Bearing-only SLAM using SIFT-based Object Recognition in Indoor Environments

Seo-Yeon Hwang
Dept. of Mech. Eng.
Korea University
Seoul, Korea
etoile02@korea.ac.kr

Yong-Ju Lee
Dept. of Mech. Eng.
Korea University
Seoul, Korea
yongju_lee@korea.ac.kr

Byung-Doo Yim
Dept. of Mechatronics
Korea University
Seoul, Korea
ybd1021@korea.ac.kr

Jae-Bok Song
Dept. of Mech. Eng.
Korea University
Seoul, Korea
jbsong@korea.ac.kr

Abstract − Bearing-only SLAM is the process of building a map for unknown environments and localizing a robot relative to this map simultaneously using only a mono camera mounted on the robot. For bearing-only SLAM, landmarks should be continuously observed. If the robot fails to track the features, pose estimation of the robot becomes inaccurate. Most existing algorithms for bearing-only SLAM use corners or lines as landmarks, but they are hard to track continuously because they are not robust to the difference in the distance between the robot and landmarks while the robot moves. In this paper, bearing-only SLAM in indoor environments using the relative orientation of an object recognition based on the SIFT (Scale Invariant Feature Transform) algorithm is proposed. Experimental results show that the estimated robot pose is accurate enough to be used for building an accurate map.

Keywords − Mobile robot, Bearing-only SLAM, SIFT, EKF (extended Kalman filter).

1. Introduction

Using a SLAM (Simultaneous Localization and Mapping) technique, a mobile robot can build a map with on-board sensors in the unknown environment while it is localizing itself relative to the partially built map. SLAM is one of the most fundamental and challenging issues in the field of mobile robotics and much research has been done for the last two decades.

An encoder, which is the most basic sensor for a mobile robot, is rather inaccurate because of the misalignment of wheels and the wheel slippage on the ground. Since the odometric error accumulates as the robot moves, it is difficult to estimate the accurate robot pose only with this sensor. Therefore, other types of sensors such as a range sensor and a vision sensor are employed together with the encoder so as to compensate for the pose error caused by the encoder. The information from a range sensor can be easily processed to extract the features, but these features are usually limited to lines and corners. On the other hand, a vision sensor can provide much more information than a range sensor, but more complicated process is required to extract the features. In recent years, therefore, most research on SLAM has been conducted using vision sensors.

Most existing SLAM algorithms use the range and bearing information of the landmarks in the environment. In order to get the range of a landmark with a vision sensor, we use either the stereo vision sensor or the mono vision sensor together with range sensors. But in the bearing-only SLAM, a single mono camera can be used because only relative orientations of landmarks from the robot are used for SLAM.

Most bearing-only SLAM schemes employ EFK (extended Kalman filter) to estimate the robot pose [1, 2], and use line features [3] or corner features [4-6] in the environments as a natural landmark. Although line features exist widely in indoor environments, but they are easily affected by several external sources such as image noise and scale change. Therefore, it is not easy to extract the line features in a robust fashion in real environments.

Indoor environments usually possess more corner features than line features, and corners are easier to extract. Davison implemented the real-time bearing-only SLAM using corner extraction with a wearable camera [5]. In this research motion of the camera was predicted by the corners extracted from the camera images of the camera, which was the only sensor used for measurement. In order to estimate the human pose in real-time, the processing time of extracting corners must be faster than the motion speed of the camera, thus requiring fast image processing.

Consistent tracking of the feature is an important factor for the accuracy of localization in the bearing-only SLAM. However, continuous estimation of the robot pose becomes difficult when the lines or corners are used, because they are not robust to changes in scale and rotation.

To cope with the above problems, the bearing-only SLAM using the center points of objects which are extracted by using the SIFT (Scale Invariant Feature Transform) features is conducted in this paper. It is possible to keep track of the same feature more consistently than the corners because the SIFT features are robust to changes in scale and rotation. Furthermore, the robot can be controlled to move in front of the target objects because the robot has the information of the objects. However, the process of extracting the SIFT features requires long computation time. In case of using

only a vision sensor [5], the computing time for extracting the landmarks has a great impact on the performance of SLAM. However, many types of feature extraction methods can be employed in conducting SLAM of a mobile robot because the robot motion and thus the landmark position can be predicted roughly by the encoders while recognizing objects.

