Proceedings of 2013 IEEE
International Conference on Mechatronics and Automation
August 4 - 7, Takamatsu, Japan

Data Association and Map Management for Robot SLAM using Local Invariant Features

Yin-Tien Wang and Ying-Chieh Feng
Department of Mechanical and Electro-Mechanical Engineering
Tamkang University
New Taipei City, Taiwan 25137
{ytwang@mail & 696372076@s96}.tku.edu.tw

Abstract **-** To build a persistent map with visual landmarks is one of the most important steps for implementing the visual simultaneous localization and mapping (SLAM). The corner detector is a common method utilized to detect visual landmarks for constructing a map of the environment. However, due to the scale-variant characteristic of corner detection, extensive computational cost is needed to recover the scale and orientation of corner features in SLAM tasks. The purpose of this paper is to build the map using a local invariant feature detector, namely speeded-up robust features (SURF), to detect scale- and orientation-invariant features as well as provide a robust representation of visual landmarks for SLAM. The procedures of detection, description and matching of regular SURF algorithms are modified in this paper in order to provide a robust data-association of visual landmarks in SLAM. Furthermore, the effective method of map management for SURF features in SLAM is also designed to improve the accuracy of robot state estimation.

Index Terms **-** *Robot Mapping, Local Invariant Feature Detectors, Speeded-Up Robust Features (SURF), Simultaneous Localization and Mapping (SLAM)*.

I. INTRODUCTION

To build a persistent map with robust image features (landmarks) is an important step for implementing the visual simultaneous localization and mapping (SLAM). The image features must have the properties of high repeatability and unique description to be successfully detected at each time step. The research in this paper focuses on the development of algorithms to build a persistent map for robot visual SLAM using local invariant feature detectors. Furthermore, the characteristic of the environment is represented using a sparse but persistent map for real-time implementation in SLAM. The spare representation has the advantage of reducing the computational cost for information update in SLAM tasks using extended Kalman filter (EKF).

Harris corner detector [1] is the most popular method for image feature detection in visual SLAM task [2-3]. This method detects image corners or point positions by investigating the eigenvalues of the second moment matrix. The Harris corner detector is a simple algorithm and is easy to implement for robot SLAM task. However, it is difficult to track the corner features robustly when the camera is moving. When the distance and angle of camera viewpoint is changed, the scale and orientation of the corner feature will be changed and then result in the failure of image feature tracking. Therefore, more efforts are needed to recover the scale and orientation of image features [2]. On the other hand, the detection method of local invariant features [4-5] can automatically resolve above-mentioned problems and provide a robust representation method for scale- and orientation-invariant features in SLAM task [6].

Lindeberg proposed the concept of automatic scale selection to overcome the disadvantage of Harris corner [4]. He established a Hessian matrix whose elements are the convolution of the image and Laplacian of Gaussians (LoG). Then the local invariant features can be detected by investigating the determinant of the Hessian matrix. Lindeberg's method has the advantages of stability and high repeatability. However, the shortcoming of the local invariant feature detectors is computational inefficient. Several tactics in later literature have been proposed to improve the processing speed. Lowe [5] replaced LoG by Difference of Gaussians (DoG) and reduced the complexity of image convolution. In order to improve the computational performance, Bay *et al*. [7] further utilized the concepts of box filter and integral image [8] for the detection of local invariant features, called speeded-up robust features (SURF). In the literature, it was suggested that SURF is superior to other methods for detection and representation of image point features [9]. However, the computational cost of SURF method in visual SLAM was not evaluated. Many researchers have applied local invariant features in robot localization and mapping tasks. Karlsson *et al.* [6] developed visual SLAM using scale-invariant feature transform (SIFT) which is a method for detecting local invariant features proposed by Lowe [5]. Murillo *et al*. [10] utilized SURF to replace the SIFT method for image feature detection and recognition. The results showed that the efficiency for robot visual localization has been improved. Wu and Fu [11] used SURF to build the frontal system for visual SLAM and provide stable static landmarks. However, real-time implementation and performance evaluation of SURF-based SLAM was no described in these researches.

