Local and Global Localization for Mobile Robots using Visual Landmarks

Stephen Se, David Lowe, Jim Little
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1Z4, Canada
{se,lowe,little}@cs.ubc.ca

Abstract

Our mobile robot system uses scale-invariant visual landmarks to localize itself and build a 3D map of the environment simultaneously. As image features are not noise-free, we carry out error analysis and use Kalman Filters to track the 3D landmarks, resulting in a database map with landmark positional uncertainty. By matching a set of landmarks as a whole, our robot can localize itself globally based on the database containing landmarks of sufficient distinctiveness. Experiments show that recognition of position within a map without any prior estimate can be achieved using the scale-invariant landmarks.

1 Introduction

Mobile robot localization and mapping, the process of simultaneously tracking the position of a mobile robot relative to its environment and building a map of the environment, has been a central research topic for the past few years. Accurate localization is a prerequisite for building a good map, and having an accurate map is essential for good localization. Therefore, Simultaneous Localization And Map Building (SLAMB) is a critical underlying factor for successful mobile robot navigation in a large environment.

To achieve SLAMB, there are different types of sensor modalities such as sonar, laser range finders and vision. Many early successful approaches [4] utilize artificial landmarks, and therefore do not function properly in beacon-free environments. Vision-based approaches using stable natural landmarks in unmodified environments are highly desirable for a wide range of applications.

Harris's 3D vision system DROID [9] uses the visual motion of image corner features for 3D reconstruction. Kalman filters are used for tracking features from which it determines both the camera motion and the 3D positions of the features. It is accurate in the short to medium term, but long-term drifts can occur.

There are two types of localization: local and global. Local techniques aim at compensating odometric errors during robot navigation. They require that the initial location of the robot is approximately known and they typically cannot recover if they lose track of the robot's position.

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 kidnapped and carried to some unknown location. Global localization techniques are more powerful than local ones and can cope with situations in which the robot is likely to experience serious positioning errors.

Markov localization was employed by various teams with success [15, 19]. For example, the Deutsches Museum Bonn tour-guide robot RHINO [5] utilizes a metric version of this approach with laser sensors. However, it needs to be supplied with a manually derived map, and cannot learn maps from scratch. Using Markov localization, [6] proposes active localization, i.e., the localization routine can control the robot where to move and where to look, to increase the efficiency and robustness of localization.

Unlike RHINO, the latest museum tour-guide robot MINERVA [20] learns its map and uses camera mosaics of the ceiling in addition to the laser scan occupancy map. It uses the EM algorithm to learn the occupancy map and the Markov localization with filter techniques for global localization [8].

The Monte Carlo Localization method based on the CONDENSATION algorithm was proposed in [7]. This vision-based Bayesian filtering method uses a sampling-based density representation. Unlike the Kalman filter based approaches, it can represent multi-modal probability distributions. Given a visual map of the ceiling obtained by mosaicing, it localizes the robot globally using a scalar brightness measurement. [12] proposed some modifications to this algorithm for better efficiency in large symmetric environments.

Since the sensor information (sonar, laser or brightness measurement) only provides very low feature specificity, these methods are probabilistic and require the robot to move around, while the probabilities converge towards one localized peak gradually.

Sim and Dudek [18] proposed learning natural visual features for pose estimation. Landmark matching is achieved using principal components analysis and a tracked landmark is a set of image thumbnails detected in the learning phase, for each grid position in pose space. [3] selects image patches in terms of their uniqueness within the local region and dynamic reliability as landmarks for navigation.

We have proposed a vision-based SLAMB algorithm [17] by tracking SIFT (Scale Invariant Feature Transform) landmarks correcting odometry *locally*. As our robot is equipped with Triclops [14], a trinocular stereo system, the estimated 3D position of the landmarks can be obtained and hence a 3D map can be built and the robot can be localized simultaneously. The 3D map, represented as a SIFT landmark database, is incrementally updated over time and adaptive to dynamic environments.

The *kidnapped robot* problem 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, and therefore these visual landmarks are good for robot localization.

