ as well and the total time taken is $Nt_1 + Kt_2 + t_3$. With $K =$

$10, N = 5, t_1 = 0.025, t_2 = 0.05, t_3 = 0.1$, the total time taken is around 0.725 s.

*2) Hough Transform With Nonspecific Features:* In this case, we do not have any feature specificity, so we need to consider all possible matches between the current frame features and the database landmarks. The computational cost is linear to the number of landmarks in the database and the total time taken is: $0.025 \times 2500 + 0.05 \times 10 + 0.1 = 63.1$ s.

*3) RANSAC With SIFT:* For the RANSAC approach, the computational cost is affected greatly by how many times we need to sample, which depends on the contamination ratio.

With $p = 2$ and $\epsilon = 1 - c/f$ where $f$ is the number of features in the current frame and $c$ is the number of correct matches, we can re-write (5) as:

$$m_1 = \frac{\ln(1-\tau)}{\ln(1-c^2/f^2)} \approx -\frac{f^2}{c^2}\ln(1-\tau) \qquad (6)$$

using Taylor's expansion as an approximation.

For each random selection, we need to check the support from all the $f$ tentative matches, so the time required is $(f_t + t_4)$ where $f_t$ is the time to check for support from $f$ tentative matches and $t_4$ is a fixed overhead. Therefore, the total cost is $(f_t + t_4)m_1 + t_5$ where $t_5$ is the time to create the list of tentative matches.

In our case, $f_t = 1.4 \times 10^{-5}, t_4 = 10^{-5}, t_5 = 0.02$, the total time is therefore $(1.4 \times 10^{-5} + 10^{-5})m_1 + 0.02$. Assuming a contamination ratio of 0.70, to achieve 99% probability of a good sample, $m_1$ is 50 and the time is around 0.02 s. RANSAC is much more efficient than the Hough transform when SIFT features are used.

*4) RANSAC With Nonspecific Features:* When nonspecific features are used, we need to consider all the possible matches between the current frame and the database landmarks. The contamination ratio $(1 - c/Nf)$ now is significantly higher, as the number of all possible matches is $Nf$ where $N$ is the number of landmarks in the database. Therefore, the number of samples required in this case is given by:

$$m_2 = \frac{\ln(1-\tau)}{\ln(1-c^2/N^2f^2)} \approx -\frac{N^2f^2}{c^2}\ln(1-\tau) \approx N^2 m_1$$

using Taylor's expansion and (6).

For each random selection, we need to check support from all possible matches, i.e., from $Nf$ matches. The time required now is $(Nf_t + t_4)$. It is no longer necessary to create the tentative match list, so the total cost is $(Nf_t + t_4)m_2$. The computation time increases very rapidly, by the cube of the database size. The time required when $N = 2500$ is $(2500^3 \times 1.4 \times 10^{-5} + 2500^2 \times 10^{-5}) \times 50 \approx 10\,000\,000$ s. The contamination ratio now is 0.999 88, making $m_2$ extremely large.

*5) Discussion:* The computation time for the two approaches are compared in Fig. 3 which shows that RANSAC is more efficient for distinctive features (low contamination ratio), whereas the Hough transform is more efficient for nonspecific features (high contamination ratio). Using SIFT features is much more efficient than using nonspecific features for both approaches.

The graph assumes that the feature extraction cost is constant for all contamination ratios. SIFT feature extraction only takes slightly more time than the extraction of nonspecific features, hence the advantage of having a more discriminating descriptor outweighs the cost of calculating the image metrics.

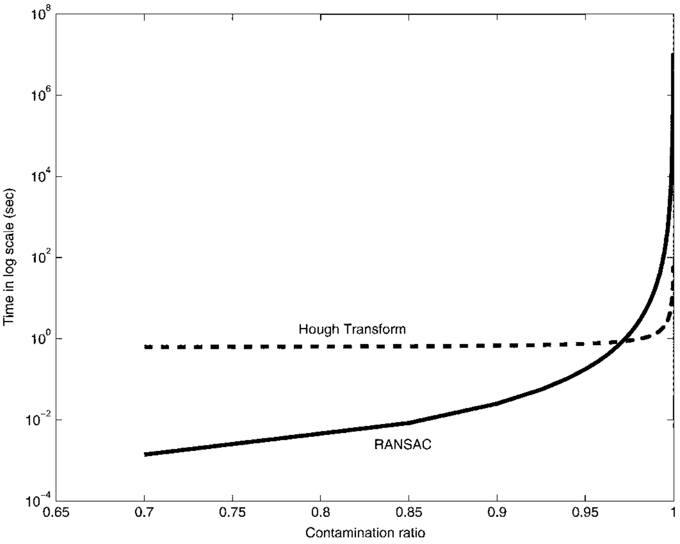

Fig. 3. Effect of contamination ratio on computation time of the Hough transform and RANSAC.

SIFT features and nonspecific features are the two extremes of feature distinctiveness. There are features in between which offer some specificity such as lines and color corners. Therefore, the complexity trade-off between RANSAC and the Hough transform depends on the particular application and how specific the features are.