To decrease the uncertainty of the feature, the robot observes the feature from various angles, which requires much computation time. To reduce this computation time, the particle filter based feature initialization method was adopted in this paper.

The remainder of this paper is organized as follows. Section 2 introduces object recognition using the SIFT method. Section 3 presents the application procedure of EKF (Extended Kalman Filter) based bearing only SLAM using the SIFT-based object recognition for feature extraction. Section 4 describes how to initialize landmarks using a particle filter. Finally, section 5 and 6 present experimental results and conclusions.

2. SIFT

SIFT (Scale Invariant Feature Transform) is one of the image recognition methods used to extract the feature points which are invariant to the image scaling, rotation and viewpoint. A BBF (Best-Bin First) algorithm [7] is used in the matching process between the extracted feature points and the feature points stored in the database. A brief introduction to these algorithms will be given below.

2.1 SIFT

Scale-space extrema detection: A Gaussian pyramid is created with respect to the input image, and the point which has a maximum or minimum value is found by a difference-of-Gaussian function.

Extraction of stable keypoints: To extract stable keypoints, the Taylor series expansion is used to remove unstable points. The unstable keypoints along the object edges are eliminated by the Harris corner detector.

Extraction of dominant keypoints: A dominant direction with respect to each keypoint is obtained. The orientation histogram quantizes 360 degrees of orientation into 36 intervals. The peaks in the histogram are selected as the dominant directions of the keypoints. Any other local peaks which are within 80% of the highest peak are also used to create the dominant direction of a keypoint. Extracted dominant directions are robust to 2D rotation of images, but 2D rotation of the camera image does not occur because only the yawing motion exists in navigation of a mobile robot. Thus, the process which makes keypoints robust to the 2D rotation is excluded to minimize the processing time.

Keypoint descriptor: Orientation of 360 degrees is quantized into 8 units with the keypoint as a center. SIFT divides the area, in which the feature vector is extracted, into the 4 x 4 local regions and creates an angle histogram of 8 discrete orientations for each region. Thus, each keypoint possesses a feature vector with 128 elements. This vector is normalized to be robust to illumination change and then renormalized by holding the values to be no larger than 0.2.

BBF (Best-Bin-First) Priority queue is adopted in BBF to make up for the weak points of the KD-tree method and improve its processing time. The keypoint matching of the KD-tree tends to become less accurate as the number of vector elements increases. Since this number is 128 in SIFT, the matching results based on the KD-tree scheme is relatively inaccurate. The accuracy and processing time for keypoint matching can be improved by using priority queue which detects the nodes close to the feature vectors.

2.2 Extraction of Center Point

Since a number of keypoints usually exist in an object, a representative point is needed to compute the relative distance and angle from the robot. For this purpose, the center point of an object is extracted. The center point should be extracted accurately to reduce the localization error. Therefore, the affine transformation is employed to represent the geometrical relation between the recognized object and stored object in the database.

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{1}$$

where the vector $[x_i, y_i]^T$ is the keypoint stored in the database, and $[u_i, v_i]^T$ is the keypoint extracted in the current image. The vector $[t_x, t_y]^T$ represents the translation and m_i (i = 1, .., 4) are the parameters of the affine transform associated with the 3D rotation. These 6 parameters describe the geometrical relation between the recognized object and the object stored in the database.

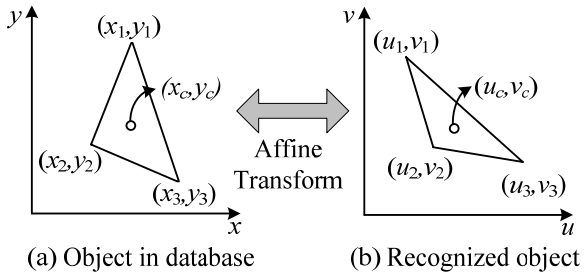


Fig. 1. Example of affine transformation

For example, for given 3 pairs of (x_i, y_i) and (u_i, v_i) shown in Fig. 1, the 6 parameters can be obtained by solving the following equation.

