Stephen Se

MD Robotics
9445 Airport Road
Brampton, Ontario L6S 4J3, Canada
sse@mdrobotics.ca

David Lowe
Jim Little

Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1Z4, Canada
lowe@cs.ubc.ca
little@cs.ubc.ca

Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks

Abstract

A key component of a mobile robot system is the ability to localize itself accurately and, simultaneously, to build a map of the environment. Most of the existing algorithms are based on laser range finders, sonar sensors or artificial landmarks. In this paper, we describe a vision-based mobile robot localization and mapping algorithm, which uses scale-invariant image features as natural landmarks in unmodified environments. The invariance of these features to image translation, scaling and rotation makes them suitable landmarks for mobile robot localization and map building. With our Triclops stereo vision system, these landmarks are localized and robot ego-motion is estimated by least-squares minimization of the matched landmarks. Feature viewpoint variation and occlusion are taken into account by maintaining a view direction for each landmark. Experiments show that these visual landmarks are robustly matched, robot pose is estimated and a consistent three-dimensional map is built. As image features are not noise-free, we carry out error analysis for the landmark positions and the robot pose. We use Kalman filters to track these landmarks in a dynamic environment, resulting in a database map with landmark positional uncertainty.

KEY WORDS—localization, mapping, visual landmarks, mobile robot

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 in mobile robotics. 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 (SLAM) is a critical underlying factor for successful mobile robot navigation in a large environment, irrespective of what the high-level goals or applications are.

To achieve SLAM, there are different types of sensor modalities such as sonar, laser range finders and vision. Sonar is fast and cheap but usually very crude, whereas a laser scanning system is active, accurate but slow. Vision systems are passive and of high resolution. Many early successful approaches (Borenstein et al. 1996) utilize artificial landmarks, such as bar-code reflectors, ultrasonic beacons, visual patterns, etc., and therefore do not function properly in beacon-free environments. Therefore, vision-based approaches using stable natural landmarks in unmodified environments are highly desirable for a wide range of applications. The map built from these natural landmarks will serve as the basis for performing high-level tasks such as mobile robot navigation.

1.1. Literature Review

Harris's three-dimensional (3D) vision system DROID (Harris 1992) uses the visual motion of image corner features for 3D reconstruction. Kalman filters are used for tracking features, and from the locations of the tracked image features, DROID determines both the camera motion and the 3D positions of the features. Ego-motion determination by matching image features is generally very accurate in the short to medium term. However, in a long image sequence, long-term drifts can occur as no map is created. In the DROID system where monocular image sequences are used without

The International Journal of Robotics Research
Vol. 21, No. 8, August 2002, pp. 735-758,

odometry, the ego-motion and the perceived 3D structure can be self-consistently in error. It is an incremental algorithm and runs at near real-time.

Thrun et al. (1998) proposed a probabilistic approach using the Expectation–Maximization (EM) algorithm. The E-step estimates robot locations at various points based on the currently best available map and the M-step estimates a maximum likelihood map based on the locations computed in the E-step. The EM algorithm searches for the most likely map by simultaneously considering the locations of all past sonar scans. Being a batch algorithm, it is not incremental and cannot be run in real-time.

Thrun et al. (2000) proposed a real-time algorithm combining the strengths of EM algorithms and incremental algorithms. Their approach computes the full posterior probability over robot poses to determine the most likely pose, instead of just using the most recent laser scan as in incremental mapping. The mapping is achieved in two dimensions using a forward-looking laser, and an upward-pointed laser is used to build a 3D map of the environment. However, it does not scale to large environments as the calculation cost of the posterior probability is too expensive.

The Monte Carlo localization method was proposed in Dellaert et al. (1999) based on the CONDENSATION algorithm. This vision-based Bayesian filtering method uses a sampling-based density representation and can represent multi-modal probability distributions. Given a visual map of the ceiling obtained by mosaicing, it localizes the robot using a scalar brightness measurement. Jensfelt et al. (2000) proposed some modifications to this algorithm for better efficiency in large symmetric environments. CONDENSATION is not suitable for SLAM due to scaling problems and hence it is only used for localization.

In SLAM, as the robot pose is being tracked continuously, multi-modal representations are not needed. Grid-based representation is problematic for SLAM because maintaining all grid positions over an entire region is expensive and grids are difficult to match.

Using global registration and correlation techniques, Gutmann and Konolige (1999) proposed a method to reconstruct consistent global maps from laser range data reliably. Their pose estimation is achieved by scan matching of dense two-dimensional (2D) data and is not applicable to sparse 3D data from vision.

Sim and Dudek (1999) proposed learning natural visual features for pose estimation. Landmark matching is achieved using principal components analysis. A tracked landmark is a set of image thumbnails detected in the learning phase, for each grid position in pose space. It does not build any map for the environment.

In SLAM, a robot starts at an unknown location with no knowledge of landmark positions. From landmark observations, it simultaneously estimates its location and landmark locations. The robot then builds up a complete map of landmarks which are used for robot localization. In stochastic mapping (Smith et al. 1987), a single filter is used to maintain estimates of robot position, landmark positions and the covariances between them.

Many existing systems (Leonard and Durrant-Whyte 1991; Castellanos et al. 1999; Williams et al. 2000) are based on this framework but the computational complexity of stochastic mapping is $\mathbf{O}(n^2)$ and hence increases greatly with the map size.

Various approaches have been developed to reduce this complexity problem. Sub-optimal methods can provide speedier filtering by neglecting some of the coupling in the landmarks (Castellanos et al. 2000). Decoupled stochastic mapping reduces this computational burden by dividing the environment into multiple overlapping submap regions, each with its own stochastic map (Leonard and Feder 1999).

The postponement technique (Davison 1998; Knight et al. 2001) is an optimal method which updates a constant-sized data set based on current measurements and carries out updates on all unobserved parts of the map at a later stage. Essentially, it gathers all the changes that would need to be made at each step, and then carries out an expensive full map update occasionally.

The compressed filter proposed by Guivant and Nebot (2001) does not affect the optimality of the system while it significantly reduces the computation requirements when working in local areas. It only maintains the information gained in a local area which is transferred to the overall map in one iteration at full SLAM computation cost.

Most of the existing mobile robot localization and mapping algorithms are based on laser or sonar sensors, as vision is more processor intensive and good visual features are more difficult to extract and match. Existing vision-based approaches use low-level features such as vertical edges (Castellanos et al. 1999) and have complex data association problems. Our approach uses high-level image features which are scale invariant, thus greatly facilitating feature correspondence. Moreover, these features are distinctive and therefore their maps allow efficient algorithms to tackle the “kidnapped robot” problem (Se et al. 2001a).

1.2. Paper Structure

In this paper, we propose a vision-based SLAM algorithm by tracking the scale-invariant feature transform (SIFT) visual landmarks in unmodified environments (Se et al 2001b). As our robot is equipped with the Triclops[1], a trinocular stereo system, 3D positions of the landmarks can be obtained. Hence, a 3D map can be built and the robot can be localized simultaneously in three dimensions. The 3D map, represented as a SIFT feature database, is constantly updated over frames and is adaptive to dynamic environments.

[1] www.ptgrey.com

In Section 2 we explain the SIFT features and the stereo matching process. Ego-motion estimation by matching features across frames is described in Section 3. SIFT database landmark tracking is presented in Section 4 with experimental results shown in Section 5, where our 10 × 10 m^2 laboratory environment is mapped with thousands of SIFT landmarks. In Section 6 we describe some enhancements to the SIFT database. Error analysis for both the robot position and the landmark positions is carried out in Section 7, resulting in a SIFT database map with landmark uncertainty. Finally, we conclude and discuss some future work in Section 8.

2. SIFT Stereo

SIFT was developed by Lowe (1999) 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 SLAM because 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 (Harris and Stephens 1988), 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.

At each frame, we extract SIFT features in each of the three images and stereo match them among the images. Matched SIFT features are stable and will serve better as landmarks for the environment to be tracked over time. Moreover, stereo matched features provide their 3D world positions.

2.1. Generating SIFT Features

The SIFT feature locations are determined by identifying repeatable points in a pyramid of scaled images. This is computed by first smoothing the image with a Gaussian kernel with a sigma of $\sqrt{2}$. The smoothed image is subtracted from the original image to produce a difference-of-Gaussian image. The smoothed image is then resampled with a pixel spacing 1.5 times larger to produce the next level of the image pyramid. The operations are repeated at decreasing scales until the image size is too small for feature detection. This is a particularly efficient scale-space structure, as the operations of smoothing, subtraction, and subsampling can all be performed with a few dozen operations per pixel.

Feature locations are identified by detecting maxima and minima in the difference-of-Gaussian pyramid. This is efficiently implemented by comparing each pixel to its surrounding pixels and those at adjacent scales. A change in scale of the original image will produce a corresponding change in the scale at which the critical point is detected.

The difference-of-Gaussian function is circularly symmetric, so feature locations are invariant to changes in image orientation. The SIFT features then assign a canonical orientation at each location so that descriptions relative to this orientation will remain constant following image rotation. The orientation is selected by determining the peak in a histogram of the local image gradient orientations sampled over a Gaussian-weighted circular region around the point.

Figure 1 shows the SIFT features that were found for the top, left, and right images taken with our Triclops cameras. A subpixel location, scale and orientation are associated with each SIFT feature. The scale and orientation of each feature is indicated by the size and orientation of the corresponding square. The image resolution is 320 × 240 and eight levels of scale were used. There were about 180 features found in each image, which was sufficient for this task, but if desired, the number could be increased by processing all scales and using full image resolution.

2.2. Stereo Matching

In the Triclops system, the right camera serves as the reference camera, as the left camera is 10 cm beside it and the top camera is 10 cm directly above it. We will first match the SIFT features in the right and left images and then refine the resulting matches using the top image.

2.2.1. Stage I: Right to Left Match

For a SIFT feature in the right image and a SIFT feature in the left image to match, the following criteria should be satisfied:

Epipolar constraint. The vertical image coordinates must be within 1 pixel of each other, as the images have been aligned and rectified.

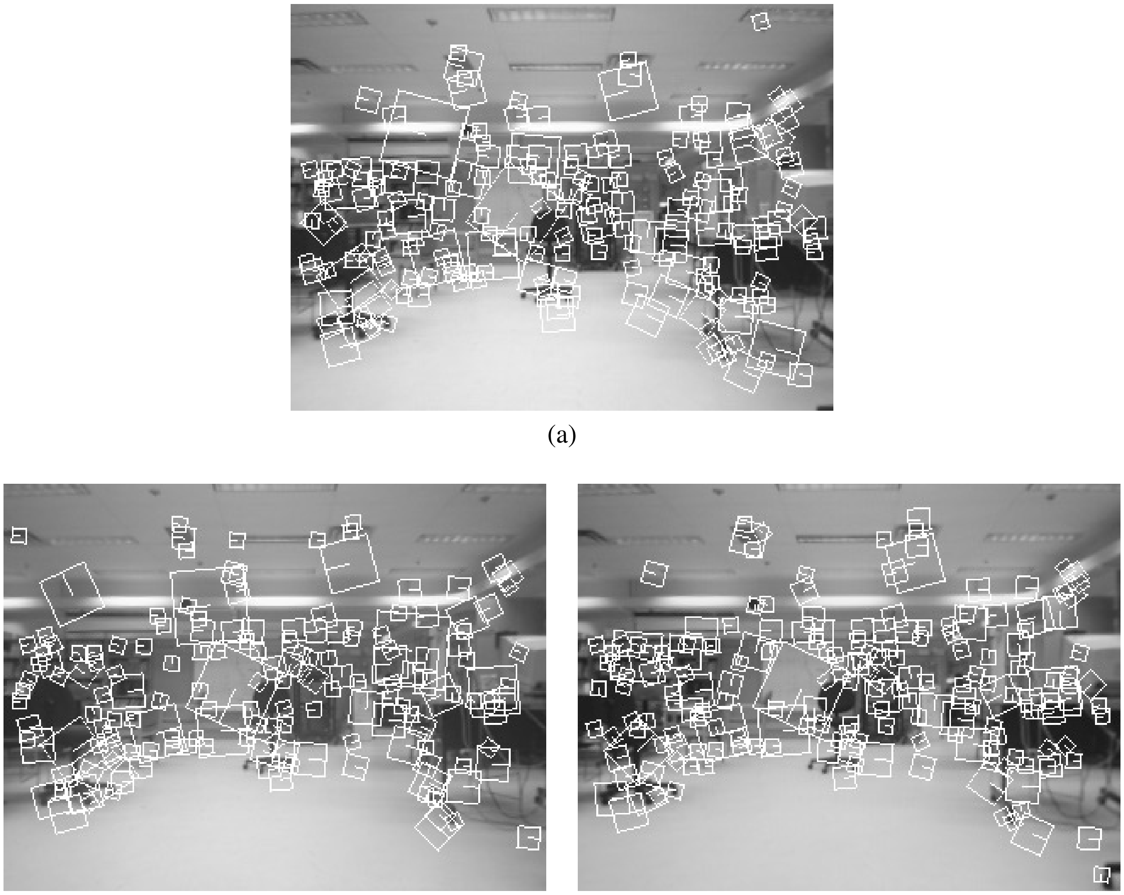
Disparity constraint. The horizontal image coordinates of the left image must be greater than those of the right image and the difference must be within some predefined disparity range (currently 20 pixels).