In this paper, we carry out error analysis for the SIFT landmarks as image features are not noise-free. By associating each landmark with a covariance matrix, we employ the Kalman Filters to track the landmarks, resulting in a database with landmark positional uncertainty. By enhancing the specificity of the SIFT features, we consider matching a set of SIFT landmarks as a whole, using the highly distinctive visual information to localize the robot *globally* from the current view of the scene.

2 SIFT Features

SIFT was developed by Lowe [13] for image feature generation in object recognition applications. The features are invariant to image translation, scaling, rotation, and partially invariant to illumination changes and affine or 3D projection. These characteristics make them suitable landmarks for robust SLAMB, since when mobile robots are moving around in an environment, landmarks are observed over time, but from different angles, distances or under different illumination.

Previous approaches to feature detection, such as the widely used Harris corner detector [10], 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.

2.1 Feature Generation

Key locations are selected at maxima and minima of a difference of Gaussian function applied in scale space. This is computed by forming a pyramid of images using Gaussian smoothing and subsampling, with adjacent scales differing by a factor of $\sqrt{2}$.

Prior to subsampling, each image is subtracted from its Gaussian smoothed image to produce a difference-of-Gaussian image for each pyramid level. Feature locations are identified by detecting maxima and minima relative to surrounding pixels and adjacent scales. This provides an efficient method for identifying repeatable locations in scale space.

SIFT locates key points at regions and scales of high variation, making these locations particularly stable for characterizing the image. [13] demonstrated the stability of SIFT keys to image transformations. At each feature location, an orientation is selected by determining the peak of a histogram of local image gradient orientations. A subpixel image location, scale and orientation are associated with each SIFT feature.

2.2 Stereo Matching

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 similarly, with an extra constraint for agreement between the horizontal and vertical disparities. The horizontal disparity is the difference in row pixel position in the left and right cameras whereas the vertical disparity is the difference in column pixel position in the top and right cameras.

If a feature has more than one match satisfying these criteria, it is ambiguous and discarded so that the resulting matches are more consistent and reliable. The final disparity is taken as the average between the horizontal and vertical disparities.

From the positions of the matches and knowing the camera intrinsic parameters, we can compute the 3D world coordinates relative to the robot for each feature. They can subsequently serve as landmarks for map building and tracking.

2.3 Ego-motion Estimation

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 data can only give a rough estimate and it is prone to error such as drifting, slipping, etc. 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 camera ego-motion and hence better localization. This will also help adjust the 3D coordinates of the SIFT landmarks for map building.

We would like to maintain a database tracking the SIFT landmarks and use it to match features found in subsequent views. The initial coordinate frame is used as the reference and all landmarks are relative to this fixed frame.

Figure 1 show the SIFT detection, stereo matching and frame-to-frame matching for some typical scene. Readers are referred to [17] for further details.

3 Landmark Uncertainty

There are various errors such as noisy sensors and quantization associated with the images and the SIFT features found. They introduce inaccuracy in both the landmarks' position as well as the least-squares estimation of the robot position. We would like to know how reliable the estimates are, so we incorporate covariance into the SIFT database.

We employ a Kalman Filter [1] to update the position of each landmark. A 3x3 covariance matrix is associated with each SIFT landmark in the database. When a match is found in the current frame, the current covariance matrix for the landmark will be combined with the covariance matrix in the database so far, and the 3D position will be updated accordingly.

3.1 Image Coordinates Uncertainty

Uncertainty of the image coordinates and disparity obtained during the SIFT feature detection and matching will be propagated to uncertainty in the landmark 3D positions.

For stereo under a typical pinhole camera model, we can compute the the 3D location (X, Y, Z) of a landmark from its row and column image coordinates (r, c) and its disparity d:

$$X = \frac{(c - c_0)I}{d}; \qquad Y = \frac{(r_0 - r)I}{d}; \qquad Z = \frac{fI}{d}$$

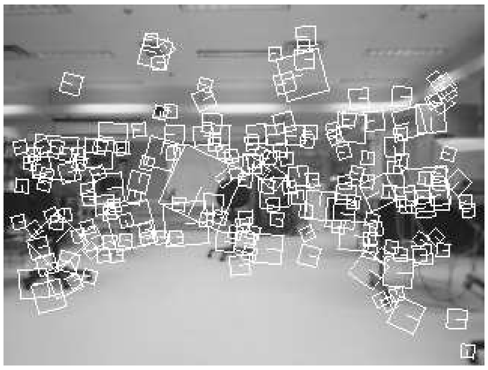
where (r_0, c_0) are the image centre coordinates, I is the interocular distance and f is the focal length.