Apart from the higher computational cost when less specific features are used, global localization is more difficult to achieve when only using information from one frame, as multiple possible robot poses may not be reliably differentiated. For the brightness measurements in [8], stochastic localization methods are required to localize the robot gradually while it moves around.

## V. Map Alignment

When the robot is facing a scene with very few SIFT landmarks, it might not be able to localize itself globally using just the current frame. To achieve more robustness, we can build a small submap of a local region from multiple frames instead and then align this submap to the database map.

To align two maps, we employ an algorithm very similar to global localization above. Either the Hough transform approach or the RANSAC approach can be applied, but we consider the RANSAC approach here due to its efficiency when used with SIFT features. It is basically the same as in global localization, except that during the support seeking stage we now use the world positions of the landmarks to check for support, instead of the image coordinates.

In this experiment, when the robot wants to localize itself globally, it rotates a little bit, from $-15$ degrees to 15 degrees and builds a submap of this local region using information from multiple frames. Fig. 4 shows the various submaps built at several test positions. There are 411 landmarks, 207 landmarks, 383 landmarks and 270 landmarks in the submaps respectively.

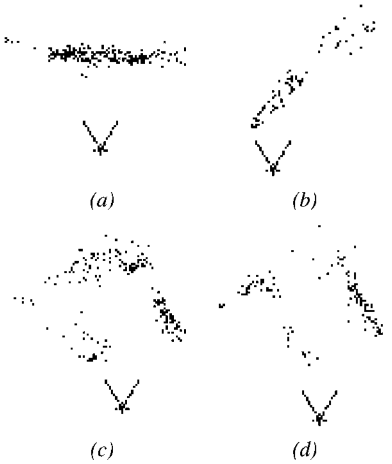

Fig. 4. Submaps built at test positions. (a) Case M1. (b) Case M2. (c) Case M3. (d) Case M4.

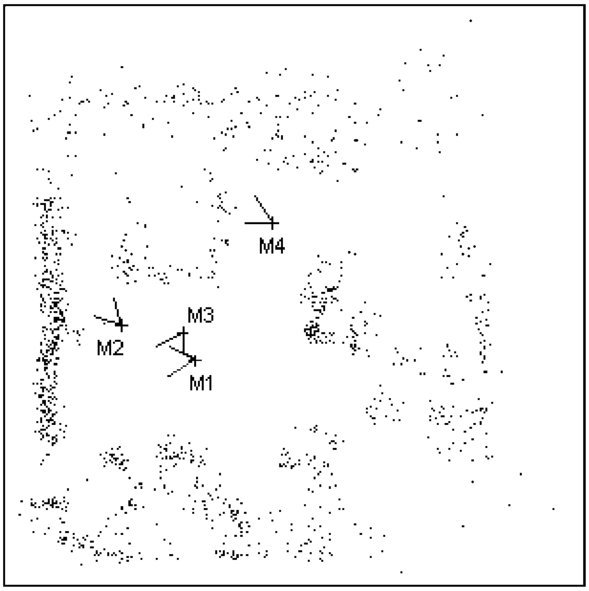

Fig. 5. Global localization by map alignment.

There are significantly more landmarks than in just one frame, typically around 70. Despite the stability and repeatability of SIFT features, the submaps contain some spurious landmarks while they are built over multiple frames.

Map alignment using RANSAC is then carried out between these submaps and the database map, and we obtain the following results $(X\mathrm{cm}, Z\mathrm{cm}, \theta^\circ)$:

| Case | Measured Pose | Estimated Pose | Matches |
|---|---|---|---|
| M1 | (-110,30,-90) | (-104.8,30.5,-92.2) | 191 |
| M2 | (-270,100,-45) | (-259.7,101.8,-43.5) | 32 |
| M3 | (-130,100,-150) | (-125.9,90.3,-146.9) | 143 |
| M4 | (60,310,-65) | (56.9,312.8,-63.5) | 44 |

We can see that very good alignments are obtained with many matches in all cases. These global localization results are shown visually in Fig. 5. If only the current frame is used for global localization here, there are insufficient matches for a reliable estimate in cases M2 and M4.

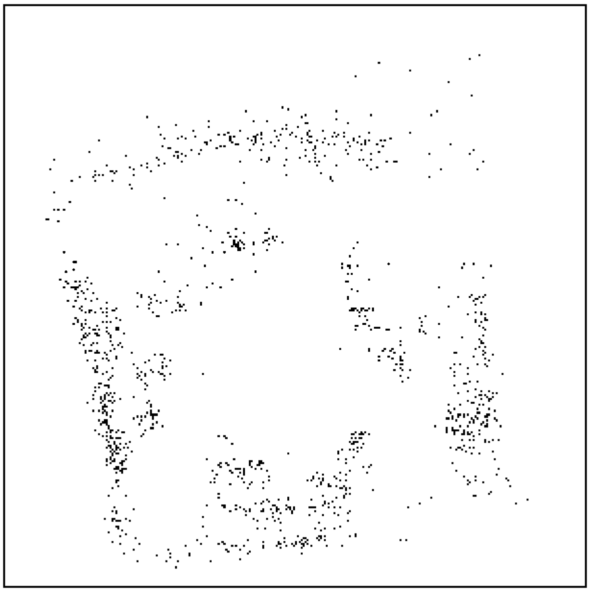

Fig. 6. Map built without taking into account slippage occurrences.