Orientation constraint. The difference of the two orientations must be within 20 degrees.

Scale constraint. One scale must be at most one level higher or lower than the other. Adjacent scales differ by a factor of 1.5 in our SIFT implementation.

Unique match constraint. If a feature has more than one match satisfying the above criteria, the match is ambiguous and discarded so that the resulting matches are more consistent and reliable.

After matching the SIFT features, we retain a subset of the right image SIFT features which match some SIFT features in the left image. These matches allow us to compute the



(a)

(b)

(c)

Fig. 1. SIFT features found, with scale and orientation indicated by the size and orientation of the squares: (a) top image; (b) left image; (c) right image.

subpixel horizontal disparity for each matched feature in this subset.

2.2.2. *Stage II: Right to Top Match*

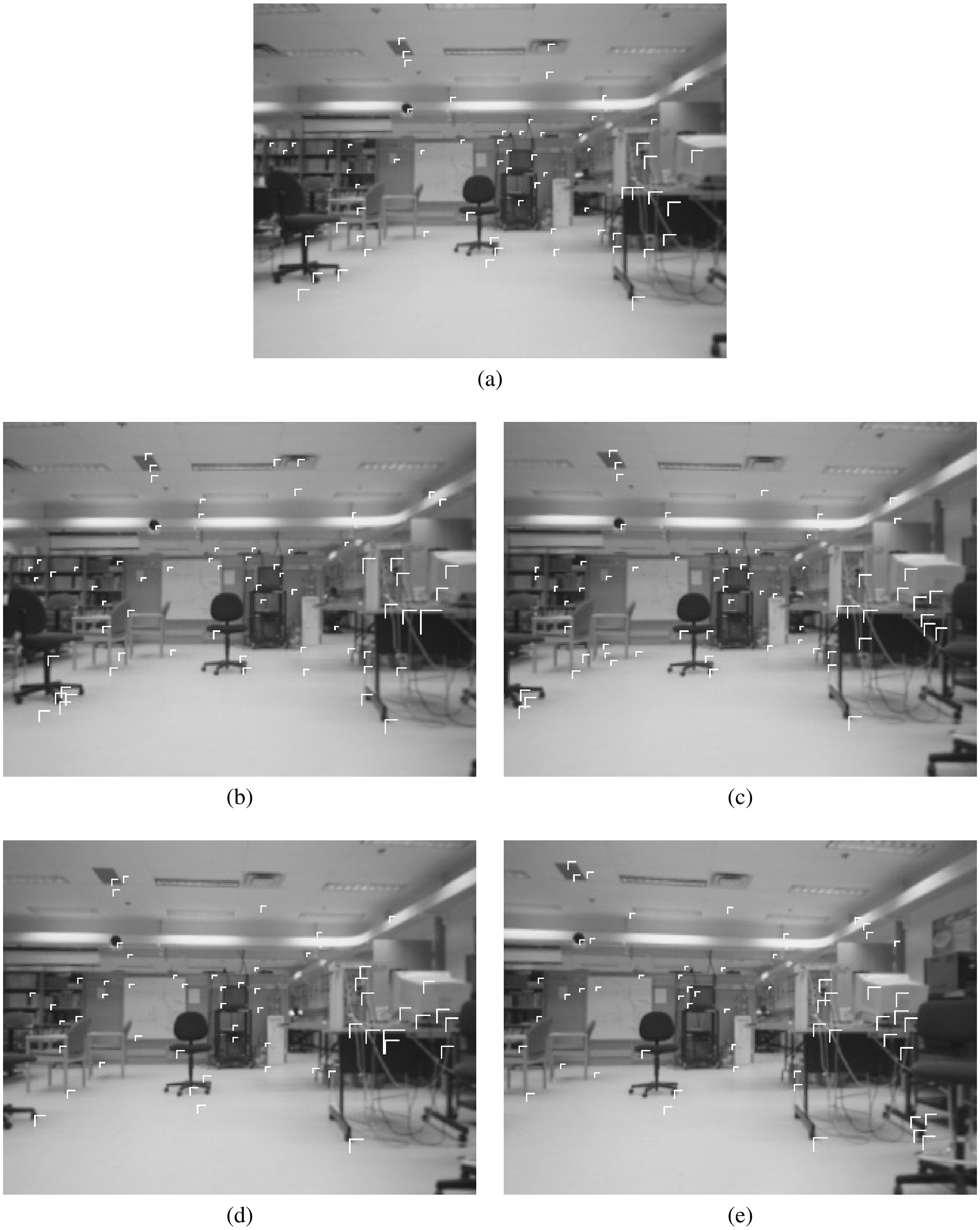
For the next stage, we use the top image to refine this intermediate subset. The criteria to be satisfied are similar to those in stage I and we obtain a subset of this intermediate subset. The resulting matches allow us to compute the subpixel vertical disparity. An additional constraint is employed here to refine this final set: the horizontal disparity and the vertical disparity of each match must be within 1 pixel of one another.

The orientation and scale of each matched SIFT feature are taken as the average of the orientation and scale among the corresponding SIFT feature in the left, right and top images. The disparity is taken as the average of the horizontal disparity and the vertical disparity. Together with the positions of the features and the camera intrinsic parameters, we can compute the 3D world coordinates (X, Y, Z) relative to the robot for each feature in this final set.[2] They can subsequently serve as landmarks for map building and tracking.

2.3. Results

For the three images shown in Figure 1, stereo matching is carried out on the SIFT features. After stage I matching between the right and left images, the number of resulting matches is 106. After stage II matching with the top image, the final number of matches is 59. The result is shown in Figure 2(a), where each matched SIFT feature is marked. The length of the horizontal line indicates the horizontal disparity and the vertical line indicates the vertical disparity for each feature. Figures 2(b), (c), (d) and (e) show more SIFT stereo results for slightly different views when the robot makes some small translation and rotation. There are around 60 final matches in each view.

[2] Alternatively, 3D positions can be obtained by minimizing the intersection errors of the three rays for the right, left and top images.



(a)

(b)

(c)

(d)

(e)

Fig. 2. Stereo matching results for different views from a moving robot. The horizontal line indicates the horizontal disparity and the vertical line indicates the vertical disparity. Closer objects will have larger disparities. Tracking results are shown in Figure 3.

Slightly different values for the various constraints have been tested and their effect on the stereo results is very small. The matches are stable with respect to the constraint parameters. Relaxing some of the constraints above does not necessarily increase the number of final matches because some SIFT features will then have multiple potential matches and therefore be discarded.

3. Ego-motion Estimation

After SIFT stereo matching, we obtain

$$[r_m, c_m, s, o, d, X, Y, Z]$$

for each matched SIFT feature, where (r_m, c_m) are the measured image coordinates in the reference camera, (s, o, d) are the scale, orientation and disparity associated with each feature, and (X, Y, Z) are the 3D coordinates of the landmark relative to the camera.

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 (Borenstein and Feng 1996) only gives a rough estimate and it is prone to errors such as drifting, slipping, etc.

We would therefore like to improve the odometry estimate of the ego-motion by matching SIFT features between frames. To find matches in the second view, the odometry information allows us to predict the region to search for each match, and hence more efficiently, as opposed to searching in a much larger unconstrained region.

Once the SIFT features are matched, we can then use the matches in a least-squares procedure to compute a more accurate six degrees-of-freedom (DoF) camera ego-motion and hence better localization. This will also help in adjusting the 3D coordinates of the SIFT landmarks for map building.

3.1. Predicting Feature Characteristics

As our robot is restricted to approximate 2D planar motion, the odometry gives us the approximate movement (p, q) in X and Z directions as well as the orientation rotation (δ).

Given (X, Y, Z), the 3D coordinates of a SIFT landmark and the odometry, we can compute (X', Y', Z'), the relative 3D position, in the new view:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} (X-p)\cos\delta - (Z-q)\sin\delta \\ Y \\ (X-p)\sin\delta + (Z-q)\cos\delta \end{bmatrix}. \qquad (1)$$

Using the typical pinhole camera model, we project this 3D position to its expected image coordinates and compute its expected disparity in the new view

$$\begin{bmatrix} r' \\ c' \\ d' \end{bmatrix} = \begin{bmatrix} v_0 - f\ Y'/Z' \\ u_0 + f\ X'/Z' \\ f\ I/Z' \end{bmatrix} \qquad (2)$$

where (u_0, v_0) are the image centre coordinates, I is the interocular distance and f is the focal length. The expected SIFT orientation remains unchanged. As the scale is inversely related to the distance, the expected scale is given by

$$s' = \frac{s * Z}{Z'}.$$

We can search for the appropriate SIFT landmark match based on the following criteria:

Position. The feature in the new view must be within a 10×10 pixel region of the expected feature position (r', c').

Scale. The expected scale and the measured scale must be within 20% of each other.

Orientation. The orientation difference must be within 20 degrees.

Disparity. The predicted disparity and the measured disparity in the second view must be within 20% of one another.

3.2. Match Results

For the images shown in Figure 2, the rough robot movement from odometry is tabulated in Table 1.

The consecutive frames are matched according to the criteria described above. The matches are stable with respect to the criteria parameters. The specificity of the SIFT features allows the correct features to be matched even if the window size is increased. Table 2 shows the number of matches across frames and the percentage of matches for the different views.

Figure 3 shows the match results visually where the shift in image coordinates of each feature is marked. The white dot indicates the current position and the white cross indicates the new position, with the line showing how each matched SIFT feature moves from one frame to the next, analogous to sparse optic flow. Figures 3(a) and (c) are for a forward motion of 10 cm and Figures 3(b) and (d) are for a clockwise rotation of 5°. It can be seen that all the matches found are correct and consistent.

Table 1. Robot Movement According to Odometry for the Various Views

Figure	Movement
Figure 2(a)	Initial position
Figure 2(b)	Forward 10 cm
Figure 2(c)	Rotate clockwise 5°
Figure 2(d)	Forward 10 cm
Figure 2(e)	Rotate clockwise 5°

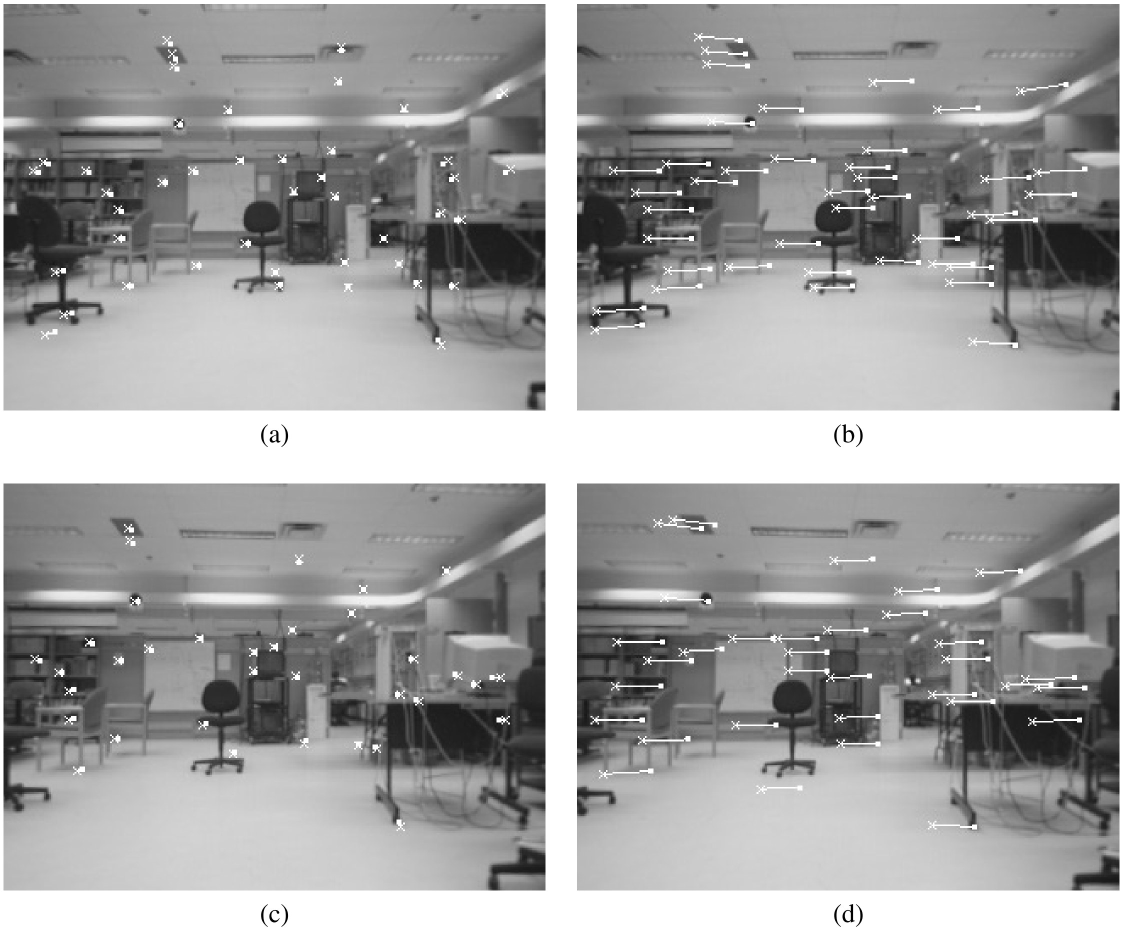


Fig. 3. The SIFT feature matches between consecutive frames: (a) between Figures 2(a) and (b) for a 10 cm forward movement; (b) between Figures 2(b) and (c) for a 5° clockwise rotation; (c) between Figures 2(c) and (d) for a 10 cm forward movement; (d) between Figures 2(d) and (e) for a 5° clockwise rotation.