For the first order error propagation [2], we have:

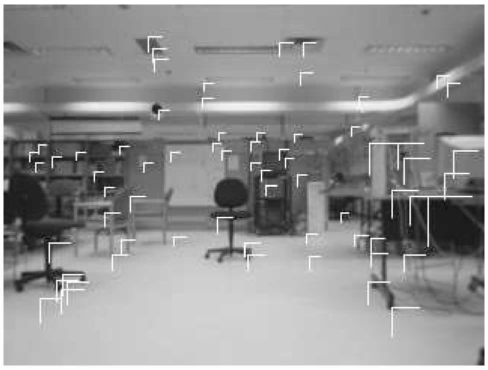
$$\sigma_X^2 = \frac{I^2\sigma_c^2}{d^2} + \frac{I^2(c - c_0)^2\sigma_d^2}{d^4}$$

$$\sigma_Y^2 = \frac{I^2\sigma_r^2}{d^2} + \frac{I^2(r_0 - r)^2\sigma_d^2}{d^4}$$

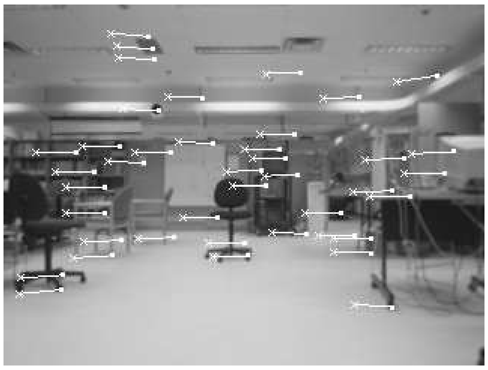
$$\sigma_Z^2 = \frac{f^2I^2\sigma_d^2}{d^4}$$



(a)



(b)



(c)

Figure 1: *(a) SIFT features found, with scale and orientation indicated by the size and orientation of the squares. (b) Stereo matching result, where horizontal and vertical lines indicate the horizontal and vertical disparities respectively. (c) The SIFT feature matches between consecutive frames for a 5° clockwise rotation, where the white dot indicates the current position and the white cross indicates the new position.*

where σ_X^2, σ_Y^2, σ_Z^2, σ_c^2, σ_r^2 and σ_d^2 are the variances of X, Y, Z, c, r and d respectively.

Experiments show that both the row and column coordinates are roughly normally distributed with $\sigma_r^2 = 1$ and $\sigma_c^2 = 1$ (i.e., standard deviation of the feature location is about 1 pixel).

We compute the horizontal disparity *hd* as the difference between two column coordinates, and the vertical disparity *vd* as the difference between two row coordinates. Therefore, the variances for *hd* and *vd* are both 2. Then, the final disparity is obtained as the average between *hd* and *vd*, so the variance for the final disparity is one quarter of the sum of the variances for *hd* and *vd*, hence $\sigma_d^2 = 1$.

Knowing the intrinsic parameters of our system, we can compute the variances for the landmark positions according to the error propagation formulae above.

3.2 Landmark Matching

Based on the robot odometry, we predict the database landmark positions in the new frame and look for corresponding matches in the vicinity. We then verify the matches according to the SIFT scale and orientation information. This deals with the data association problem and allows us to identify which filters to be updated.

A landmark will be discarded from the database if it has not been matched for more than a predefined number of consecutive frames, otherwise, it will be maintained to cater for temporary occlusion.

3.3 Backward Transformation

We use the robot pose estimate to transform landmarks in the current coordinates frame into the initial frame. From the least-squares minimization procedure, we can obtain the robot pose as well as its covariance, which needs to be propagated to the landmark 3D position uncertainty.