## VI. Building Submaps

For map building in large environments over time, there are considerable errors possible due to poor image matching (such as featureless walls or someone walking in front of the camera) and long-term drifts. To correct these errors, we can build multiple 3-D submaps, which are then aligned and merged together.

Fig. 6 shows the map built without taking into account poorly matched regions where some parts of the map are skewed. Three rotational slippages of around 5 degrees clockwise each are intentionally added at 90, 180, and 270 degrees robot orientation. By detecting the regions without sufficient matches, we can preferentially divide the map at those points so that the map alignment process can then fix the problem.

Let $F$ be a function which returns the number of matches between the current frame and the database given the current odometry position $\mathbf{p}$. When an unreliable matching occurs, the number of matches will be low at $\mathbf{p}$ but significantly higher at a nearby position $\mathbf{p}+\mathbf{e}$ (for a small $\mathbf{e}$). The condition is therefore

$$F(\mathbf{p}) < t \wedge F(\mathbf{p}+\mathbf{e}) \gg F(\mathbf{p})$$

where $t$ is a threshold and $\gg$ denotes "significantly higher." The second condition is necessary, as the first condition can be satisfied even when someone is blocking the camera view.

To cater for the effect of discontinuity in map building, we can simply estimate the robot pose based on the current frame as in global localization, and use it to correct the odometry for subsequent frames. However, we employ an alternative method which starts building a new map from scratch whenever a discontinuity is detected. Afterwards, all the submaps are aligned and combined to obtain a complete map. This approach is more robust as the discontinuity estimation is based on submap to submap alignment rather than frame to map alignment and hence more information can be utilized.

Using this method, we have obtained four submaps in this case, as shown in Fig. 7, due to the three slippages. They are in

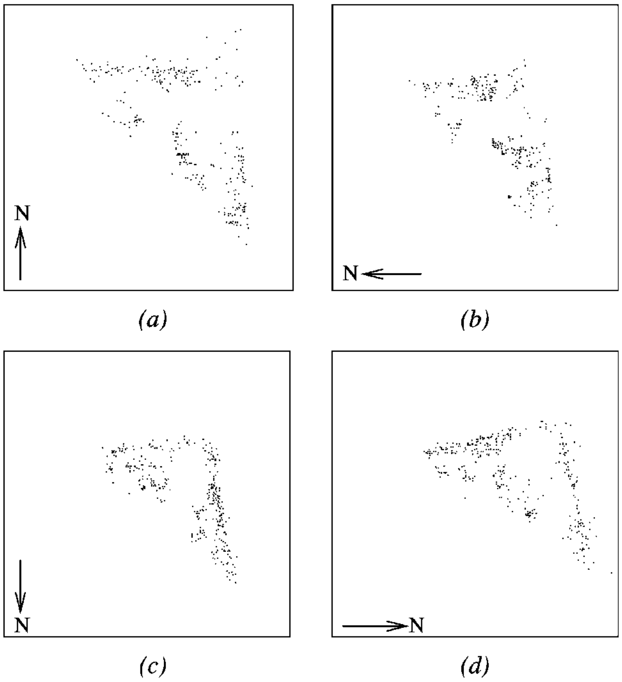

Fig. 7. Separate submaps are built due to slippage occurrences.

different coordinates now, since the submap coordinates are the robot coordinates at the initial position for each submap.

To avoid long-term drifts over time, this division can be done on a regular basis even if no discontinuity is detected, by building submaps every $M$ frames as described in Section VII-B.

### A. Pairwise Alignment

Using our map alignment algorithm in Section V, we can align two submaps together, provided there is some overlap. Since we terminate building the previous submap and then initiate building the current submap immediately, the last frame of the previous submap overlaps substantially with the first frame of the current submap, therefore some overlapping landmarks almost certainly exist. In global localization, we are interested in the alignment only, but for map building, we also merge the two input submaps together to obtain a combined map.

For pairwise alignment, we align each consecutive pair of submaps, and combine them based on the transformation from submap 1 to submap 2, from submap 2 to submap 3, and from submap 3 to submap 4.

In the map alignment algorithm, we assume no positional information nor odometry information of the robot, but just use the two input submaps, which contain highly specific information about the 3-D landmarks in the environments. Therefore, we can align the submaps correctly irrespective of the slippages. Fig. 8(a) shows the pairwise alignment results where the map is much better and unskewed. Submaps 1, 2, 3, and 4 occupy the top right, bottom right, bottom left and top left portions of the map, respectively.

### B. Incremental Alignment

For incremental alignment, we align and combine submaps 1 and 2 to obtain a new map, and then align this new map with submap 3 to obtain a new map, and so on. Fig. 8(b) shows the incremental alignment result and it looks very similar to the pairwise alignment result.