Table 2. Number of Matches Across Frames and the Percentage of Matches for the Different Views, Based on the SIFT Features Found in Figure 2

Figures to Match	Number of Matches	Percentage of Matches
Figures 2(a) and (b)	43	73%
Figures 2(b) and (c)	41	68%
Figures 2(c) and (d)	35	64%
Figures 2(d) and (e)	33	60%

3.3. Least-Squares Minimization

Once the matches are obtained, ego-motion is determined by finding the camera movement that would bring each projected SIFT landmark into the best alignment with its matching observed feature. To minimize the errors between the projected image coordinates and the observed image coordinates, we employ a least-squares procedure (Lowe 1992) to compute this six DoF camera ego-motion.

Rather than solving directly for the six DoF vector of camera ego-motion, Newton's method computes a vector of corrections **x** to be subtracted from the current estimate, namely the odometry estimate **p**:

$$\mathbf{p}' = \mathbf{p} - \mathbf{x}.$$

Given a vector of error measurements **e** between the expected projection of the SIFT landmarks and the matched image position observed in the new view, we would like to solve for an **x** that would eliminate this error. Therefore, we would like to solve for **x** in

$$\mathbf{J}\,\mathbf{x} = \mathbf{e}$$

where **J** is the Jacobian matrix $J_{i,j} = \partial e_i / \partial x_j$. If there are more measurements than parameters, a least-squares minimization (Gelb 1984) is carried out and **x** is given by

$$\mathbf{J}^{\top}\,\mathbf{J}\,\mathbf{x} = \mathbf{J}^{\top}\,\mathbf{e}. \tag{3}$$

3.4. Setting up the Equation

The ego-motion $\mathbf{p}$ in this case is the six-vector

$$[x\ y\ z\ \theta\ \alpha\ \beta]^\top$$

where $[x\ y\ z]^\top$ are the translations in X, Y, Z directions, and $[\theta\ \alpha\ \beta]^\top$ are the yaw, pitch and roll, respectively. Although the odometry only gives us three DoF, namely the translations in X and Z directions and the orientation, we use a full six DoF for the general motion to account for small non-planar motions.

The error vector $\mathbf{e}$ is of size $2N$ where N is the number of SIFT feature matches between views

$$[e_{r1}\ e_{c1}\ e_{r2}\ e_{c2}\ \ldots\ e_{rN}\ e_{cN}]^\top$$

where (e_{ri}, e_{ci}) are the row and column errors for the ith match. We can estimate the new 3D position (X_i', Y_i', Z_i') from the previous 3D position (X_i, Y_i, Z_i) using odometry according to eq. (1). Afterwards, we can predict its projection (r_i', c_i') in the new view using eq. (2). Together with the measured image position (r_{mi}, c_{mi}) for the matches, we can compute this error vector $\mathbf{e}$ where

$$\begin{aligned} e_{ri} &= r_i' - r_{mi} \\ e_{ci} &= c_i' - c_{mi}. \end{aligned}$$

$\mathbf{J}$ is a $2N \times 6$ matrix whose $(2i-1)$th row is

$$[\frac{\partial r_i}{\partial x}\frac{\partial r_i}{\partial y}\frac{\partial r_i}{\partial z}\frac{\partial r_i}{\partial \theta}\frac{\partial r_i}{\partial \alpha}\frac{\partial r_i}{\partial \beta}]$$

and whose $2i$th row is

$$[\frac{\partial c_i}{\partial x}\frac{\partial c_i}{\partial y}\frac{\partial c_i}{\partial z}\frac{\partial c_i}{\partial \theta}\frac{\partial c_i}{\partial \alpha}\frac{\partial c_i}{\partial \beta}].$$

The computation of these partial derivatives is performed numerically. For example, to compute $\partial c_i/\partial x$, we perturb x by a small amount Δx and compute how much c_i changes, i.e.,

$$\Delta c_i = (u_0 + \frac{f\ (X_i' - \Delta x)}{Z_i'}) - (u_0 + \frac{f\ X_i'}{Z_i'}).$$

The rate of change $\Delta c_i/\Delta x$ of c_i with respect to x approximates $\partial c_i/\partial x$ as Δx tends to zero; similarly for the other partial derivatives terms.

We use Gaussian elimination with pivoting to solve eq. (3) which is a linear system of six equations. The least-squares error $\hat{\mathbf{e}}$ can be computed using the correction terms $\hat{\mathbf{x}}$ found

$$\hat{\mathbf{e}} = \mathbf{J}\ \hat{\mathbf{x}}$$

and for each feature, the residual error E_i is given by

$$E_i = \sqrt{\hat{e}_{ri}^2 + \hat{e}_{ci}^2}.$$

The good feature matching quality implies a very high percentage of inliers, therefore outliers are simply eliminated by discarding features with significant residual errors E_i (currently two pixels). The minimization is repeated with the remainder matches to obtain the new correction terms.

3.5. Results

For each of the motion matches shown in Figure 3, we pass all the SIFT matches to the least-squares procedure with the odometry as the initial estimate of ego-motion. The results obtained are shown in Table 3, where the least-squares estimate $[x, y, z, \theta, \alpha, \beta]$ corresponds to the translations in X, Y, Z directions, yaw, pitch and roll respectively.

A 3D geometric representation should retain valuable structural information while discarding the large amount of redundant pixel data in an image. This is crucial for real-time computer vision and mobile robot systems because there are too many pixels in a video-rate image sequence for processing. Corner features were used in Harris (1992) for tracking, whereas we have employed SIFT features for our system. SIFT features are largely invariant to translations, scaling, rotation, and illumination changes, and hence are more stable than corners for tracking over time.

We observe the same scene from slightly different directions at various distances and we investigate the stability of SIFT landmarks in the environment.

Figure 4 shows six views of the same scene: Figure 4(b) at the original robot position; Figure 4(a) at the same distance around 10° from the left; Figure 4(c) at the same distance around 10° from the right; Figure 4(e) around 60 cm in front; Figure 4(d) around 60 cm in front and 10° from the left; Figure 4(f) around 60 cm in front and 10° from the right.

We compare the SIFT scale and orientation of some landmarks which appear in all six views, as marked in Figure 4(a). Since the SIFT scale is inversely proportional to the distance, we use a scale measure given by the product between the SIFT scale and the landmark distance, which should remain more or less unchanged when observed at different views. The orientation is currently obtained at a discretized space.

The results in Table 4 show the scale measure and the orientation of the corresponding landmarks from different views in Figure 4(a)–(f).

We can see that, for each landmark, the scale measure and orientation are very stable from different views. This will allow landmarks to be matched consistently across frames.

4. Landmark Tracking

After matching SIFT features between frames, we would like to maintain a database map containing the SIFT features detected and to use this database of landmarks to match features found in subsequent views. The initial camera coordinates frame is used as a reference and all landmarks are relative to this frame.

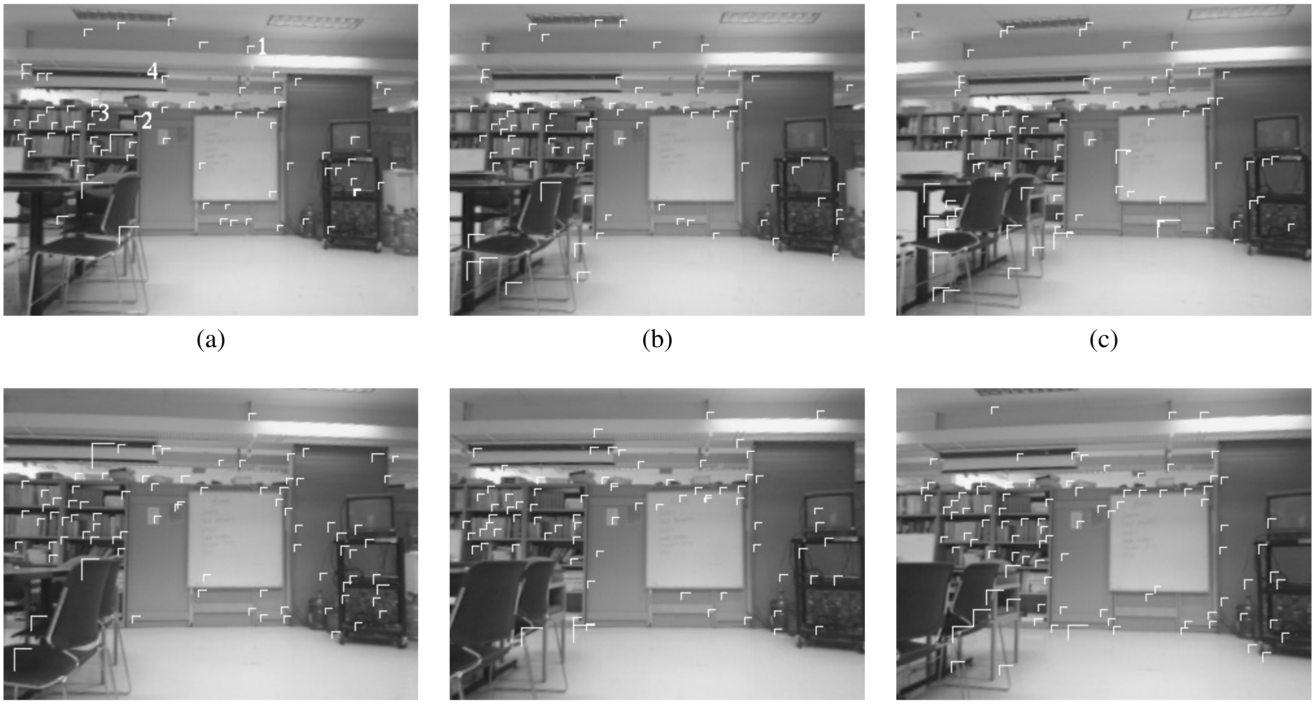
For each SIFT feature that has been stereo matched and localized in 3D coordinates, its entry in the database is

$$[X, Y, Z, s, o, l]$$

where (X, Y, Z) is the current 3D position of the SIFT

Table 3. Least-squares Estimate of the Six DoF Robot Ego-motion Based on the SIFT Features Matches Across Frames in Figure 3

Figure	Odometry	Mean E_i	Least-Squares Estimate
3(a)	$q = 10$ cm	1.125	[1.353 cm, −0.534 cm, 11.136 cm, 0.059°, −0.055°, −0.029°]
3(b)	$\delta = 5°$	1.268	[0.711 cm, 0.008 cm, −0.989 cm, 4.706°, 0.059°, −0.132°]
3(c)	$q = 10$ cm	0.882	[−0.246 cm, −0.261 cm, 9.604 cm, 0.183°, 0.089°, −0.101°]
3(d)	$\delta = 5°$	1.314	[1.562 cm, 0.287 cm, −0.562 cm, 4.596°, 0.004°, −0.073°]



(a) (b) (c)

(d) (e) (f)

Fig. 4. A scene observed from different views. The four landmarks considered are numbered in (a).

Table 4. The Scale and Orientation for Some SIFT Landmarks from Different Views, Showing the Landmark Stability

(Scale,Orientation)	Landmark 1	Landmark 2	Landmark 3	Landmark 4
View (a)	(23.42, −1.31)	(10.69, −1.66)	(17.34, 1.48)	(10.31, −1.48)
View (b)	(23.28, −1.31)	(10.60, −1.66)	(17.37, 1.48)	(10.44, −1.48)
View (c)	(24.27, −1.31)	(10.87, −1.66)	(17.54, 1.48)	(10.50, −1.48)
View (d)	(22.91, −1.31)	(11.91, −1.66)	(15.68, 1.43)	(9.22, −1.48)
View (e)	(22.99, −1.31)	(9.84, −1.66)	(15.10, 1.48)	(9.26, −1.48)
View (f)	(22.95, −1.31)	(9.82, −1.66)	(16.34, 1.48)	(9.48, −1.48)

The scale measure is given by the product between the SIFT scale and the landmark distance and the orientation is obtained at a discretized space.

landmark relative to the initial coordinates frame, (s, o) are the scale and orientation of the landmark, and l is a count to indicate over how many consecutive frames this landmark has been missed.