To transform from current frame to the initial frame:

$$\mathbf{r}_{new} = (\mathbf{P}_\theta(\mathbf{P}_\alpha(\mathbf{P}_\beta \mathbf{r}_{obs}))) + \mathbf{V}$$

where $\mathbf{r}_{obs}$ and $\mathbf{r}_{new}$ are the observed position in the current frame and the transformed position in the initial frame respectively. $\mathbf{V}$ is the translational transformation while $\mathbf{P}_\theta$, $\mathbf{P}_\alpha$ and $\mathbf{P}_\beta$ are the rotational transformations required (for yaw θ, pitch α and roll β) around each of the three axes.

We would like to obtain the covariance of $\mathbf{r}_{new}$ (Σ_{new}) from the covariance of the observed position Σ_{obs}, given by a diagonal matrix consisting of σ_X^2, σ_Y^2 and σ_Z^2.

3.4 Error Propagation

Assuming the roll, pitch and yaw components are independent, the transformation proceeds in 4 stages: $\mathbf{P}_\beta$ (roll), $\mathbf{P}_\alpha$ (pitch), $\mathbf{P}_\theta$ (yaw) and then $\mathbf{V}$ (translations). We obtain the variances (σ_β^2,σ_α^2,σ_θ^2) for these parameters from the robot position covariance.

In general, given

$$\mathbf{X}' = \mathbf{P}\,\mathbf{X}$$

where $\mathbf{P}$ is a 3x3 matrix, $\mathbf{X}$ and $\mathbf{X}'$ are the 3-vectors for the old and new position respectively. When there are errors associated with both $\mathbf{P}$ and $\mathbf{X}$: $\Lambda_\mathbf{P}$ (9x9 covariance for $\mathbf{P}$) and $\Lambda_\mathbf{X}$ (3x3 covariance for $\mathbf{X}$), the 3x3 covariance for the resulting vector $\mathbf{X}'$, based on first order uncertainty analysis, is given by:

$$\left[\begin{array}{ccc|c} \mathbf{X}^\top & \mathbf{0} & \mathbf{0} & \\ \mathbf{0} & \mathbf{X}^\top & \mathbf{0} & \mathbf{P} \\ \mathbf{0} & \mathbf{0} & \mathbf{X}^\top & \end{array}\right] \left[\begin{array}{cc} \Lambda_\mathbf{P} & \mathbf{0} \\ \mathbf{0} & \Lambda_\mathbf{X} \end{array}\right] \left[\begin{array}{ccc} \mathbf{X} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{X} \\ \hline & \mathbf{P}^\top & \end{array}\right]$$

This covariance matrix is the product of three matrices: the first matrix is a 3x12 matrix, the second matrix is a 12x12 matrix and the third matrix is the transpose of the first matrix (hence a 12x3 matrix).

3.5 Filter Update

Afterwards, we combine the new covariance matrix Σ_{new} with the previous covariance matrix of the landmark in the database Σ_{KF} to obtain the new covariance matrix Σ'_{KF}. We combine the new position of the landmark $\mathbf{r}_{new}$ with the database landmark position $\mathbf{s}_{KF}$ using the covariances to obtain a better estimate of its new position $\mathbf{s}'_{KF}$. We have:

$$\Sigma'_{KF} = (\Sigma_{KF}^{-1} + \Sigma_{new}^{-1})^{-1}$$

$$\mathbf{s}'_{KF} = \Sigma'_{KF}(\Sigma_{KF}^{-1}\mathbf{s}_{KF} + \Sigma_{new}^{-1}\mathbf{r}_{new})$$

3.6 Database Map With Uncertainty

During the map building process, the robot traverses around our 10m by 10m lab tracking the SIFT landmarks. A Kalman Filter is initiated for each landmark and updated over frames. From the bird's eye view of the map, we can see that its uncertainty ellipse shrinks when a landmark is being observed repeatedly, while its positional uncertainty decreases.

Figure 2 shows the bird's eye view of the SIFT database as well as the robot trajectory after 148 frames with 4828 landmarks in the database. The landmarks are three-dimensional and their uncertainty are represented as ellipsoids, but ellipses are shown in the bird's eye view. Assuming errors are normally distributed, each landmark position has a χ^2 distribution with 3 degrees of freedom. Error ellipses covering a region of 1 standard deviation in either sides of X and Z directions are shown.