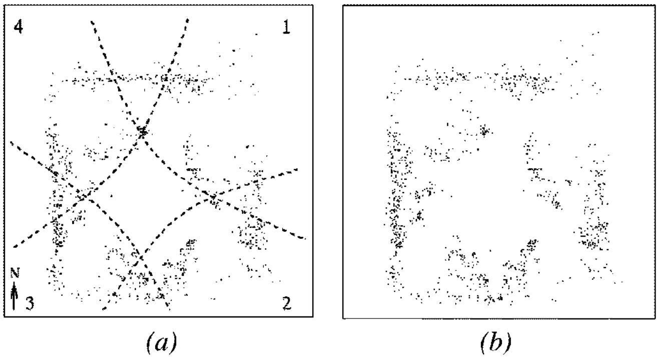

Fig. 8. (a) Pairwise alignment map for submaps in Fig. 7, with the submap composition indicated. (b) Incremental alignment map for submaps in Fig. 7.

In pairwise alignment, the alignment of the current submap depends only on the single previous submap, but in incremental alignment, the alignment of the current submap depends on all the previous submaps covering that region.

When submap 4 is aligned in this case, its landmarks are matched with those in submap 3 as well as those in submap 1, since we have rotated one revolution. On careful comparison of Fig. 8(a) and (b), we can see that submap 4 has been pulled in a little bit toward submap 1 in Fig. 8(b).

Pairwise alignment and incremental alignment results are the same if each submap overlaps only with the previous submap, but different if the robot closes the loop.

## VII. Closing the Loop

Closing the loop means revisiting a previously observed scene. It can be detected by checking if there is a significant overlap of landmarks between the current submap and the initial submap. When detected, we can find out the accumulated error over time and determine a correction, which should be spread out throughout each intermediate alignment because errors have gathered over time. We employ a global minimization strategy to do backward correction to all the submap alignments.

### A. Global Minimization

The submaps should be kept individually but not merged together to allow subsequent backward correction. The incremental alignment requires merging the previous submaps before the next alignment. Therefore, we apply the pairwise alignment to each consecutive pair of submaps to find the pairwise alignment. All the submaps are merged together at the end after the backward correction step has adjusted all the alignments.

For submaps $1, 2, \ldots, n$ where submap $n$ closes the loop, i.e., submap $n$ goes back to the scene observed by submap 1 in the beginning, we firstly find the pairwise alignments as before. We also find the pairwise alignment between submap $n$ and submap 1, and obtain $n$ transformations in total. Let $\mathbf{T}_i$ denote the coordinate transformation for aligning submap $i$ to submap $(i + 1)$, or submap $n$ to submap 1 when $i$ equals $n$.

For a perfect alignment, we have the following constraint:

$$\mathbf{T}_1\mathbf{T}_2\ldots\mathbf{T}_{n-1}\mathbf{T}_n = \mathbf{I} \tag{7}$$

where $\mathbf{I}$ is a 3 × 3 identity matrix.

During the pairwise alignment, each $\mathbf{T}_i$ is obtained independently from the least-squares minimization of the inlier matches between submap $i$ and submap $(i+1)$. To enforce the constraint given by (7), we set up a matrix consisting of this constraint as well as all the local pairwise alignments. We then minimize this to obtain alignments which can best satisfy this constraint globally but still conform to the local constraints due to the pairwise alignments.

We employ Newton's method which computes a vector of corrections $\mathbf{c}$ to be subtracted from the current estimate, i.e., the pairwise alignment estimate. Given a vector of error measurements $\mathbf{e}$ between the expected position of the SIFT landmarks and the matched position observed, and the deviation from our global constraint, we would like to solve for $\mathbf{c}$ that would eliminate this error. Based on the assumption of local linearity, the effect of each parameter correction $c_i$ on an error measurement will be $c_i$ multiplied by the partial derivative of the error with respect to that parameter. Therefore, we would like to solve for $\mathbf{c}$ in $\mathbf{Jc} = \mathbf{e}$ where $\mathbf{J}$ is the Jacobian matrix $J_{i,j} = \partial e_i / \partial x_j$.

Each row of this matrix equation states that one measured error $e_i$ should be equal to the sum of all the changes in that error resulting from the parameter corrections. If all these constraints can be simultaneously satisfied, then the error will be reduced to zero after subtracting the corrections.

If there are more error measurements than parameters (in this case $3n$ as there are 3 parameters for each alignment), this system of equations is overdetermined, and we will find a $\mathbf{c}$ that minimizes $|\mathbf{Jc} - \mathbf{e}|^2$. This minimization is achieved by solving

$$\mathbf{J}^\top\mathbf{Jc} = \mathbf{J}^\top\mathbf{e} \tag{8}$$

assuming the original nonlinear function is locally linear over the range of typical errors.