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots\cdots & & & \\ & & \cdots\cdots & & & \\ & & \cdots\cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ . \\ . \end{bmatrix} \quad \text{or} \quad \mathbf{Ap} = \mathbf{b} \tag{2}$$

Once the parameters are extracted, the center point (u_c, v_c) of the recognized object can be computed from the

center point (x_c, y_c) of the object stored in the database by Eq. (1). If the number of matched keypoints is more than 3, the parameters are computed by using the pseudo inverse of **A** as follows:

$$\mathbf{p} = [\mathbf{A}^T\ \mathbf{A}]^{-1}\mathbf{A}^T\mathbf{b} \tag{3}$$

3. Bearing-only SLAM

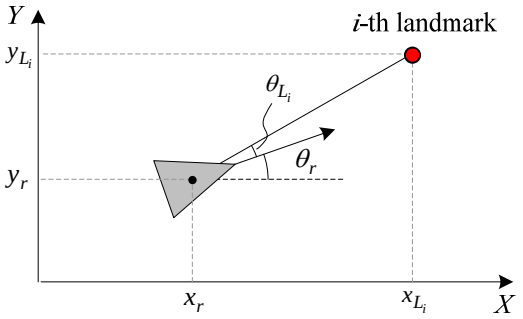


Fig. 2. Robot and landmark in the environment.

The EKF (Extended Kalman Filter) algorithm is adopted to handle nonlinearities involved in the robot motion. The following state vector is defined.

$$\mathbf{X} = [\mathbf{X}_r^T, \mathbf{X}_{L_1}^T, \cdots, \mathbf{X}_{L_N}^T]^T \tag{4}$$

where $\mathbf{X}_r = [x_r, y_r, \theta_r]^T$ represents the robot pose and $\mathbf{X}_{L_i} = [x_{L_i}, y_{L_i}]^T$ denotes the i-th landmark, where N is the number of landmarks.

The EKF algorithm based on Bayes filtering [8] consists of the prediction stage and update stage. At the prediction stage, the state vector $\hat{\mathbf{X}}_k$ and its covariance matrix $\mathbf{P}_k$ at time k are calculated from those at time k-1 and the encoder reading $\mathbf{u}_k$ as follows:

$$\hat{\mathbf{X}}_k^- = f(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_k) \tag{5}$$

$$\mathbf{P}_k^- = \nabla\mathbf{F}_x\mathbf{P}_{k-1}\nabla\mathbf{F}_x^T + \nabla\mathbf{F}_u\mathbf{Q}\nabla\mathbf{F}_u^T \tag{6}$$

$$\nabla\mathbf{F}_x = \frac{\partial f}{\partial \mathbf{X}},\ \nabla\mathbf{F}_u = \frac{\partial f}{\partial \mathbf{u}} \tag{7}$$

where **Q** represents the covariance matrix of the process noise, $\mathbf{F}_x$ and $\mathbf{F}_u$ are the Jacobian matrix of the nonlinear function f with respect to the state and input, respectively. Note that the superscript "-" indicates the state before the measurement at time k is taken.

If the robot observes a feature, it compares this feature to the features in the state vector **X**. If it turns out to be a new feature, the feature initialization process starts. If it is found to be one of the existing features, EKF conducts the update stage. The bearing of a landmark predicted on the sensor frame can be obtained by the following sensor model based on the system state.

$$\hat{\mathbf{Z}}_k = h(\hat{\mathbf{X}}_k^-) \tag{8}$$

where h represents the sensor model used in this paper. The sensor model performs the transform from the world frame to the robot frame with respect to the feature information to compare the actual observations. The sensor model for all the landmarks is expressed by

$$\begin{bmatrix} \theta_{L_1,k} \\ \vdots \\ \theta_{L_N,k} \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\dfrac{y_{L_1,k} - y_{r,k}}{x_{L_1,k} - x_{r,k}}\right) - \theta_{r,k} \\ \vdots \\ \tan^{-1}\left(\dfrac{y_{L_N,k} - y_{r,k}}{x_{L_N,k} - x_{r,k}}\right) - \theta_{r,k} \end{bmatrix} \tag{9}$$

At the update stage, the state vector and its covariance matrix **P** at time k are updated as follows:

$$\hat{\mathbf{X}}_k = \hat{\mathbf{X}}_k^- + \mathbf{K}_k(\mathbf{Z}_k - \hat{\mathbf{Z}}_k) \tag{10}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \tag{11}$$

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{V}_k\mathbf{R}_k\mathbf{V}_k^T)^{-1} \tag{12}$$

$$\mathbf{H} = \frac{\partial h}{\partial \mathbf{X}} \tag{13}$$

$$\mathbf{V} = \frac{\partial h}{\partial v} \tag{14}$$

where **K** represents the Kalman gain matrix, and **H** and **V** are the Jacobian matrices of the sensor model with respect to the state vector and the sensor noise, respectively. If none of landmarks are matched, the uncertainties of landmarks are kept unchanged.