Uncertainty for landmarks closer to the robot tends to be lower, as expected for landmarks with larger disparities. The largest uncertainty is associated with landmarks that are seen only from a distance, such as those seen through an open door in the upper right corner of Figure 2. Visual judgement indicates that the SIFT landmarks correspond well to actual objects in the lab.

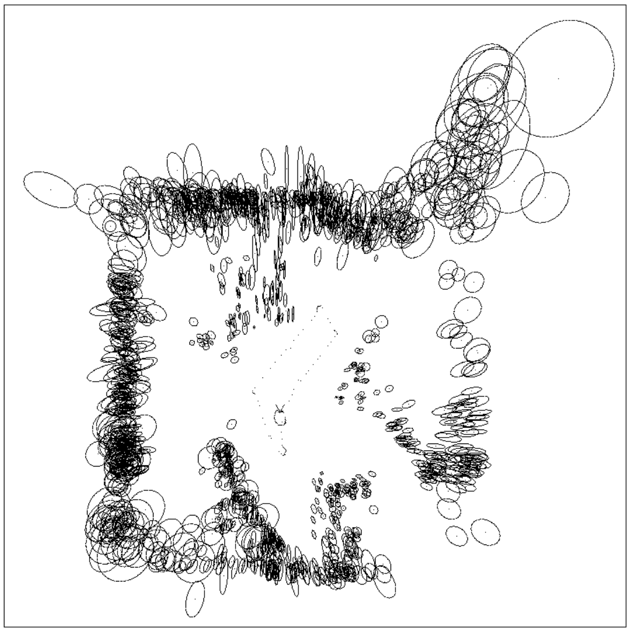


Figure 2: *Bird's eye view of the 3D SIFT database map, showing the uncertainty ellipses of the landmarks, and the robot trajectory during the map building. Note that the smallest ellipses represent the most reliable and useful landmarks.*

As the disparity error transforms to a larger error in the depth direction, we can see that for most landmarks, the uncertainty ellipses are elongated in the direction along which they are observed. For example, the robot was facing rightward in the X direction, when the landmarks on the right hand side are observed, therefore the ellipses are elongated in the X direction.

4 Local Image Characteristics

So far, we have been using the scale and orientation of each SIFT key for localization and map building. In order to recognize where the robot is relative to a previously built map, features sufficiently distinctive are required to identify scenes in the map. In order to obtain a feature vector for high specificity, we describe the local image region in a manner invariant to various image transformations [13].

This feature vector is formed by measuring the local image gradients at a number of orientations in coordinates relative to the location, scale and orientation of the feature. The gradient locations are further blurred to reduce sensitivity to small local image deformations, such as result from 3D viewpoint change.

The local and multi-scale nature of the features makes them insensitive to noise, clutter and occlusion, while the detailed local image properties represented by the features makes them highly selective for matching to large databases of previously viewed features.

Lowe's object recognition application used 8 orientations, each sampled over 4x4 grid of locations, so the total number of samples for each SIFT key is 128. For our application, we experimentally compare different sample sizes and it seems that a smaller vector is already sufficiently discriminating in our environment. We use 4 orientations, each sampled over a 2x2 grid of locations. 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 or not they match.

Stereo matching and frame-to-frame matching is still based on the scale and orientation, as very consistent results can be obtained already, to avoid extra computational burden.

5 Hough Transform Set Matching

During stereo and frame-to-frame matching, we only consider each SIFT feature on its own. To tackle global localization, we consider matching a set of SIFT landmarks as a whole. Given a small set of current SIFT features and a large set of SIFT landmarks in the database, we would like to estimate the robot position that would have brought the largest number of landmarks into close alignment, provided that the robot has previously viewed the current scene during the map building stage.

Hough Transform [11] hashing methods are known to be very efficient for this type of task. We start with a three-dimensional discretized search space (X, Z, θ) where X is the sideways displacement, Z is the forward displacement and θ is the orientation. The algorithm is as follows:

1. For each SIFT feature in the current frame, find the set of potential SIFT landmarks in the database that match, using the local image vector and the height as the preliminary constraints.
2. For each of the potential matches, for all discretized values of θ within the search space, compute the X and Z required to match. Vote for the Hough Transform bin that corresponds to the particular X and Z.
3. Due to landmark uncertainty described above, we also vote the neighbouring Hough bins within $\pm 2.8\sigma_X$ and $\pm 2.8\sigma_Z$, i.e., covering a 95% confidence region. Although the uncertainty is elliptical in shape for the X and Z directions, we simply vote in the rectangular region covering the ellipse.
4. Afterwards, the bins with many votes correspond to the pose configurations that are more likely to

result in a large number of matches.