To include the constraint in (7) into our framework, we expand the matrix equation into several scalar equations

$$\begin{bmatrix} \cos\theta_1 & \sin\theta_1 & x_1 \\ -\sin\theta_1 & \cos\theta_1 & z_1 \\ 0 & 0 & 1 \end{bmatrix} \cdots \begin{bmatrix} \cos\theta_n & \sin\theta_n & x_n \\ -\sin\theta_n & \cos\theta_n & z_n \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I}$$

where $(x_i, z_i, \theta_i)$ are the alignment parameters from submap $i$ to submap $(i+1)$, or submap $n$ to submap 1 when $i$ equals $n$. We can obtain three independent scalar constraints to minimize

$$\begin{aligned} e_1 &= \sin(\theta_1 + \cdots + \theta_n) \\ e_2 &= x_1 + x_2\cos\theta_1 + z_2\sin\theta_1 + x_3\cos(\theta_1+\theta_2) \\ &\quad + z_3\sin(\theta_1+\theta_2) + \cdots + x_n\cos(\theta_1 + \cdots \\ &\quad + \theta_{n-1}) + z_n\sin(\theta_1 + \cdots + \theta_{n-1}) \\ e_3 &= z_1 - x_2\sin\theta_1 + z_2\cos\theta_1 - x_3\sin(\theta_1+\theta_2) \\ &\quad + z_3\cos(\theta_1+\theta_2) + \cdots - x_n\sin(\theta_1 + \cdots \\ &\quad + \theta_{n-1}) + z_n\cos(\theta_1 + \cdots + \theta_{n-1}) \end{aligned}$$

These three constraints will correspond to the RHS of the first three rows of our matrix. Let $m_i$ be the number of matches between submap $i$ and submap $(i+1)$. For each of the local pairwise alignments, we augment our matrix system with $2m_i$ rows as we need one row for the X error and one row for the Z error for each match. Let the $j$th landmark at $(X_i, Z_i)$ of submap $i$ be matched with $(X_{i+1}, Z_{i+1})$ of submap $(i+1)$, we have

$$\begin{aligned} e_{g(i)+2j-1} &= X_{i+1}\cos\theta_i + Z_{i+1}\sin\theta_i - X_i \\ e_{g(i)+2j} &= Z_{i+1}\cos\theta_i - X_{i+1}\sin\theta_i - Z_i \end{aligned}$$

where $g(i) = 3 + 2m_1 + 2m_2 + \cdots + 2m_{i-1}$.

$\mathbf{J}$ is a $3 + 2\sum_{i=1}^{i=n} m_i$ by $3n$ matrix whose $i$th row is

$$\begin{bmatrix} \frac{\partial e_i}{\partial x_1} \frac{\partial e_i}{\partial z_1} \frac{\partial e_i}{\partial \theta_1} \cdots \frac{\partial e_i}{\partial x_n} \frac{\partial e_i}{\partial z_n} \frac{\partial e_i}{\partial \theta_n} \end{bmatrix}$$

The computation of these partial derivatives is done analytically based on the $\mathbf{e}$ above. Once $\mathbf{e}$ and $\mathbf{J}$ are determined, we can compute $\mathbf{J}^\top\mathbf{J}$ and $\mathbf{J}^\top\mathbf{e}$ and then can solve (8) for the correction terms $\mathbf{c}$.

For the experiments carried out, the pairwise alignment is good enough so that a single iteration is sufficient. In general, this correction can be repeated if necessary, by using the current corrected estimate for the next iteration.

### *B. Landmark Uncertainty*

While the submaps are built, a covariance matrix for each 3-D landmark is kept [25]. Therefore, we can incorporate this information into the pairwise alignment as well as into the backward correction procedure.

During pairwise alignment, we take into account the covariances of the matching 3-D landmarks and employ a weighted least-squares minimization instead. This will also give us the covariance of the pairwise alignments.

The weighted least-squares equation is given by

$$\mathbf{WJc} = \mathbf{We} \tag{9}$$

where $\mathbf{W}$ is a diagonal matrix consisting of the inverse of the standard deviation of the measurements, assuming that landmarks are independent. The covariance of the solution is given by $(\mathbf{J}^\top\mathbf{W}^\top\mathbf{WJ})^{-1}$.

For our global minimization, we can compute the covariance of the three scalar constraints from the uncertainty of each pairwise alignment based on first order error propagation [3]

$$\begin{aligned} \mathrm{cov}(e_1) &= \left(\cos^2\sum_{i=1}^{i=n}\theta_i\right)\left(\sum_{i=1}^{i=n}\mathrm{cov}(\theta_i)\right) \\ \mathrm{cov}(e_2) &= \mathrm{cov}(x_1) + \cos^2\theta_1\mathrm{cov}(x_2) + x_2^2\sin^2\theta_1\mathrm{cov}(\theta_1) \\ &\quad + \sin^2\theta_1\mathrm{cov}(z_2) + z_2^2\cos^2\theta_1\mathrm{cov}(\theta_1) + \cdots \\ \mathrm{cov}(e_3) &= \mathrm{cov}(z_1) + \sin^2\theta_1\mathrm{cov}(x_2) + x_2^2\cos^2\theta_1\mathrm{cov}(\theta_1) \\ &\quad + \cos^2\theta_1\mathrm{cov}(z_2) + z_2^2\sin^2\theta_1\mathrm{cov}(\theta_1) + \cdots \end{aligned}$$

Each of these three rows is also multiplied by the total number of submaps we are aligning, so that they contribute the appropriate weights. We also have the covariance matrix information for each 3-D landmark for the rest of the matrix. We can then carry out a weighted least-squares minimization on the whole matrix, as in (9).