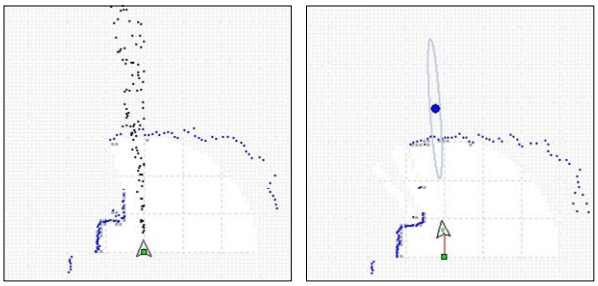
$$\hat{\mathbf{X}}_k = \hat{\mathbf{X}}_k^- \tag{15}$$

$$\mathbf{P}_k = \mathbf{P}_k^- \tag{16}$$

In this case, only the robot pose is calculated by the motion model and the uncertainty of the robot pose increases.

4. Landmark Initialization

When a newly observed feature is registered as a new landmark, it takes too much time to estimate the accurate pose of a landmark because the initial uncertainty of the landmark is very large. Thus a particle filter is adopted to minimize the time to determine the variance of uncertainty of the landmark. If a new landmark is observed, samples are drawn in a region corresponding to the error range of observation with the direction of the observation as an axis, as depicted in Fig. 3(a). In this paper, the maximum distance between the robot and the objects is assumed to be 7m and 1,000 samples are used.



(a) (b)

Fig. 3. Landmark initialization using particle filter: (a) sample distribution, and (b) registration of landmark after convergence of samples.

Uncertainty of a landmark decreases as the robot observes the same landmark from different angles while the robot is moving. Since the landmark is assumed to be stationary, its motion model is not considered and only the sensor model for the landmark is used, as shown in Fig. 4. As the robot moves, the initially distributed samples converge according to the probability of samples used in the particle filter. The sensor model used in this paper is

$$\begin{cases} w = 1 - \dfrac{|\theta_{sample}|}{\theta_{error}} & \text{for } |\theta_{sample}| \le \theta_{error} \\ w = 0 & \text{for } |\theta_{sample}| > \theta_{error} \end{cases} \quad (17)$$

where w is the importance factor multiplied by the probability of each sample. The angles θ_{sample} and θ_{error} represent the relative angle of the sample and the error of the relative angle observed by the camera, respectively.

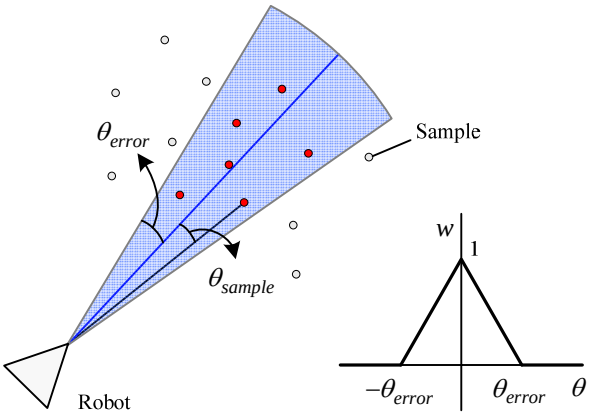


Fig. 4. Sensor model used in the particle filter.