5. Select the top N pose configurations and carry out feature matching for each pose. Use the matches for least-squares minimization to obtain a pose estimate. Then, for each of the N pose estimates, iterate using least-squares minimization again to obtain a better pose estimate. Selecting a small set of hypotheses in a large space for verification allows for noisy data and uncertainty.

6. Among the N hypothesis, we select the one with the maximum number of matches and the lowest least-squares error, corresponding to a robot pose which can best match the most landmarks in the database.

6 Global Localization Results

Using the SIFT database map built in Section 3.6, the robot is carried to various positions and asked to estimate where it is. We measure manually the approximate location and orientation of the robot and compare with the SIFT global localization results (with N set to 10) for various test positions.

The results are tabulated as follows, where (X,Z,H) represents X cm in the X direction, Z cm in the Z direction and H degree orientation:

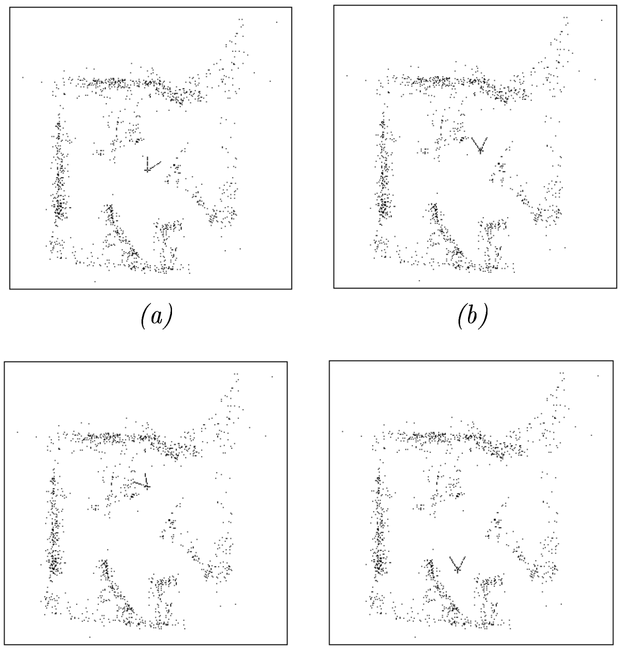
Case	Measured Pose	Pose Estimation
1	(45,90,25°)	(50.953,92.034,26.963°)
2	(90,180,−5°)	(93.670,187.963,−4.278°)
3	(70,300,−40°)	(75.800,295.850,−41.074°)
4	(0,-150,0°)	(-2.733,-142.823,−1.226°)
5	(0,0,0°)	(0.792,1.746,−0.426°)
6	(0,0,15°)	(0.886,-4.071,16.035°)
7	(-100,10,−90°)	(-93.816,4.357,−87.867°)
8	(-10,-70,130°)	(-9.602,-74.298,128.918°)

From these results, we can see that global localization using SIFT features augmented with local image description is good in all cases. Unlike Markov localization, the robot does not need to traverse around to localize, thanks to the specificity of the SIFT features.

The results are independent estimates obtained from different positions and viewpoints. The average translational and rotational errors are 6.08cm and 1.21° respectively for this set of experiments. All estimation results are within 10cm of the ground truth.

Figure 3 shows the localization results visually for cases 1 to 4, indicating the robot's estimated position and orientation relative to the environment.

The Hough bin size does not seem to affect the results very much - currently we use 4cm discretization in the X and Z directions and 2° discretization for the orientation. Only a rough pose is required from the Hough Transform stage, since we then proceed with



(a) (b) (c) (d)

Figure 3: *Robot global localization results showing the estimated position and orientation. The vee indicates the robot field of view. (a) Case 1. (b) Case 2. (c) Case 3. (d) Case 4.*

least-squares minimization to converge to a better estimate. The bin size does affect the run-time performance. It takes around 2.5 seconds on a Pentium III 700MHZ processor for the localization here, with the Hough space covering the lab size and all directions. But this is a bootstrap phase which only needs to be carried out once in the beginning, since after knowing where it is, the robot will continue the usual concurrent localization and map building.