Over subsequent frames, we would like to maintain this database, add new entries to it, track features and prune entries when appropriate, in order to cater for dynamic environments and occlusions.

4.1. Track Maintenance

Between frames, we obtain a rough estimate of camera ego-motion from robot odometry to predict the feature characteristics for the landmarks in the next frame, as discussed in Section 3.1. There are the following types of landmarks to consider:

Type I. This landmark is not expected to be within view in the next frame. Therefore, it is not being matched and its miss count remains unchanged.

Type II. This landmark is expected to be within view, but no matches can be found in the next frame. Its miss count is incremented by 1.

Type III. This landmark is within view and a match is found according to the position, scale, orientation and disparity criteria described before. Its miss count is reset to zero.

Type IV. This is a new landmark corresponding to a SIFT feature in the new view which does not match any existing landmarks in the database. A new track is initiated with a miss count of 0.

All the Type III features matched are then used in the least-squares minimization procedure to obtain a better estimate for the camera ego-motion. The landmarks in the database are updated by averaging for now. This update will be based on Kalman filters (Bar-Shalom and Fortmann 1988) and the least-squares ego-motion estimate will be processed further in Section 7.

If there are insufficient Type III matches due to occlusion for instance, the odometry will be used as the ego-motion for the current frame.

4.2. Track Initiation

Initially the database is empty. When SIFT features from the first frame arrive, we start a new track for each of the features initializing their miss count l to 0. In subsequent frames, a new track is initiated for each of the Type IV features.

We may change this policy to only initiate a track when a particular feature appears consistently over a few frames.

4.3. Track Termination

If the miss count l of any landmark in the database reaches a predefined limit N (20 was used in experiments), i.e., this landmark has not been observed at the position it is supposed to appear for N consecutive times, this landmark track is terminated and pruned from the database. Instead of discarding a missed track immediately, we can cater for temporary occlusion by adjusting N. Moreover, this can deal with volatile landmarks such as chairs. After a movable landmark has been moved, its old entry will be discarded as it is no longer observed at the expected position, and its new entry will be added.

4.4. Field of View

To check whether or not a landmark in the database is expected to be within the field of view in the next frame, we compute the expected 3D coordinates (X', Y', Z') from the current coordinates and the odometry, according to eq. (1).

The landmark is expected to be within view if the following three conditions are satisfied:

- $Z' > 0$ for being in front of the camera;
- $\tan^{-1}(|X'|/Z') < V/2$ for being within the horizontal field of view;
- $\tan^{-1}(|Y'|/Z') < V/2$ for being within the vertical field of view.

Here V is the camera field of view, currently 60° for the Triclops camera.

5. Experimental Results

SIFT feature detection, stereo matching, ego-motion estimation and tracking algorithms have been implemented in our robot system, a Real World Interface (RWI) B-14 mobile robot as shown in Figure 5. A SIFT database is kept to track the landmarks over frames.

As the robot camera height does not change much over flat ground, we have reduced the estimation to five parameters, forcing the height change parameter to zero. Depending on the distribution of features in the scene, there can be some ambiguity between a yaw rotation and a sideways movement, which is a well-known problem. The odometry information can be used to stabilize our least-squares minimization in these ill-conditioned cases.

Moreover, we set a limit to the correction terms allowed for the least-squares minimization. Because the odometry information for between frame movement should be quite good, the correction terms required should be small. This will safeguard frames with erroneous matches that may lead to excessive correction terms and affect the subsequent estimation.

Fig. 5. Our RWI B-14 mobile robot equipped with the Triclops.

The odometry information only gives the X, Z movement and rotation of the *robot*, but our ego-motion estimation determines the movement of the *camera*. Since the Triclops system is not placed in the centre of the robot, we need to adjust the odometry information to give an initial approximation for the camera motion. For example, a mere robot rotation does not lead to changes in X and Z values of the odometry, but the camera itself will have displaced in the X and Z directions.

The following experiment was carried out online, i.e., the images are captured and processed, the database is kept and updated on the fly while the robot is moving around. We manually drive the robot to go around a chair in the laboratory for one loop and to come back. At each frame, it keeps track of the SIFT landmarks in the database, adds new ones and updates existing ones if matched.

Figure 6 shows some frames of the 320 × 240 image sequence (249 frames in total) captured while the robot is moving around. The white markers indicate the SIFT features found. At the end, a total of 3590 SIFT landmarks with 3D positions are gathered in the SIFT database, which are relative to the initial coordinates frame. The typical time required for each iteration is around 0.4–0.5 s, in which the majority of time is spent on the SIFT feature detection stage.

Figure 7 shows the bird's-eye view of all these landmarks. Consistent clusters are observed corresponding to objects such as chairs, shelves, posters and computers in the scene.

In this experiment, the robot has traversed forward more than 4 m and then has come back. The maximum robot translation and rotation speeds are set to around 40 cm/s and 10°/s, respectively, to make sure that there are sufficiently many matches between consecutive frames.

The accuracy of the ego-motion estimation depends on the SIFT features and their distribution and the number of matches. In this experiment, there are sufficiently many matches at each frame, ranging mostly between 40 and 60, depending on the particular part of the laboratory and the viewing direction.

At the end when the robot comes back to the original position (0,0,0) judged visually: the SIFT estimate is X: −2.09 cm; Y: 0 cm; Z: −3.91 cm; θ: 0.30°; α: 2.10°; β: −2.02°.

6. SIFT Database

Our basic approach has been described above, but there are various enhancements dealing with the SIFT database that can help our tracking to be more robust and our map building to be more stable.

6.1. Database Entry

In order to find how reliable a certain SIFT landmark is in the database, we need some information regarding how many times this landmark has been matched and how many times it has not been matched so far, not just the number of times it has not been matched consecutively. Therefore, the new database entry is

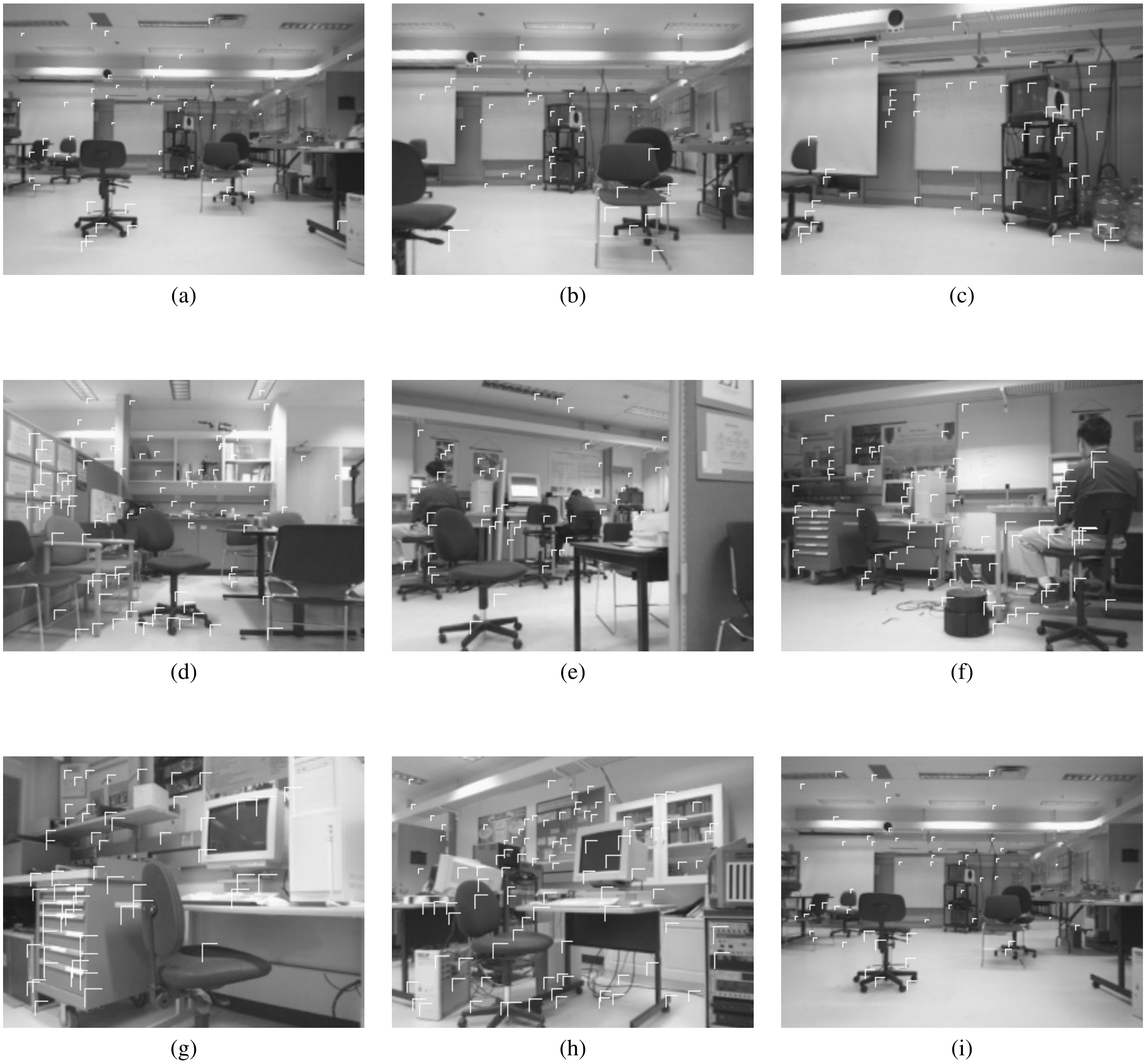
$$[X, Y, Z, s, o, m, n, l]$$

where l is still the count for the number of times being missed consecutively, which is used to decide whether or not the landmark should be pruned. m is a count for the number of times it has been missed so far, i.e., an accumulative count for l. n is a count for the number of times it has been seen so far.

With the new information, we can impose a restriction that, for a feature to be considered as valid, its n count has to exceed some threshold. Each feature has to appear at least three times ($n \geq 3$) in order to be considered as a valid feature; this is to eliminate false alarms and noise, as it is highly unlikely that some noise will cause a feature to match in the right, left and top images for three times (a total of nine camera views).

In this experiment, we move the robot around the laboratory environment without the chair in the middle. In order to demonstrate visually that the SIFT database map is 3D, we use the visualization package *Geomview*.[3] The user can interactively view the 3D map from different elevation angles, pan angles or distances. Figure 8 shows several views of the 3D SIFT map from different angles. We can see that the centre region is clear, as false alarms and noise features are discarded. The SIFT landmarks correspond well to actual objects in the laboratory.

[3] www.geomview.org



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 6. Frames of an image sequence with SIFT features marked: (a) 1st frame; (b) 30th frame; (c) 60th frame; (d) 90th frame; (e) 120th frame; (f) 150th frame; (g) 180th frame; (h) 210th frame; (i) 240th frame.

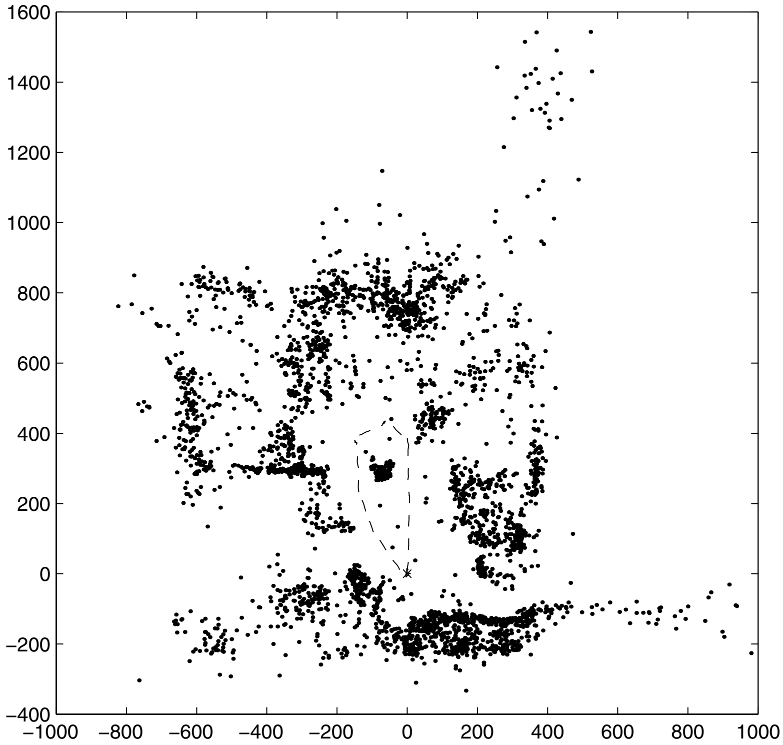


Fig. 7. Bird's-eye view of the SIFT landmarks in the database. The cross at (0,0) indicates the initial robot position and the dashed line indicates the robot path while obtaining the images shown in Figure 6.