If the variance of the sample pose goes below the predetermined threshold, then the feature initialization process is completed and the feature is registered to a new feature used for localization of a robot. The mean and variance of a sample becomes those of the landmark. The landmark whose uncertainty decreases to a certain level is registered at the prediction stage of EKF as follows:

$$\hat{\mathbf{X}}_{new}^{-} = \begin{bmatrix} \left(\hat{\mathbf{X}}_{old}^{-}\right)^T & x_{L,new} & y_{L,new} \end{bmatrix}^T \quad (18)$$

$$\mathbf{P}_{new}^{-} = \begin{bmatrix} \mathbf{P}_{old}^{-} & 0 \\ 0 & \mathbf{P}_{L,new} \end{bmatrix} \quad (19)$$

5. Experiments

Various experiments have been conducted with the robot equipped with an IR scanner (Hokuyo PBS-03JN) and a mono camera (Logitech Quickcam with a field of view of 60°). The camera image was converted into black and white image with 320 x 240 pixels. The experimental environment was constructed to mimic the living room of 8m x 10m in size with several pieces of furniture, as shown in Fig. 5(a). The environment was modeled by cells of 10cm x 10cm. Fig. 5(b) represents visual objects used as visual landmarks. The grid map resulting from the proposed algorithm was created by the IR scanner and only the data less than 3m are used because the data exceeding 3m are found to be incorrect.

In the first experiment, as shown in Fig. 6, a map was generated only by the odometry information. During the 3 laps, the map was rotated about 80° when compared with the Fig. 6(a) because of the inherent encoder error mainly due to the slippage between the robot and the floor.

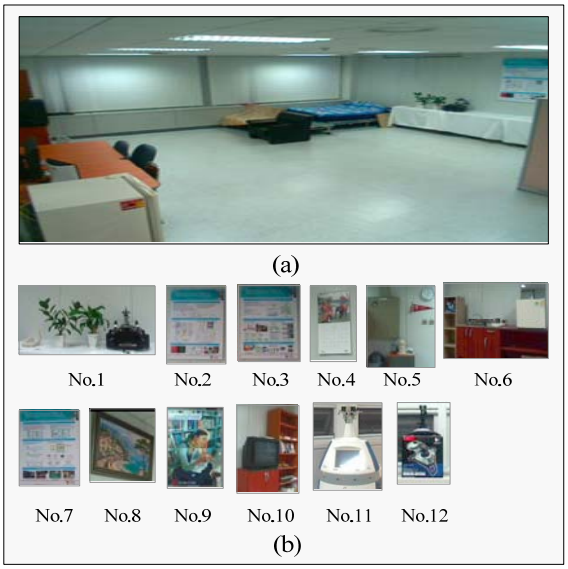


Fig. 5. (a) Experimental environment, and (b) visual objects stored in the database.

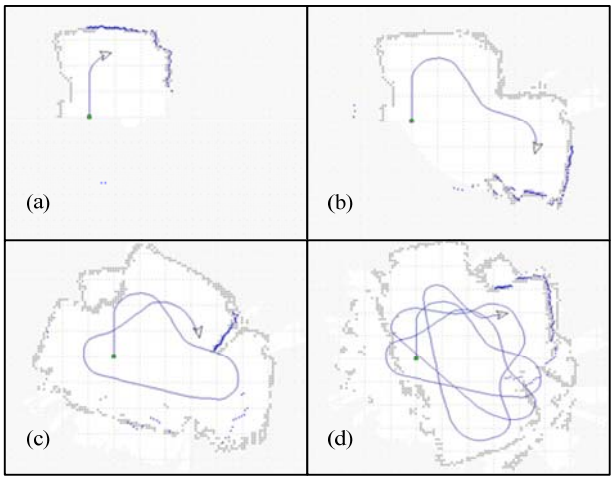


Fig. 6. Map building process only based on the odometry information.

Fig. 7. Scale-invariant properties of SIFT-based object recognition.

The next experiment used the SIFT-based object recognition to compensate for the odometric error. The recognized objects were consistently tracked regardless of its scale change, as shown in Fig. 7.

Figure 8 shows the result of bearing-only SLAM using the center points of visual objects recognized by SIFT features. A solid line represents the path of the estimated robot and a dashed line represents the path generated by the encoder. The blue circles represent the mean value of the uncertainty ellipses. Compared with the actual positions, the accuracy of the estimated landmark positions can be determined. The robot tracked the landmarks stably and the covariance of the landmarks was reduced well. When the robot observed the previous landmarks, its pose was corrected accurately. This could be confirmed by the grid map which was constructed during navigation and was not distorted.