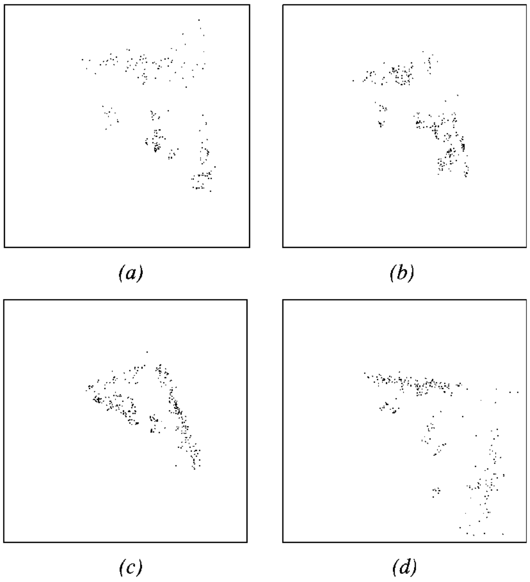

Fig. 9. Submaps built every 30 frames.

For the experiment above, we compute the product of all the pairwise alignments obtained originally, i.e.,

$$\mathbf{T}_1\mathbf{T}_2\mathbf{T}_3\mathbf{T}_4 = \begin{bmatrix} 0.9988 & -0.0489 & 0.0545 \\ 0.0489 & 0.9988 & 0.0885 \\ 0 & 0 & 1 \end{bmatrix}$$

which corresponds to a (5.45 cm, 8.85 cm) translational and 2.8 degrees rotational misalignment. For the weighted least-squares pairwise alignment, the misalignment becomes (3.00 cm, 5.92 cm, 0.43 deg).

This is better because, by taking into account the uncertainty of the matching landmarks, we can trust the more reliable landmarks more, whereas previously each landmark is trusted equally. The misalignment is further improved to (0.15 cm, 0.37 cm, 0.03 deg) for the weighted least-squares alignment with backward correction. We can now trust the more reliable pairwise alignment more since not all the pairwise alignment estimates are equally reliable.

The whole process is fast and it only takes 0.12 s on a Pentium III 700 MHz processor, excluding file I/O time. Each RANSAC pairwise alignment takes around 0.03 s to align submaps with several hundred landmarks each, and the global minimization takes less than 0.01 s.

The complexity of our approach increases by the square of the number of submaps, not by the square of the number of landmarks. Even if we do not have the pairwise alignments as the initial estimate but start with a zero vector, it still converges to the same result within 10 iterations for our experiments.

To avoid drift accumulation, we can build a new submap every $M$ frames (in this case $M = 30$) and combine the submaps together afterwards using this weighted least-squares approach. Fig. 9 shows the 4 submaps, each of them constructed from 30 frames. The pairwise alignment result has a misalignment of (0.40 cm, 7.48 cm, 7.35 deg), but with the backward correction, the misalignment is reduced to just (0.23 cm, 1.59 cm, 0.45 deg).

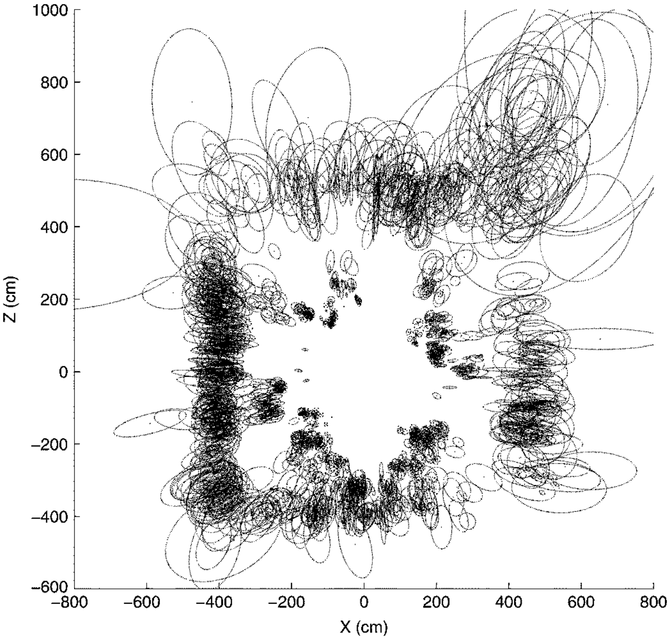

Fig. 10. Bird's eye view of 3-D map resulting from backward correction alignment with weighted least-squares for submaps in Fig. 9, showing the uncertainty ellipses of the landmarks. Note that the smallest ellipses represent the most reliable landmarks.

Uncertainty for the SIFT landmarks is propagated to the resulting map, as shown in Fig. 10. The uncertainty ellipsoids of the 3-D landmarks are projected as ellipses in the bird's eye view. The error ellipses represent one standard deviation of error. Uncertainty for landmarks closer to the robot tend to be lower, as expected for landmarks with larger disparities.