6.2. Permanent Landmarks

In some scene where there could be many volatile landmarks, e.g., when someone blocks the camera view for a while, because it has not been matched for a larger number of consecutive frames, the previously observed good landmarks will be discarded.

Therefore, when the environment is clear, we can build a database of SIFT landmarks beforehand and mark them as permanent landmarks, if they are valid (having appeared in at least three frames) and if the percentage of their occurrence (given by $\frac{n}{n+m}$) is above a certain threshold (currently 30%). Afterwards, this set of reliable landmarks will not be wiped out even if they are being missed for many consecutive frames. They are important for subsequent localization after the view is unblocked.

6.3. Viewpoint Variation

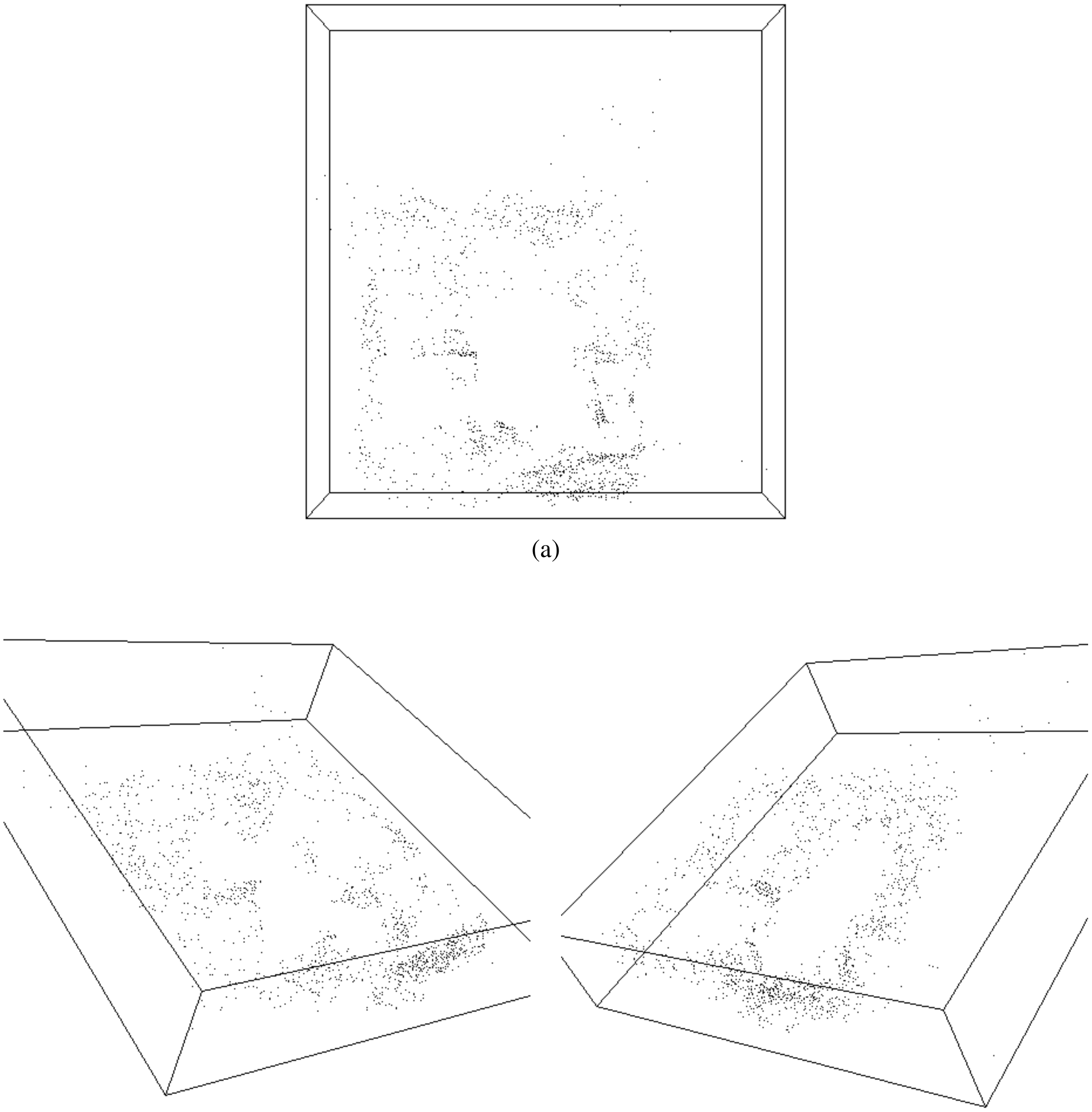
Although SIFT features are invariant in image orientation and scale, they are image projections of 3D landmarks and hence vary with large changes of viewpoints and are subject to landmark occlusion.

For example, when the front of an object is seen first, after the robot moves around and views the object from the back, the image feature, in general, is completely different. As the original feature may not be observable from this viewpoint, or is observable but appears different, its miss count will increase gradually and it will be pruned even though it is still there.

Therefore, each SIFT characteristic (scale and orientation) is associated with a view vector keeping track of the viewpoint from which the landmark is observed. Subsequently, if the new view direction differs from the original view direction by more than a threshold (currently set to 20°), its miss count will not be incremented even if it does not match. Feature matching is not considered at all when the new view direction differs by more than 90°. In this way, we can avoid corrupting the feature information gathered earlier by the current partial view of the world.

Moreover, we allow each SIFT landmark to have more than one SIFT characteristic. If a feature matches from a direction larger than 20°, we add a new view vector with the new SIFT characteristic to the existing landmark. Therefore, a database landmark can have multiple SIFT characteristics $(s_i, o_i, \mathbf{v}_i)$ where s_i and o_i are the scale and orientation for the view direction $\mathbf{v}_i$. Over time, if a landmark is observed from various directions, much richer SIFT information is gathered. The matching procedure is as follows:

- compute view vector $\mathbf{v}$ between the database landmark and the current robot position;



(a)

(b)

(c)

Fig. 8. The 3D SIFT database map viewed from different angles in *Geomview*. Each feature has appeared consistently in at least nine views: (a) from top; (b) from left; (c) from right.

- find the existing view direction $\mathbf{v}_i$ associated with the database landmark which is *closest* to $\mathbf{v}$;
- view vectors are normalized and, therefore, angle ϕ between $\mathbf{v}$ and $\mathbf{v}_i$ can be computed as the arccosine of the dot product between the two vectors

$$\mathbf{v} \cdot \mathbf{v}_i = |\mathbf{v}||\mathbf{v}_i| \cos\phi \Rightarrow \phi = \cos^{-1}(\mathbf{v} \cdot \mathbf{v}_i);$$

- omit the following step if ϕ is greater than 90 degrees;
- check if ϕ is less than 20 degrees
  - if so, update the existing s and o if feature matching succeeds, or increment miss count if feature matching fails;
  - else, add a new entry of SIFT characteristic $(s, o, \mathbf{v})$ to the existing landmark if feature matching succeeds.

The 3D position of the landmark is updated if matched and the counts are updated accordingly. The new database entry becomes

$$[X, Y, Z, m, n, l, k, s_1, o_1, \mathbf{v}_1, s_2, o_2, \mathbf{v}_2, \dots, s_k, o_k, \mathbf{v}_k]$$

where k is the number of SIFT characteristics associated with this SIFT landmark.

7. Error Modeling

There are various errors such as noise and quantization associated with the images and the features found. They introduce inaccuracy in both the position of the landmarks as well as the least-squares estimation of the robot position. We would like to know how reliable the estimations are and therefore we incorporate a covariance matrix into each of the SIFT landmarks in the database.

We employ a Kalman filter (Bar-Shalom and Fortmann 1988) for each database SIFT landmark with a 3 × 3 covariance matrix for its position. The robot pose uncertainty is also required because landmarks in the current frame are to be transformed to the initial coordinates frame using the robot pose estimate.

When a match is found in the current frame, the covariance matrix for the landmark in the current frame will be transformed using the robot pose covariance and then combined with the covariance matrix in the database so far, and its 3D position will be updated accordingly.

7.1. Robot Pose Covariance

We use the odometry as the initial approximation for the robot pose least-squares minimization, and the resulting least-squares estimate is regarded as the final estimate. There are errors associated with both the odometry and the least-squares localization. Therefore, we will employ a Kalman filter to fuse these two sources of information using their covariances.

The state in our Kalman filter is the five DoF robot camera pose (assuming fixed height). We have a prediction stage for the state and the state covariance using the odometry information. Then using the SIFT localization as the measurement model, we can update the state and covariance accordingly to obtain a better estimate.

7.1.1. Odometry Model

Odometry is widely used to provide easy accessible real-time positioning information for mobile robot. However, it is based on the assumption that wheel revolution can be translated into linear displacement relative to the floor, so odometry is prone to errors. There are two types of odometry errors: systematic and non-systematic. Various methods of modeling and reducing these errors have been proposed (Everett 1995; Borenstein et al. 1996).

Systematic errors (Borenstein and Feng 1996) include unequal wheel diameters, wheelbase uncertainty, wheel misalignment, etc. They accumulate constantly and orientation errors dominate because they can grow without bound into translational position errors (Crowley 1989).

Non-systematic errors (Borenstein 1995) include traveling over uneven floors, wheel slippage due to slippery floor, skidding, etc. These are caused by the interaction of the robot with unpredictable features of the environment and hence are difficult to bound.

In the odometry modeling below, we look at the overall odometry uncertainty for our prediction. For our robot, it can only move forward (w) and rotate (δ) with odometry measurements (p, q, δ). Referring to Figure 9, we have

$$p = w \sin\delta \qquad (4)$$

$$q = w \cos\delta \qquad (5)$$

$$w^2 = p^2 + q^2. \qquad (6)$$

We can compute the variances of p and q in terms of the variances of w and δ using first order error propagation formulae (Bevington and Robinson 1992). From eq. (4), we have

$$\sigma_p^2 = \sigma_w^2 \sin^2\delta + \sigma_\delta^2 w^2 \cos^2\delta.$$

From eq. (5), we have

$$\sigma_q^2 = \sigma_w^2 \cos^2\delta + \sigma_\delta^2 w^2 \sin^2\delta.$$

We also need to compute the cross-correlation terms, since (p, q, δ) are not independent of each other. For example, any changes to δ will also affect p and q.

Computing the first-order error terms for eq. (6), we have

$$w^2\sigma_w^2 = p^2\sigma_p^2 + q^2\sigma_q^2 + 2pq\sigma_{pq}^2$$

and hence

$$\sigma_{pq}^2 = \frac{w^2\sigma_w^2 - p^2\sigma_p^2 - q^2\sigma_q^2}{2pq}.$$

Rewriting eq. (4) as

$$w = \frac{p}{\sin\delta}$$

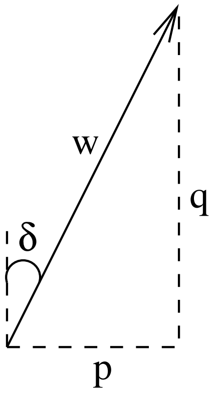


Fig. 9. Relationship between robot motion and odometry measurements.

and computing the first-order error terms, we have

$$\frac{\sigma_w^2}{w^2} = \frac{\sigma_p^2}{p^2} + \frac{\sigma_\delta^2 \cos^2\delta}{\sin^2\delta} - \frac{2\cos\delta\,\sigma_{p\delta}^2}{p\sin\delta}$$

and hence

$$\sigma_{p\delta}^2 = \frac{p\sin\delta}{2\cos\delta}\left(\frac{\sigma_p^2}{p^2} - \frac{\sigma_w^2}{w^2} + \frac{\sigma_\delta^2\cos^2\delta}{\sin^2\delta}\right).$$

Similarly, we rewrite eq. (5) as

$$w = \frac{q}{\cos\delta}.$$

Computing the first-order error terms, we have

$$\frac{\sigma_w^2}{w^2} = \frac{\sigma_q^2}{q^2} + \frac{\sigma_\delta^2 \sin^2\delta}{\cos^2\delta} + \frac{2\sin\delta\,\sigma_{q\delta}^2}{q\cos\delta}$$

and hence

$$\sigma_{q\delta}^2 = \frac{q\cos\delta}{2\sin\delta}\left(\frac{\sigma_w^2}{w^2} - \frac{\sigma_q^2}{q^2} - \frac{\sigma_\delta^2\sin^2\delta}{\cos^2\delta}\right).$$

From these results, we can compute the odometry covariance matrix

$$\begin{bmatrix} \sigma_p^2 & \sigma_{pq}^2 & \sigma_{p\delta}^2 \\ \sigma_{pq}^2 & \sigma_q^2 & \sigma_{q\delta}^2 \\ \sigma_{p\delta}^2 & \sigma_{q\delta}^2 & \sigma_\delta^2 \end{bmatrix}$$

assuming some σ_w^2 and σ_δ^2 values which can be acquired experimentally, for example, by checking the wheel slipping rate of the robot odometry system.