In this study, we propose a novel algorithm for map building using the SURF method. The feature detection, description, and recognition procedures are investigated and modified to provide the robust data-association of visual

978-1-4673-5560-5/13/$31.00 ©2013 IEEE

landmarks. Meanwhile, experiments on an actual system are carried out in static SLAM scenes to validate the proposed algorithm. In the system, a binocular vision is utilized as the only sensing device to implement the visual SLAM tasks. The remaining sections of the paper are organized as follows: The robot motion model and measurement model in SLAM tasks is firstly described. Secondly, the proposed modifications of SURF method as well as the procedures of data association and map management are presented. Thirdly, experimental results are demonstrated to support the proposed algorithms. Some concluding remarks are discussed in the last section.

II. Robot SLAM

When the robot performs SLAM tasks, the states of the robot and landmarks in the environment are estimated based on the measurement information. The state sequence of a system at time step k can be expressed as

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, w_{k-1}) \tag{1}$$

where $\mathbf{x}_k$ is the state vector; $\mathbf{u}_k$ is the input; w_k is the process noise. When perform SLAM tasks using a camera, the state vector contains the states of the camera and landmarks,

$$\mathbf{x} = [\mathbf{x}_C^T, \boldsymbol{M}^T]^T = [\mathbf{x}_C^T, \mathbf{m}_1^T, \mathbf{m}_2^T, \cdots, \mathbf{m}_j^T]^T \tag{2}$$

Where $\mathbf{x}_C = [r^T, \phi^T, \mathrm{v}^T, \omega^T]^T$ denotes the camera coordinates in world frame, $\mathbf{m}_j$ represent the jth landmark in the environment map $\boldsymbol{M}$. The objective of the robot SLAM tasks is recursively estimating state $\mathbf{x}_k$ of the target according to the measurement $\mathbf{z}_k$ at k,

$$\mathbf{z}_k = g(\mathbf{x}_k, v_k) \tag{3}$$

where v_k is the measurement noise. A binocular vision system (Figure 1) is the only sensing device considered in the recursive state estimation algorithm. At time $t=k$, the vectors of measurement $\mathbf{z}_k$ and the ith observed image feature are, respectively,

$$\mathbf{z}_k = [\mathbf{z}_{1k}^T\ \mathbf{z}_{2k}^T \cdots \mathbf{z}_{mk}^T]^T ; \quad \mathbf{z}_{ik} = \begin{bmatrix} I_{ix} \\ I_{iy} \end{bmatrix} \tag{4}$$

where i =1, 2, …, m; m is the number of measurements at time k, and (I_x, I_y) are the pixel coordinates of a feature in the image plane. The perspective projection [12] is used to model the transformation of the 3D space coordinate system to a 2D image plane. An equivalent image plane is established to represent the measurement of the ith observed image feature as

$$I_{ix} = u_0 + f_u \frac{h_{ix}}{h_{ix}} ; \quad I_{iy} = v_0 + f_v \frac{h_{iy}}{h_{iy}} \tag{5}$$

Focal lengths f_u and f_v denote the distance from the camera center to the image plane in u- and v-axis, respectively; (u_0, v_0) is the offset pixel vector of the image plane; $\boldsymbol{h}_i = [h_{ix}\ \ h_{iy}\ \ h_{iz}]^T$ is the ray vector of ith image feature in camera frame. Since the camera frame is set at the center of the left camera, the coordinate representation in the right camera is transformed to the left camera. The coordinate of the ray vector $\boldsymbol{h}_i^R$ in the right camera is transformed to $\boldsymbol{h}_i^L$ in the left camera by

$$\boldsymbol{h}_i^L = \boldsymbol{R}^{-1}\boldsymbol{h}_i^R - \boldsymbol{b} \tag{6}$$