## VIII. Conclusion

In our previous SLAM work [25], we built a database map with distinctive SIFT landmarks, and they were shown to be good natural landmarks for tracking and localization over time. In this paper, we proposed a Hough transform approach and a RANSAC approach for global localization, demonstrating that the robot can globally localize itself well using highly specific SIFT features. We then investigated the computational costs of these two approaches and found that RANSAC is much more efficient than the Hough transform for matching distinctive features, whereas RANSAC is significantly worse for matching nonspecific features. Experiments show that global localization can be achieved with just the current frame data in feature-rich environments, thanks to the distinctive SIFT features.

Our use of distinctive visual features eliminates the data association problems commonly seen in other methods that use corners or lines for mapping. An individual SIFT feature is already specific, so a combined set of them is very distinctive and serves as a type of location fingerprint. The current local image vector size used is 16, but it can be increased if needed to increase the feature specificity for larger environments. Nevertheless, for symmetric environments or when there is a lack of features, global localization with the current frame may be uncertain and the robot should rotate or move around. It can then build a small submap of the local region to match to the database for more robustness. Moreover, we proposed a map building algorithm based on aligning and combining 3-D submaps using SIFT features. We can avoid the effect of drifts and slippage by aligning maps containing highly specific landmarks of the environment. On closing the loop, our framework will carry out a backward correction, attribute errors to all the pairwise alignments according to the landmark uncertainty and obtain a better 3-D map.

As the SIFT features are highly distinctive, even very few matches can provide a good alignment, therefore, it should work in sparse environments. Currently, no odometric information is used for map alignment, but for sparse environments and environments with many similar features, we can use the odometry to verify the map alignment.

Our pairwise alignment and backward correction are similar to the scan alignment and maximum likelihood optimization in [20] and the scan matching and map correlation in [10]. These algorithms, as well as many described in Section II-C, are developed mainly for dense 2-D range data obtained from laser or sonar and are not applicable to sparse 3-D data from vision.

Integrating new data to the map incrementally and bundle adjustment using all image frames are two extremes of map building. Incremental map building does not require keeping any information from each frame and, as a result, it does not allow any backward correction when we close the loop. It has low storage and computational costs, but may lead to an inconsistent map. On the other hand, bundle adjustment requires keeping image information from each frame but it allows backward correction at each frame. It has high storage and computational costs.

Our approach is a practical solution that provides a trade-off between these two methods. It only requires information for each submap and allows backward correction between submaps. Backward correction within each submap is not necessary because while building each submap, odometry has been corrected locally based on the SIFT landmarks. The complexity of our approach increases by the square of the number of submaps, not by the square of the number of landmarks.

We have yet to experiment in extensive environments or more complicated loop closure scenarios. To detect loop closures without a costly search, we may maintain a topological relation between submaps to hypothesize the closures [16]. Our approach can then correct the pairwise alignments such that these closure constraints are met with the least-squares matching errors minimized.

Future work also includes extending the global localization and map alignment algorithms to 3-D, i.e., estimating all 6 degrees of freedom robot pose and alignment, for rovers in outdoor environments.

## REFERENCES

[1] N. Ayache and O. D. Faugeras, "Maintaining representations of the environment of a mobile robot," *IEEE Trans. Robot. Autom.*, vol. 5, no. 6, pp. 804–819, Dec. 1989.

[2] H. Baltzakis and P. Trahanias, "An iterative approach for building feature maps in cyclic environments," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, Oct. 2002, pp. 576–581.

[3] P. R. Bevington and D. K. Robinson, *Data Reduction and Error Analysis for the Physical Sciences*, 2nd ed. New York: McGraw-Hill, 1992.

[4] M. Bosse *et al.*, "An atlas framework for scalable mapping," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA'03)*, Taipei, Taiwan, Sep. 2003, pp. 1899–1906.

[5] K. Chong and L. Kieeman, "Large scale sonarray mapping using multiple connected local maps," in *Proc. Int. Conf. Field and Service Robotics*, Canberra, Australia, Dec. 1997, pp. 538–545.

[6] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE Trans. Robot. Autom.*, vol. 17, no. 2, pp. 125–137, Apr. 2001.

[7] D. Cobzas and H. Zhang, "Cylindrical panoramic image-based model for robot localization," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Maui, HI, Oct. 2001, pp. 1924–1930.

[8] F. Dellaert *et al.*, "Using the condensation algorithm for robust, vision-based mobile robot localization," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'99)*, Fort Collins, CO, Jun. 1999.

[9] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography," *Commun. Assoc. Comp. Mach.*, vol. 24, pp. 381–395, 1981.

[10] J. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. IEEE Int. Symp. Computational Intelligence in Robot. Autom. (CIRA)*, Monterey, CA, Nov. 1999, pp. 318–325.

[11] C. J. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.*, Manchester, U.K., 1988, pp. 147–151.

[12] J. B. Hayet *et al.*, "Visual landmarks detection and recognition for mobile robot navigation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'03)*, Madison, WI, Jun. 2003, pp. 313–318.

[13] P. V. C. Hough, "Method and Means of Recognizing Complex Patterns," U.S. Patent 306,965,418, Dec. 18, 1962.