7.1.2. Prediction

Following the standard Kalman filter notation, we let $\mathbf{x}(k|k)$ be the state $[x, z, \theta, \alpha, \beta]$ at time k given information up to time k. Letting $\mathbf{P}(k|k)$ be the state covariance, $\mathbf{u}(k)$ be the odometry information $[p, q, \delta, 0, 0]$ for how much the robot has translated and rotated, and $\mathbf{Q}(k)$ be the covariance for $\mathbf{u}(k)$, we have the state prediction

$$\mathbf{x}(k+1|k) = \mathbf{f}(\mathbf{x}(k|k), \mathbf{u}(k))$$

where $\mathbf{f}$ is the state transition function, in this case, just simply adding $\mathbf{u}$ to $\mathbf{x}$. The state covariance prediction is

$$\mathbf{P}(k+1|k) = \mathbf{P}(k|k) + \mathbf{Q}(k).$$

7.1.3. Measurement

The measurement prediction is

$$\mathbf{z}(k+1|k) = \mathbf{H}(k+1)\mathbf{x}(k+1|k)$$

where $\mathbf{H}$ is the identity matrix because both the state and measurement are the robot pose.

Matching the current features to the SIFT database, we use least-squares minimization (Section 3.3) to estimate the robot position $\mathbf{x}_{LS}$, provided that there are sufficiently many matches. Innovation is the difference between the predicted measurement and the actual measurement, given by

$$\mathbf{v}(k+1) = \mathbf{z}(k+1) - \mathbf{z}(k+1|k) = \mathbf{x}_{LS} - \mathbf{x}(k+1|k).$$

The covariance $\mathbf{P}_{LS}$ for the measurement can be obtained by computing the inverse of $\mathbf{J}^{\top}\,\mathbf{J}$ (Lowe 1992) in Section 3.3. The innovation covariance is

$$\begin{aligned}\mathbf{S}(k+1) &= \mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}(k+1)^{\mathrm{T}} + \mathbf{P}_{LS} \\ &= \mathbf{P}(k+1|k) + \mathbf{P}_{LS}.\end{aligned}$$

7.1.4. Update

The filter gain is

$$\begin{aligned}\mathbf{W}(k+1) &= \mathbf{P}(k+1|k)\mathbf{H}(k+1)^{\mathrm{T}}\mathbf{S}^{-1}(k+1) \\ &= \mathbf{P}(k+1|k)[\mathbf{P}(k+1|k) + \mathbf{P}_{LS}]^{-1}.\end{aligned}$$

The state update is

$$\mathbf{x}(k+1|k+1) = \mathbf{x}(k+1|k) + \mathbf{W}(k+1)[\mathbf{x}_{LS} - \mathbf{x}(k+1|k)].$$

The covariance update is

$$\begin{aligned}\mathbf{P}(k+1|k+1) &= \mathbf{P}(k+1|k) \\ &\quad -\mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^{\mathrm{T}}(k+1) \\ &= \mathbf{P}(k+1|k) - \mathbf{P}(k+1|k) \\ &\quad [\mathbf{P}(k+1|k) + \mathbf{P}_{LS}]^{-\mathrm{T}}\mathbf{P}(k+1|k)^{\mathrm{T}}.\end{aligned}$$

When there are not enough matches (less than six), we do not obtain least-squares measurement to update the prediction. The state and the covariance prediction will be used as the state and covariance for the next frame. The covariance will shrink as soon as enough matches are found to provide a least-squares update.

7.2. Landmark Position Covariance

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.

Re-arranging eq. (2), we have

$$X = \frac{(c - u_0)I}{d}$$

$$Y = \frac{I(v_0 - r)}{d}$$

$$Z = \frac{fI}{d}.$$

For the first-order error propagation (Bevington and Robinson 1992), we have

$$\sigma_X^2 = \frac{I^2\sigma_c^2}{d^2} + \frac{I^2(c-u_0)^2\sigma_d^2}{d^4}$$

$$\sigma_Y^2 = \frac{I^2\sigma_r^2}{d^2} + \frac{I^2(v_0-r)^2\sigma_d^2}{d^4}$$

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

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.

Based on the results from Section 3.5 where the mean least-squares image error is around one pixel, we assume $\sigma_r^2 = 0.5$, $\sigma_c^2 = 0.5$ and $\sigma_d^2 = 1$. Knowing the intrinsic parameters of our system, we can compute the variances for the landmark 3D positions according to the error propagation formulae above.

We use the robot pose estimate to transform landmarks in the current coordinates frame into the reference 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 reference frame, we have

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

where $\mathbf{r}_{obs}$ and $\mathbf{r}_{new}$ are the observed position in the current frame and the transformed position in the reference 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:

$$\mathbf{P}_\theta = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\mathbf{P}_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix}$$

$$\mathbf{P}_\beta = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

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. The error propagation details are in Appendix A.

Then, 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

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

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

On the other hand, the scale and orientation of the database landmarks are updated by averaging over all the frames. The database entry for each landmark is augmented with the 3 × 3 covariance matrix of its position.

7.3. Results

Incorporating the above uncertainty analysis, a Kalman filter is initiated for each landmark and updated over frames. Figure 10 shows the bird's-eye view of the SIFT database after 56 frames with 2116 landmarks in the database. This is after the robot has spun around once. Figure 11 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. This is after the robot has traversed around our laboratory. When a landmark is observed repeatedly, its uncertainty ellipse shrinks while its positional uncertainty decreases.

The landmarks are 3D and their uncertainties are represented as ellipsoids, but ellipses are shown in the bird's-eye view. Error ellipses covering one standard deviation in either sides of X and Z directions are shown.

It can be seen that the uncertainties for landmarks closer to the robot tend to be lower, as expected for landmarks with larger disparities. Visual judgement indicates that the SIFT landmarks correspond well to actual objects in the laboratory.

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. On the other hand, the robot is facing forward in the Z direction when the landmarks at the top of the map are viewed, therefore the ellipses are elongated in the Z direction.

We also see some relatively large ellipses at the upper right corner and these correspond to landmarks far away in the next laboratory. Since the landmark position is computed from the inverse of disparity, for landmarks of very small disparities, any small image errors can lead to a very large positional uncertainty.

Figure 12 shows the same database map, but with the landmark uncertainty shown in terms of intensity instead of the ellipses. We can see that the more uncertain landmarks are lighter than those with lower uncertainty. Without the ellipse clutter in Figure 11, the more reliable landmarks are now more visible.

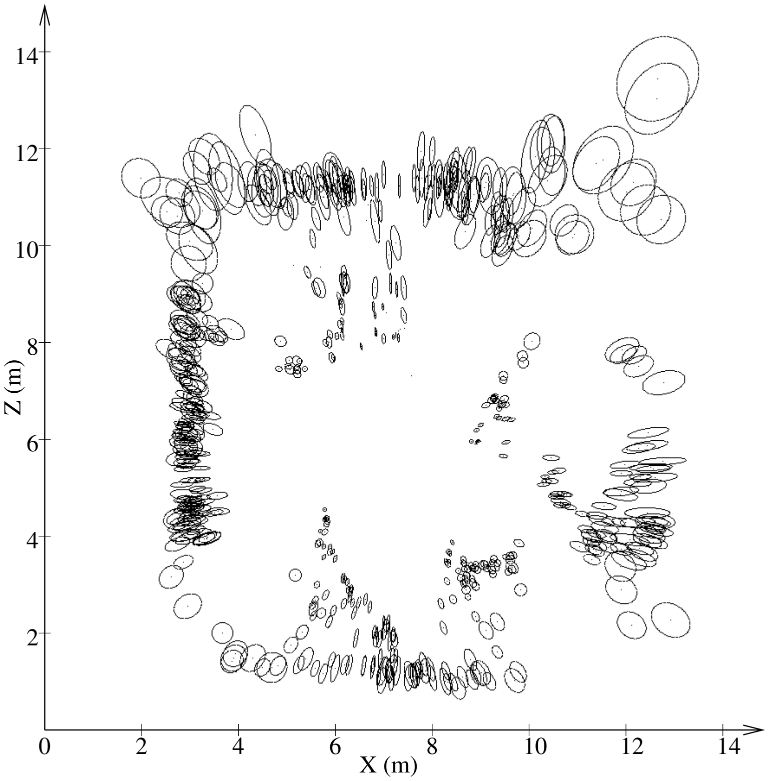


Fig. 10. Bird's-eye view of the 3D SIFT database map, showing the uncertainty ellipses of the landmarks after spinning around once.

The robot trajectory with uncertainty ellipses are shown in Figure 13, where the ellipses cover two standard deviations in either sides of X and Z directions. All the ellipses are relatively small as SIFT landmarks are matched well at all frames. It can be seen that the robot pose has a higher uncertainty in the direction that it is facing, due to the higher uncertainty in landmark depth.

In the following experiment, the robot is driven to rotate repeatedly in the laboratory environment. Figure 14(a) shows the robot orientation over frames with its corresponding standard deviation shown in Figure 14(b). Figures 14(c) and (d) show the robot pose uncertainty in the X and Z directions, respectively. We can see that the uncertainty varies slightly within each cycle, depending on the particular view it observes. The overall robot pose uncertainty for each cycle decreases over frames, showing that by observing the scene repeatedly, the 3D SIFT landmark uncertainty reduces and hence a better robot pose is obtained. The robot uncertainty also decreases initially for this reason, when it has not started rotating.

There are two components of uncertainty for both the robot and the landmarks: one is relative to the initial robot pose in a robot-based world and the other is the initial robot uncertainty relative to the external global world. As a result, there are two types of maps: relative maps and absolute maps. Relative maps take into account the first uncertainty only whereas absolute maps take both into account. To obtain absolute uncertainty from relative uncertainty, additional uncertainty for the initial robot pose needs to be added. Our SLAM builds a relative map where the uncertainty is with respect to the robot world and not the external one. Our experiments show that our estimated map and robot pose converge monotonically, agreeing with the analysis described in Dissanayake et al. (2001).

8. Conclusion

In this paper, we have proposed a vision-based mobile robot localization and map building algorithm based on SIFT features. Being scale and orientation invariant, SIFT features are good natural visual landmarks for tracking over a long period of time from different views. These tracked landmarks are used for concurrent robot pose estimation and 3D map building, with promising results shown. As there are errors associated with image features, error analysis is important to tell us how well the landmarks are localized.

We have shown that it is possible to build accurate maps efficiently without keeping the correlation between landmarks.

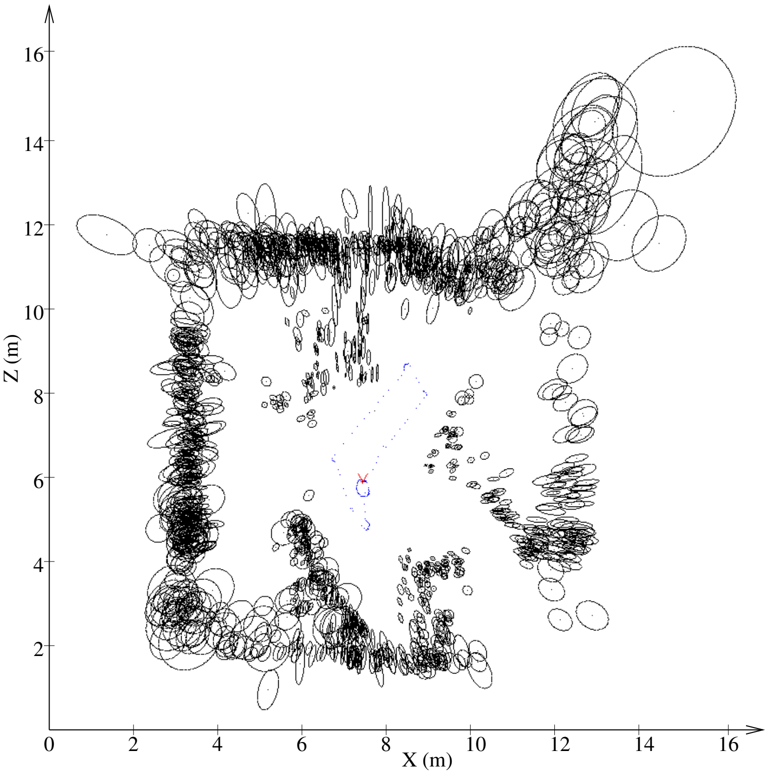


Fig. 11. Bird's-eye view of the 3D SIFT database map, showing the uncertainty ellipses of the landmarks, and the robot trajectory after traversing around our laboratory. Note that the smallest ellipses represent the most reliable and useful landmarks.