The complexity of the update in EKF is of order N^2, where N is the number of landmarks registered in EKF. If the number of landmarks exceeds 100, the EKF update time begins to grow rapidly [5]. The number of landmarks was 12 in this experiment, and it took 1ms in EKF for the update of the landmarks.

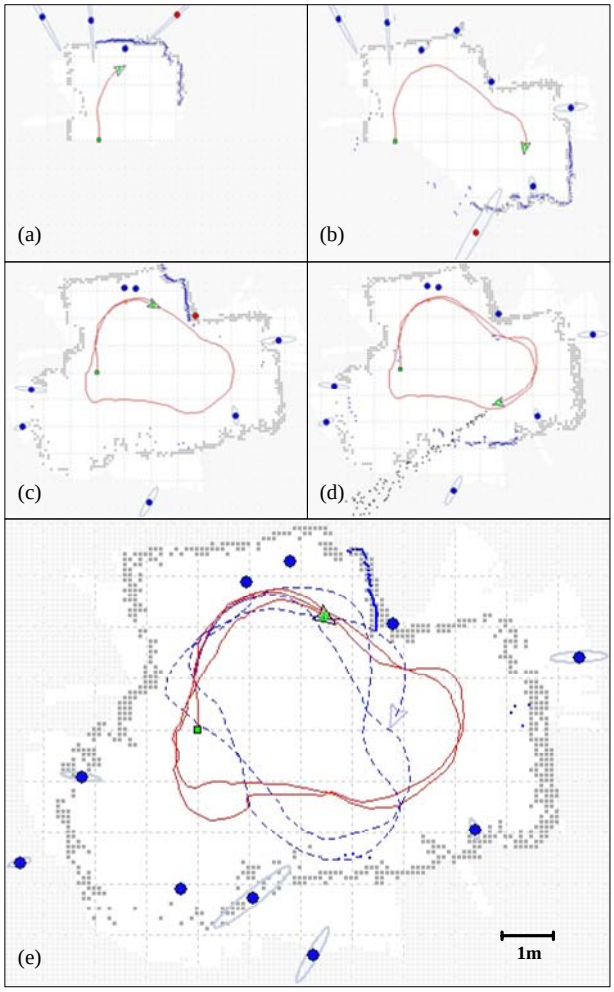


Fig. 8. The experimental result: (a)-(c) a map generated during the first lap, (d) initializing process of a new landmark, and (e) the final result.

6. Conclusions

This paper proposed the bearing-only SLAM algorithm using a mono camera and bearings from the SIFT-based visual object recognition. A particle filter was used for fast landmark initialization and EKF was used to fuse the encoder and the visual information and estimate the robot pose and the landmark positions. The validity of the proposed SLAM method investigated by various experiments, and the following conclusions were drawn.

1. In mobile robot navigation, the landmark was tracked consistently regardless of scale change, and the uncertainty of the landmarks was reduced greatly by using the object recognition based on SIFT. Thus, the robot pose was estimated accurately.

2. Usually, the computational load of the EKF increases rapidly when the number of landmarks exceeds 100. Since the number of visual objects in indoor environments is limited, the proposed algorithm can be applied to indoor navigation in real-time.

Acknowledgements

This research was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

References

[1] A. Costa, G. Kantor, and H. Choset, "Bearing-only Landmark Initialization with Unknown Data Association," *Proc. of IEEE Int. Conf. on Robotics & Automation*, pp. 1764-1770, 2004.

[2] G. Fang, G. Dissanayake, N. M. Kwok and S. Huang, "Near Minimum Time Path Planning for Bearing-Only Localisation and Mapping," *Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005.

[3] N. M. Kwok and G. Dissanayake, "An Efficient Multiple Hypothesis Filter for Bearing-Only SLAM," *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 736-741, 2004.

[4] T. Lemaire, S. Lacroix and J. Sola, "A practical 3D Bearing-Only SLAM algorithm," *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005.

[5] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," *Proc. of the ninth Int. Conf. on Computer Vision*, 2003.

[6] J. Montiel and A. Davison, "A Visual Compass based on SLAM," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2006.

[7] David. G. Lowe, "Distinctive image features from scale invariant keypoints," *Int. Journal of Computer Vision*, vol. 60, no 2, pp. 91-110, 2004.

[8] S. Thrun, W. Burgard and D. Fox, "Probability Robotics," *MIT Press*, 2005.