The $\boldsymbol{R}$ and $\boldsymbol{b}$ are the rotation matrix and translation vector, respectively, from left-camera to right-camera,

$$\boldsymbol{R} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} ; \quad \boldsymbol{b} = [b_1 \quad b_2 \quad b_3]^T \tag{7}$$

Equations (5)-(7) are used to determine the 3D coordinates of the ith image feature expressed in the left-camera frame [13] as

$$h_{ix}^L = \frac{1}{k}\left[d_3 \frac{(I_{ix}^R - u_0^R)}{f_u^R} - d_1 \right] \frac{(I_{ix}^L - u_0^L)}{f_u^L} \tag{8a}$$

$$h_{iy}^L = \frac{1}{k}\left[d_3 \frac{(I_{ix}^R - u_0^R)}{f_u^R} - d_1 \right] \frac{(I_{iy}^L - v_0^L)}{f_v^L} \tag{8b}$$

$$h_{iz}^L = \frac{1}{k}\left[d_3 \frac{(I_{ix}^R - u_0^R)}{f_u^R} - d_1 \right] \tag{8c}$$

The superscripts L and R denote the parameters for left-camera and right-camera, respectively. The coefficients k and d_n are given as

$$k = \left[\frac{R_{11}}{f_u^L}(I_{ix}^L - u_0^L) + \frac{R_{12}}{f_v^L}(I_{iy}^L - v_0^L) + R_{13} \right] - \frac{(I_{ix}^R - u_0^R)}{f_u^R}\left[\frac{R_{31}}{f_u^L}(I_{ix}^L - u_0^L) + \frac{R_{32}}{f_v^L}(I_{iy}^L - v_0^L) + R_{33} \right]$$

$$d_n = R_{n1}b_1 + R_{n2}b_2 + R_{n3}b_3 \quad , \quad n = 1,\ldots,3 \tag{9b}$$

The ith observed image feature can then be initialized using the 3D coordinates in the world frame (Figure 2) as follows:

$$\mathbf{m}_i = \boldsymbol{r} + \boldsymbol{R}_L^W \boldsymbol{h}_i^L \tag{10}$$

where $\boldsymbol{r}$ is the position vector of the camera frame; $\boldsymbol{R}_L^W$ is the rotational matrix [14] from the world frame {W} to the left-camera frame {L}; and $\boldsymbol{h}_i^L$ is the ray vector of the image features in the camera frame obtained using Eq. (8). In this study, the hand-held binocular vision sensor is the only sensing device used for measurement in the SLAM system. The hand-held camera is treated as a free-moving robot system with unknown inputs. System states are estimated by using the EKF estimator to solve the target tracking problem [2-3],

$$\mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1|k-1}, \mathbf{u}_{k-1}, 0) \tag{11a}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{A}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{A}_k^T + \boldsymbol{W}_k \boldsymbol{Q}_{k-1} \boldsymbol{W}_k^T \tag{11b}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T + \boldsymbol{V}_k \boldsymbol{R}_k \boldsymbol{V}_k^T)^{-1} \tag{11c}$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \boldsymbol{K}_k (\mathbf{z}_k - g(\mathbf{x}_{k|k-1}, 0)) \tag{11d}$$

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k-1} \tag{11e}$$

where $\mathbf{x}_{k|k-1}$ and $\mathbf{x}_{k|k}$ represent the predicted and estimated state vectors, respectively; $\boldsymbol{K}_k$ is the Kalman gain matrix; $\boldsymbol{P}$ denotes the covariance matrix; $\boldsymbol{A}_k$ and $\boldsymbol{W}_k$ are the Jacobian matrices of state equation f with respect to state vector $\mathbf{x}_k$ and noise variable w_k, respectively; $\boldsymbol{H}_k$ and $\boldsymbol{V}_k$ are the Jacobian matrices of measurement g with respect to state vector $\mathbf{x}_k$ and noise variable v_k, respectively.