Both time and memory efficiency are important and would be seriously affected by attempting to keep the full correlation matrix.

The algorithm currently runs at around 2 Hz for 320 × 240 images on our mobile robot with a Pentium III 700 MHz processor. As the majority of the processing time is spent on SIFT feature extraction, SIFT optimization is being investigated.

Further experiments in larger environments are planned to evaluate the scalability of our approach. As we keep track of the robot pose, features in the current frame will only be matched to SIFT landmarks in a particular region of the database, hence we do not expect feature matching to be an issue. Moreover, the specificity of the SIFT features can be increased if necessary to maintain their distinctiveness (Lowe 1999).

At present, the map is re-used only if the robot starts up again at the last stop position or if the robot starts at the position of the initial reference frame. Preliminary work on the "kidnapped robot" problem, i.e., global localization using the SIFT database map, has been positive (Se et al. 2001a). This will allow the robot to re-use the map at any arbitrary robot position by matching the rich SIFT database.

We are currently looking into recognizing the return to a previously mapped area after following a long path away from the area, i.e., closing the loop and detecting the occurrences of drift and correcting for it. Moreover, we intend to study the feasibility of using the SIFT landmark-based uncertainty map for path planning, obstacle avoidance and other high-level tasks.

Appendix: Error Propagation

In general, given

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

where $\mathbf{P}$ is a 3 × 3 matrix, $\mathbf{X}$ and $\mathbf{X}'$ are the three-vectors for the old and new positions, respectively. If there are errors associated with both $\mathbf{P}$ and $\mathbf{X}$: $\Lambda_\mathbf{P}$ (9 × 9 covariance for $\mathbf{P}$) and $\Lambda_\mathbf{X}$ (3 × 3 covariance for $\mathbf{X}$) which is

$$\begin{bmatrix} \sigma_X^2 & \sigma_{XY}^2 & \sigma_{XZ}^2 \\ \sigma_{XY}^2 & \sigma_Y^2 & \sigma_{YZ}^2 \\ \sigma_{XZ}^2 & \sigma_{YZ}^2 & \sigma_Z^2 \end{bmatrix}$$

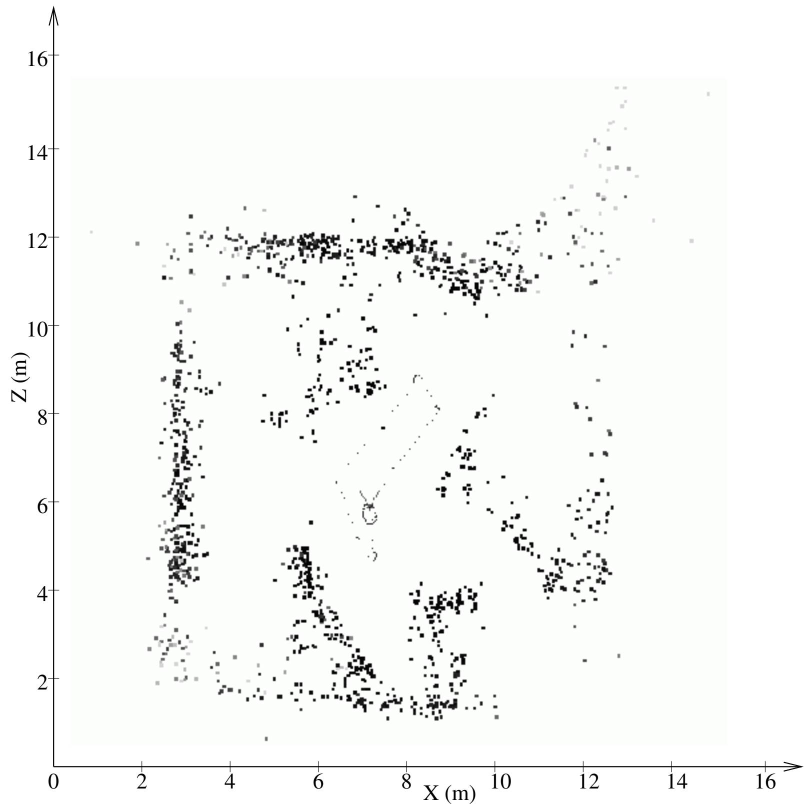


Fig. 12. Bird's-eye view of the 3D SIFT database map, where the landmark intensity is proportional to the uncertainty. The more uncertain a landmark is, it is shown in a lighter shade.

the 3×3 covariance for the resulting vector $\mathbf{X}'$ is given by

$$\left[\begin{array}{ccc|c} \mathbf{X}^\top & \mathbf{0} & \mathbf{0} & \\ \mathbf{0} & \mathbf{X}^\top & \mathbf{0} & \mathbf{P} \\ \mathbf{0} & \mathbf{0} & \mathbf{X}^\top & \end{array}\right] \left[\begin{array}{cc} \Lambda_{\mathbf{P}} & \mathbf{0} \\ \mathbf{0} & \Lambda_{\mathbf{X}} \end{array}\right] \left[\begin{array}{ccc} \mathbf{X} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{X} \\ \hline & \mathbf{P}^\top & \end{array}\right]. \qquad (7)$$

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

Assuming the roll, pitch and yaw components are independent due to their small size, the transformation proceeds in four stages: $\mathbf{P}_\beta$ (roll), $\mathbf{P}_\alpha$ (pitch), $\mathbf{P}_\theta$ (yaw) and then $\mathbf{V}$ (translations). We have already obtained the variances (σ_β^2, σ_α^2 and σ_θ^2) for these parameters from the robot position covariance. We will look at the transformation required for each stage and how the landmark position uncertainty propagates.

A.1. Roll Transformation

The 9×9 covariance matrix for the roll transformation is

$$\left[\begin{array}{ccccccccc}
\sigma_\beta^2 \sin^2\beta & -\sigma_\beta^2 \sin\beta\cos\beta & 0 & \sigma_\beta^2 \sin\beta\cos\beta & \sigma_\beta^2 \sin^2\beta & 0 & 0 & 0 & 0 \\
-\sigma_\beta^2 \sin\beta\cos\beta & \sigma_\beta^2 \cos^2\beta & 0 & -\sigma_\beta^2 \cos^2\beta & -\sigma_\beta^2 \sin\beta\cos\beta & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\sigma_\beta^2 \sin\beta\cos\beta & -\sigma_\beta^2 \cos^2\beta & 0 & \sigma_\beta^2 \cos^2\beta & \sigma_\beta^2 \sin\beta\cos\beta & 0 & 0 & 0 & 0 \\
\sigma_\beta^2 \sin^2\beta & -\sigma_\beta^2 \sin\beta\cos\beta & 0 & \sigma_\beta^2 \sin\beta\cos\beta & \sigma_\beta^2 \sin^2\beta & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]$$

where σ_β^2 is the variance for β.

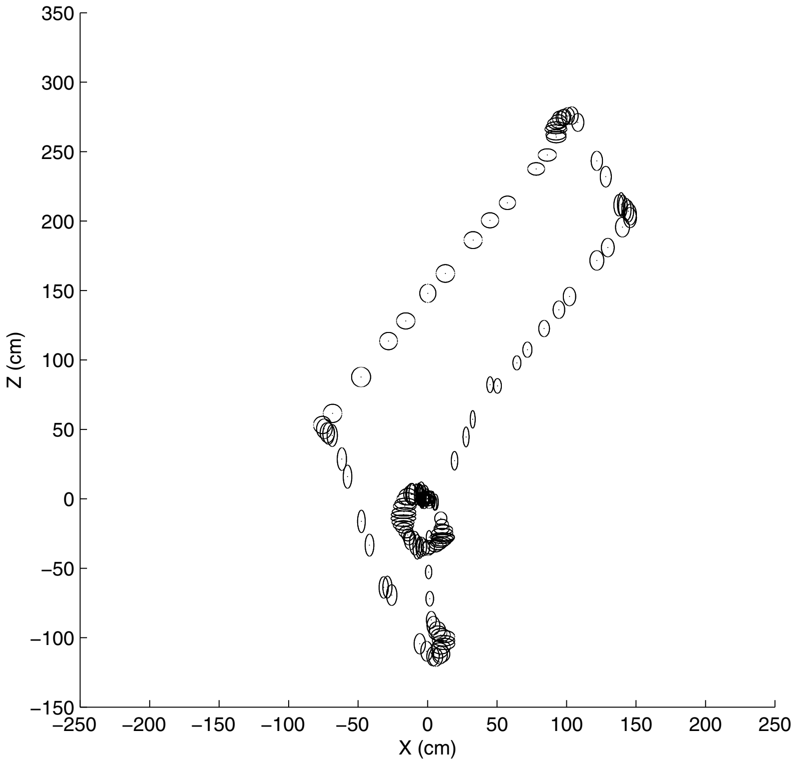


Fig. 13. Robot trajectory with uncertainty ellipses.

Using eq. (7), the resulting landmark position covariance Λ_β is

$$\begin{bmatrix} \substack{X^2\sigma_\beta^2 \sin^2\beta - 2XY\sigma_\beta^2 \sin\beta\cos\beta \\ +Y^2\sigma_\beta^2\cos^2\beta + \sigma_X^2\cos^2\beta \\ +2\sigma_{XY}^2\sin\beta\cos\beta + \sigma_Y^2\sin^2\beta} & \substack{(X^2-Y^2)\sigma_\beta^2\sin\beta\cos\beta - XY\sigma_\beta^2\cos^2\beta \\ +XY\sigma_\beta^2\sin^2\beta + (\sigma_Y^2-\sigma_X^2)\sin\beta\cos\beta \\ -\sigma_{XY}^2\sin^2\beta + \sigma_{XY}^2\cos^2\beta} & \sigma_{XZ}^2\cos\beta + \sigma_{YZ}^2\sin\beta \\ \substack{(X^2-Y^2)\sigma_\beta^2\sin\beta\cos\beta - XY\sigma_\beta^2\cos^2\beta \\ +XY\sigma_\beta^2\sin^2\beta + (\sigma_Y^2-\sigma_X^2)\sin\beta\cos\beta \\ -\sigma_{XY}^2\sin^2\beta + \sigma_{XY}^2\cos^2\beta} & \substack{X^2\sigma_\beta^2\cos^2\beta + 2XY\sigma_\beta^2\sin\beta\cos\beta \\ +Y^2\sigma_\beta^2\sin^2\beta + \sigma_X^2\sin^2\beta \\ -2\sigma_{XY}^2\sin\beta\cos\beta + \sigma_Y^2\cos^2\beta} & -\sigma_{XZ}^2\sin\beta + \sigma_{YZ}^2\cos\beta \\ \sigma_{XZ}^2\cos\beta + \sigma_{YZ}^2\sin\beta & -\sigma_{XZ}^2\sin\beta + \sigma_{YZ}^2\cos\beta & \sigma_Z^2 \end{bmatrix}$$

where (X, Y, Z) is the 3D landmark position in the current frame. Since this is the first transformation to be carried out, $\sigma_{XY}^2 = \sigma_{XZ}^2 = \sigma_{YZ}^2 = 0$ as the initial covariance of the observed position is a diagonal matrix.

Applying the roll transformation to the initial landmark position gives the transformed position, which is then used in the next stage together with this new landmark covariance.

A.2. Pitch Transformation

The 9×9 covariance matrix for the pitch transformation is

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\alpha^2\sin^2\alpha & -\sigma_\alpha^2\sin\alpha\cos\alpha & 0 & \sigma_\alpha^2\sin\alpha\cos\alpha & \sigma_\alpha^2\sin^2\alpha \\ 0 & 0 & 0 & 0 & -\sigma_\alpha^2\sin\alpha\cos\alpha & \sigma_\alpha^2\cos^2\alpha & 0 & -\sigma_\alpha^2\cos^2\alpha & -\sigma_\alpha^2\sin\alpha\cos\alpha \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\alpha^2\sin\alpha\cos\alpha & -\sigma_\alpha^2\cos^2\alpha & 0 & \sigma_\alpha^2\cos^2\alpha & \sigma_\alpha^2\sin\alpha\cos\alpha \\ 0 & 0 & 0 & 0 & \sigma_\alpha^2\sin^2\alpha & -\sigma_\alpha^2\sin\alpha\cos\alpha & 0 & \sigma_\alpha^2\sin\alpha\cos\alpha & \sigma_\alpha^2\sin^2\alpha \end{bmatrix}$$

where σ_α^2 is the variance for α.