[14] J. Kosecka *et al.*, "Qualitative image based localization in indoors environments," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'03)*, Madison, WI, Jun. 2003, pp. 3–10.

[15] B. J. A. Krose *et al.*, "Omnidirectional vision for appearance-based robot localization," in *Proc. Int. Workshop on Sensor Based Intelligent Robots*. New York: Springer-Verlag, 2002, LNCS 2238, pp. 39–50.

[16] B. Kuipers *et al.*, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, New Orleans, LA, Apr. 2004, pp. 4845–4851.

[17] J. Leonard and P. Newman, "Consistent, convergent, and constant-time SLAM," in *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*, Acapulco, Mexico, Aug. 2003, pp. 1143–1150.

[18] J. J. Leonard and H. J. S. Feder, "A computational efficient method for large-scale concurrent mapping and localization," in *9th Int. Symp. Robot. Res.*, London, U.K., 1999.

[19] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th Int. Conf. Computer Vision (ICCV'99)*, Kerkyra, Greece, Sep. 1999, pp. 1150–1157.

[20] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, pp. 333–349, 1997.

[21] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. Eur. Conf. Computer Vision (ECCV)*, Copenhagen, Denmark, 2002, pp. 128–142.

[22] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'03)*, Madison, WI, Jun. 2003, pp. 257–263.

[23] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. New York: Wiley, 1987.

[24] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or how do I organize my holiday snaps?," in *Proc. Eur. Conf. Computer Vision (ECCV)*, Copenhagen, Denmark, 2002, pp. 414–431.

[25] S. Se *et al.*, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Int. J. Robot. Res.*, vol. 21, no. 8, pp. 735–758, Aug. 2002.

[26] R. Sim and G. Dudek, "Learning and evaluating visual features for pose estimation," in *Proc. 7th Int. Conf. Computer Vision (ICCV'99)*, Kerkyra, Greece, Sep. 1999, pp. 1217–1222.

[27] R. Smith *et al.*, "A stochastic map for uncertain spatial relationships," in *Proc. 4th Int. Symp. Robot. Res.*, 1987, pp. 467–474.

[28] J. D. Tardos *et al.*, "Robust mapping and localization in indoor environments using sonar data," *Int. J. Robot. Res.*, vol. 21, no. 4, pp. 311–330, Apr. 2002.

[29] S. Thrun *et al.*, "Minerva: A second-generation museum tour-guide robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA'99)*, Detroit, MI, May 1999, pp. 1999–2005.

[30] S. Thrun *et al.*, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, San Francisco, CA, Apr. 2000, pp. 321–328.

[31] B. Tordoff and D. W. Murray, "Guided sampling and consensus for motion estimation," in *Proc. Eur. Conf. Computer Vision (ECCV'02)*, Copenhagen, Denmark, May 2002, pp. 82–98.

[32] B. Triggs *et al.*, "Bundle adjustment–A modern synthesis," in *Vision Algorithms: Theory and Practice*, A. Zisserman and R. Szeliski, Eds: Springer-Verlag, 2000, LNCS 1883.

[33] S. B. Williams *et al.*, "An efficient approach to the simultaneous localization and mapping problem," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Washington, DC, May 2002, pp. 406–411.

**Stephen Se** (M'99) received the B.Eng. degree with first class honors in computing from the Imperial College of Science, Technology, and Medicine, University of London, London, U.K., in 1995 and the D.Phil. degree in the Robotics Research Group at the University of Oxford, Oxford, U.K., in 1999.

He is with the Research and Development Department at MD Robotics, Brampton, ON, Canada, developing computer vision systems for space and terrestrial applications. He worked from 1999 to 2001 as a Postdoctoral Researcher at the University of British Columbia, Vancouver, BC, Canada, on vision-based mobile robot localization and mapping. His research interests include computer vision, mobile robotics, localization, 3-D modeling, and artificial intelligence.

**David G. Lowe** (M'85) received the Ph.D. degree in computer science from Stanford University, Stanford, CA, in 1984.

From 1984 to 1987, he was an Assistant Professor at the Courant Institute of Mathematical Sciences at New York University. He is currently a Professor of Computer Science at the University of British Columbia, Vancouver, BC, Canada. His research interests include object recognition, local invariant features for image matching, robot localization, object-based motion tracking, and models of human visual recognition.

Dr. Lowe is on the Editorial Board of the *International Journal of Computer Vision* and is a member of the Scientific Advisory Board for Evolution Robotics.

**James J. Little** (M'80) received the Ph.D. degree from the University of British Columbia, Vancouver, BC, Canada, in 1985.

He is currently a Professor in the Department of Computer Science, University of British Columbia. He has been a Research Analyst at the Harvard Laboratory for Computer Graphics and Spatial Analysis (1972–1975), a Research Associate at Simon Fraser University (1975–1978), and a Research Scientist at the MIT Artificial Intelligence Laboratory (1985–1988). His research interests include early vision, understanding image sequences, surface representation, and visually guided mobile robotics. He is a member of the GEOIDE Research Management Committee.