7 Conclusion

In this paper, we described briefly our vision-based SLAMB algorithm based on the SIFT features. Being scale and orientation invariant, SIFT features are good natural visual landmarks for tracking over long periods of time from different views, to correct odometry locally. As there are errors associated with image features, error analysis is important to tell us how well the landmarks are localized.

Beyond previous works, we have achieved mobile robot global localization based on distinctive natural visual landmarks in the environment, i.e., our robot can localize itself without any prior knowledge about its position. Through experiments, we have demonstrated that using SIFT features provides promising results for both local and global localization.

Maps can now be re-used as the robot knows where it is, it can continue to improve and augment the previous map. Using the same database map, multiple robots can localize themselves individually with reference to the same coordinates frame based on the visual landmarks they are looking at. Knowing the relative positions of the robots from each other is crucial for multi-robot collaboration, such as navigation, map building or other higher-level tasks. Unlike [16] where the robots need to be within the field of view of one another to be localized, our approach allows the robots to be at different regions in the map.

A comprehensive database map is important for global localization. For example, if the robot has only observed the front of an object, then it will not be able to recognize the back of the object, as they are completely different landmarks. We are currently investigating some mobile robot exploration strategies to build a good map for the environment, where the robot would observe objects from various viewpoints.

When the global localization at some position is not certain due to the lack of features or other reasons, the robot should spin around until it is confident of the estimate. Moreover, experiments with larger and more complex environments are necessary to evaluate the localization further, when the database map size increases. The SIFT landmark specificity can be increased if necessary to ensure enough distinctiveness is maintained.

Acknowledgements

Our work has been supported by the Institute for Robotics and Intelligent System (IRIS III), a Canadian Network of Centres of Excellence.

References

[1] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association.* Academic Press, Boston, 1988.

[2] P.R. Bevington and D.K. Robinson. *Data Reduction and Error Analysis for the Physical Sciences.* McGraw-Hill, second edition, 1992.

[3] G. Bianco and A. Zelinsky. Biologically inspired visual landmark learning and navigation for mobile robots. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99)*, pages 671–676, Korea, October 1999.

[4] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques.* A. K. Peters, Ltd, Wellesley, MA, 1996.

[5] W. Burgard, A.B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI)*, Madison, Wisconsin, July 1998.

[6] W. Burgard, D. Fox, and S. Thrun. Active mobile robot localization. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Nagoya, Japan, August 1997.

[7] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99)*, Fort Collins, CO, June 1999.

[8] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, November 1999.

[9] C. Harris. Geometry from visual motion. In A. Blake and A. Yuille, editors, *Active Vision*, pages 264–284. MIT Press, 1992.

[10] C.J. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of 4th Alvey Vision Conference*, pages 147–151, Manchester, 1988.

[11] P.V.C. Hough. Method and means of recognizing complex patterns, December 1962. U.S. Patent 306965418.

[12] P. Jensfelt, O. Wijk, D.J. Austin, and M. Andersson. Experiments on augmenting condensation for mobile robot localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, April 2000.

[13] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, pages 1150–1157, Kerkyra, Greece, September 1999.

[14] D. Murray and J. Little. Using real-time stereo vision for mobile robot navigation. In *Proceedings of the IEEE Workshop on Perception for Mobile Agents*, Santa Barbara, CA, June 1998.

[15] I. Nourbakhsh, R. Powers, and S. Birchfield. Dervish: An office-navigating robot. *AI Magazine*, 16:53–60, 1995.

[16] I. Rekleitis, G. Dudek, and E. Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1340–1345, Japan, August 1997.

[17] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2051–2058, Seoul, Korea, May 2001.

[18] R. Sim and G. Dudek. Learning and evaluating visual features for pose estimation. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, Kerkyra, Greece, September 1999.

[19] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1080–1087, San Mateo, CA, 1995. Morgan Kaufmann.

[20] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Minerva: A second-generation museum tour-guide robot. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'99)*, Detroit, Michigan, May 1999.