Using eq. (7), the resulting landmark position covariance $\Lambda_{\beta\alpha}$ is

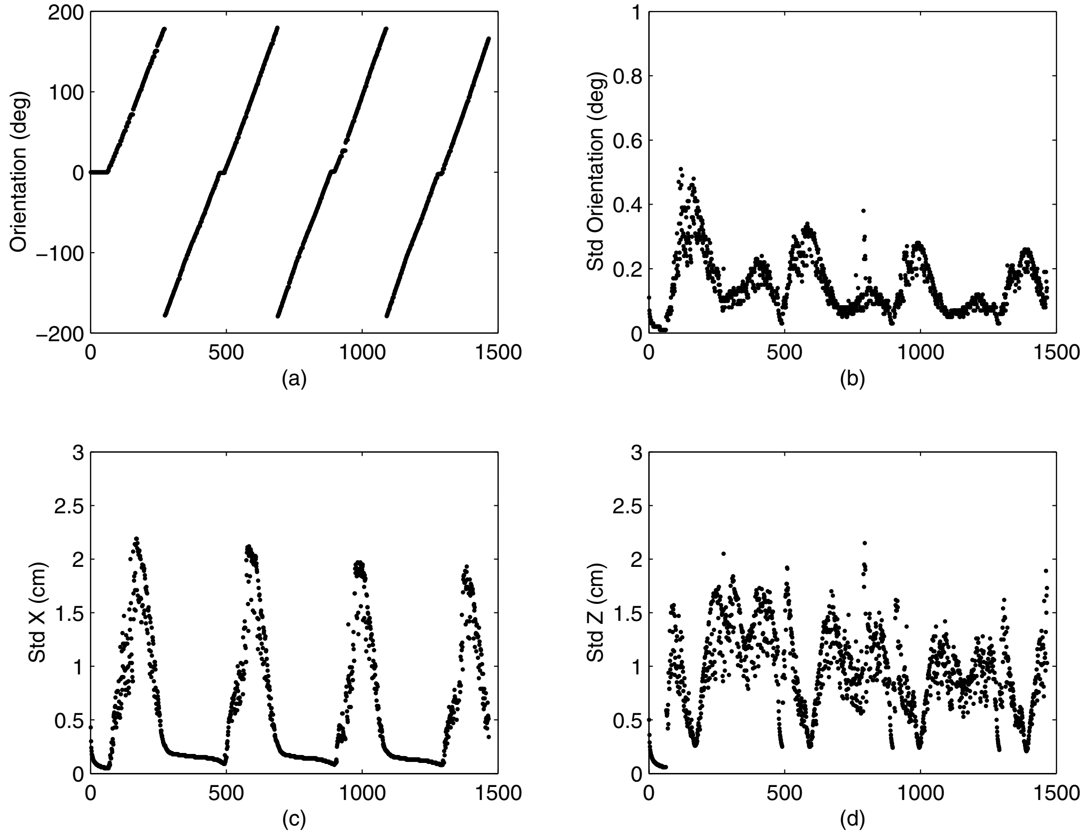


Fig. 14. Robot pose uncertainty over frames while the robot rotates repeatedly: (a) robot orientation; (b) robot orientation standard deviation; (c) robot pose uncertainty in the X direction; (d) robot pose uncertainty in the Z direction.

$$\begin{bmatrix} \sigma_X^2 & \sigma_{XY}^2\cos\alpha+\sigma_{XZ}^2\sin\alpha & -\sigma_{XY}^2\sin\alpha+\sigma_{XZ}^2\cos\alpha \\ \sigma_{XY}^2\cos\alpha+\sigma_{XZ}^2\sin\alpha & \begin{array}{c} Y^2\sigma_\alpha^2\sin^2\alpha-2YZ\sigma_\alpha^2\sin\alpha\cos\alpha \\ +Z^2\sigma_\alpha^2\cos^2\alpha+\sigma_Y^2\cos^2\alpha \\ +2\sigma_{YZ}^2\sin\alpha\cos\alpha+\sigma_Z^2\sin^2\alpha \end{array} & \begin{array}{c} (Y^2-Z^2)\sigma_\alpha^2\sin\alpha\cos\alpha-YZ\sigma_\alpha^2\cos^2\alpha \\ +YZ\sigma_\alpha^2\sin^2\alpha+(\sigma_Z^2-\sigma_Y^2)\sin\alpha\cos\alpha \\ -\sigma_{YZ}^2\sin^2\alpha+\sigma_{YZ}^2\cos^2\alpha \end{array} \\ \begin{array}{c} -\sigma_{XY}^2\sin\alpha+\sigma_{XZ}^2\cos\alpha \\ -\sigma_{YZ}^2\sin^2\alpha+\sigma_{YZ}^2\cos^2\alpha \end{array} & \begin{array}{c} (Y^2-Z^2)\sigma_\alpha^2\sin\alpha\cos\alpha-YZ\sigma_\alpha^2\cos^2\alpha \\ +YZ\sigma_\alpha^2\sin^2\alpha+(\sigma_Z^2-\sigma_Y^2)\sin\alpha\cos\alpha \end{array} & \begin{array}{c} Y^2\sigma_\alpha^2\cos^2\alpha+2YZ\sigma_\alpha^2\sin\alpha\cos\alpha \\ +Z^2\sigma_\alpha^2\sin^2\alpha+\sigma_Y^2\sin^2\alpha \\ -2\sigma_{YZ}^2\sin\alpha\cos\alpha+\sigma_Z^2\cos^2\alpha \end{array} \end{bmatrix}$$

where (X, Y, Z) is the transformed 3D landmark position after the roll transformation and σ_X^2, σ_Y^2, σ_Z^2, σ_{XY}^2, σ_{XZ}^2 and σ_{YZ}^2 are from the covariance matrix Λ_β above. Applying the pitch transformation gives the transformed landmark position, which is then used in the next stage together with this new landmark covariance.

A.3. Yaw Transformation

The 9 × 9 covariance matrix for the yaw transformation is

$$\begin{bmatrix} \sigma_\theta^2\sin^2\theta & 0 & -\sigma_\theta^2\sin\theta\cos\theta & 0 & 0 & 0 & \sigma_\theta^2\sin\theta\cos\theta & 0 & \sigma_\theta^2\sin^2\theta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\sigma_\theta^2\sin\theta\cos\theta & 0 & \sigma_\theta^2\cos^2\theta & 0 & 0 & 0 & -\sigma_\theta^2\cos^2\theta & 0 & -\sigma_\theta^2\sin\theta\cos\theta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma_\theta^2\sin\theta\cos\theta & 0 & -\sigma_\theta^2\cos^2\theta & 0 & 0 & 0 & \sigma_\theta^2\cos^2\theta & 0 & \sigma_\theta^2\sin\theta\cos\theta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma_\theta^2\sin^2\theta & 0 & -\sigma_\theta^2\sin\theta\cos\theta & 0 & 0 & 0 & \sigma_\theta^2\sin\theta\cos\theta & 0 & \sigma_\theta^2\sin^2\theta \end{bmatrix}$$

where σ_θ^2 is the variance for θ.

Using eq. (7), the resulting landmark position covariance $\Lambda_{\beta\alpha\theta}$ is

$$\begin{bmatrix} \begin{array}{c} X^2\sigma_\theta^2 \sin^2\theta - 2XZ\sigma_\theta^2 \sin\theta\cos\theta \\ +Z^2\sigma_\theta^2 \cos^2\theta + \sigma_X^2 \cos^2\theta \\ +2\sigma_{XZ}^2 \sin\theta\cos\theta + \sigma_Z^2 \sin^2\theta \end{array} & \sigma_{XY}^2 \cos\theta + \sigma_{YZ}^2 \sin\theta & \begin{array}{c} (X^2 - Z^2)\sigma_\theta^2 \sin\theta\cos\theta - XZ\sigma_\theta^2 \cos^2\theta \\ +XZ\sigma_\theta^2 \sin^2\theta + (\sigma_Z^2 - \sigma_X^2)\sin\theta\cos\theta \\ -\sigma_{XZ}^2 \sin^2\theta + \sigma_{XZ}^2 \cos^2\theta \end{array} \\ \sigma_{XY}^2 \cos\theta + \sigma_{YZ}^2 \sin\theta & \sigma_Y^2 & -\sigma_{XY}^2 \sin\theta + \sigma_{YZ}^2 \cos\theta \\ \begin{array}{c} (X^2 - Z^2)\sigma_\theta^2 \sin\theta\cos\theta - XZ\sigma_\theta^2 \cos^2\theta \\ +XZ\sigma_\theta^2 \sin^2\theta + (\sigma_Z^2 - \sigma_X^2)\sin\theta\cos\theta \\ -\sigma_{XZ}^2 \sin^2\theta + \sigma_{XZ}^2 \cos^2\theta \end{array} & -\sigma_{XY}^2 \sin\theta + \sigma_{YZ}^2 \cos\theta & \begin{array}{c} X^2\sigma_\theta^2 \cos^2\theta + 2XZ\sigma_\theta^2 \sin\theta\cos\theta \\ +Z^2\sigma_\theta^2 \sin^2\theta + \sigma_X^2 \sin^2\theta \\ -2\sigma_{XZ}^2 \sin\theta\cos\theta + \sigma_Z^2 \cos^2\theta \end{array} \end{bmatrix}$$

where (X, Y, Z) is the transformed 3D landmark position after the pitch transformation, and σ_X^2, σ_Y^2, σ_Z^2, σ_{XY}^2, σ_{XZ}^2 and σ_{YZ}^2 are from the covariance matrix $\Lambda_{\beta\alpha}$ above. Applying the yaw transformation gives the transformed landmark position, which is then used in the next stage together with this new landmark covariance.

A.4. Translational Transformation

Finally, we have the two translational components, x and z, as the camera system is mounted on the robot at a fixed height. The final covariance for each landmark point is therefore

$$\Sigma_{new} = \Lambda_{\beta\alpha\theta} + \begin{bmatrix} \sigma_x^2 & 0 & \sigma_{xz}^2 \\ 0 & 0 & 0 \\ \sigma_{xz}^2 & 0 & \sigma_z^2 \end{bmatrix}.$$

Acknowledgments

Our work has been 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.

References

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

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

Borenstein, J., and Feng, L. 1996. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation*, 12(5).

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

Borenstein, J. 1995. Internal correction of dead-reckoning errors with the compliant linkage vehicle. *Journal of Robotic Systems*, 12(4):257–273.

Castellanos, J. A., Montial, J. M. M., Neira, J., and Tardos, J. D. 1999. Sensor influence in the performance of simultaneous mobile robot localization and map building. In *Proceedings of 6th International Symposium on Experimental Robotics*, pp. 203–212, Sydney, Australia.

Castellanos, J. A., Devy, M., and Tardos, J. D. 2000. Simultaneous localisation and map building for mobile robots: A landmark-based approach. In *Proceedings of IEEE International Conference on Robotics and Automation: Workshop on Mobile Robot Navigation and Mapping*, San Francisco, USA, April 2000.

Crowley, J. L. 1989. Asynchronous control of orientation and displacement in a robot vehicle. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1277–1282, Scottsdale, Arizona, May 1989.

Davison, A. J. 1998. *Mobile Robot Navigation Using Active Vision*. PhD thesis, Department of Engineering Science, University of Oxford.

Dellaert, F., Burgard, W., Fox, D., and Thrun, S. 1999. 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.

Dissanayake, M. W. M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., and Csorba, M. 2001. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241.

Everett, H. R. 1995. *Sensors for Mobile Robots*. A.K. Peters, Ltd, Wellesley, MA.

Gelb, A. 1984. *Applied Optimal Estimation*. MIT Press.

Guivant, J., and Nebot, E. 2001. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, June 2001.

Gutmann, J., and Konolige, K. 1999. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, California, November 1999.

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

Harris, C. 1992. Geometry from visual motion. In A. Blake and A. Yuille, eds., *Active Vision*, pp. 264–284. MIT Press.

Jensfelt, P., Wijk, O., Austin, D. J., and Andersson, M. 2000. 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.

Knight, J., Davison, A., and Reid, I. 2001. Towards constant time slam using postponement. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Maui, Hawaii, October 2001.

Leonard, J. J., and Durrant-Whyte, H. F. 1991. Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'91)*, pp. 1442–1447, New York, USA.

Leonard, J. J., and Feder, H. J. S. 1999. A computational efficient method for large-scale concurrent mapping and localization. In *9th International Symposium of Robotics Research*, London, Springer-Verlag.

Lowe, D. G. 1992. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122.

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

Se, S., Lowe, D., and Little, J. 2001a. Local and global localization for mobile robots using visual landmarks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 414–420, Maui, Hawaii, October 2001.

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

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

Smith, R., Self, M., and Cheeseman, P. 1987. A stochastic map for uncertain spatial relationships. In *4th International Symposium on Robotics Research*. MIT Press.

Thrun, S., Burgard, W., and Fox, D. 1998. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots (joint issue)*, 31(5):1–25.

Thrun, S., Burgard, W., and Fox, D. 2000. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, April 2000.

Williams, S. B., Newman, P., Dissanayake, G., and Durrant-Whyte, H. 2000. Autonomous underwater simultaneous localisation and map building. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, San Francisco, USA, April 2000.