The developed SLAM system is implemented on a binocular vision system by integrating the motion and sensor models, as well as the extraction of SURF. The program structure and flowchart for the developed SLAM system is described in [15]. In the system, the images are captured by binocular camera and features are extracted using SURF method. Data association in between the landmarks in the map database and the image features of the extracted SURF is carried out using a proposed matching strategy. A map management is also designed to coordinate the newly extracted features and the bad features in the system. The strategy of image feature detection and matching as well as the tactic of map management will be discussed in detail in the following sections.

Figure 1. Binocular vision sensor system

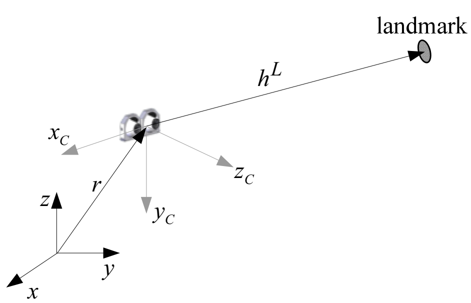

Figure 2. Coordinate setting

III. Data Association

Bay *et al*. [7] utilized the box filter to replace DoG and approximate the determinant of the Hessian matrix. The box filter was further combined with the method of integral image [8] to reduce the image processing time. The size of the box filters will decide the scale of the features detected. Bay *et al.* introduced the concept of octaves to design different size of box filters in order to detect features in different scales. After the features are detected from the image, the description vector is computed to represents the characteristics of features. In order to represent the orientation of a feature, Bay *et al.* [7] used Haar wavelet filter to compute the values of the wavelet responses in the x and y direction of the feature area. The orientation of a feature is defined at the direction with the largest sum of the Haar wavelet responses. Furthermore, a high-dimensional description vector is used to describe the uniqueness of the feature. A square area is chosen with the center located in the feature point and its direction along the direction of the feature. This square area is divided into 4x4 sub-areas. Therefore, a 4-dimensional descriptor vector is assigned for each sub-area, expressed as v=(Σdx, Σdy, Σ|dx|, Σ|dy|), where Σdx, Σdy, Σ|dx|, and Σ|dy| are the sums of the Haar wavelet responses and their absolute values in x and y direction, respectively. All the responses are defined in the coordinate system along the orientation of the feature. This will result in total dimensions of 4x4x4=64 for the description vector of an image feature.

Mapping in visual SLAM requires a robust representation method for the visual landmarks detected from the image. We can see the results in later experimental works that the regular SURF method developed by Bay *et al.* [7] does not inherently meet the requirement of mapping in visual SLAM. In this paper, we will modify the regular SURF method in order to improve the robustness of feature representation and reduce the computational cost. The modified SURF method is described in three aspects, including feature detection, description and matching, in the following subsections.

A. Feature Detection

In SURF detection process, we set up a threshold value ($D_{\text{threshold}}$) as the lowest limit for the determinant of Hessian matrix to adaptively control the number of detected features in an image. The value of $D_{\text{threshold}}$ can be online adjusted according to the environment condition. Our purpose is that, even in the dull background or environment with fewer features, there will be enough number of features detected. One example is implemented to explain the concept. In this example, we utilize a CMOS webcam to capture an image with 320×240 pixels. Then SURF features are detected using different $D_{\text{threshold}}$ values. The results show that the number of detected features varied from 837 to 41 when the value of $D_{\text{threshold}}$ is changed from 0 to 10000. Meanwhile, the computational time is reduced from 43.52ms to 38.44ms when the number of detected features is decreased. Therefore, we can control the number of detected features and save the computational cost by varying the $D_{\text{threshold}}$ value. It will further reduce the time for computing the description vector of features, as we will describe in next sub-section.

Furthermore, we utilize the box filters only in 2 low-level octaves for detecting image features in order to reduce the computation cost for SURF detection. In this experiment, an image with 320×240 pixels is utilized and the value of $D_{\text{threshold}}$ is set to be zero. The results show that the radius of the detected features is proportional to the octave level and the number of detected features is inversely proportional to the octave level. The features detected using the box filters in high-level octaves have large radius. Meanwhile, the location of these features is restricted within the central region of the image. If these large features are located around the boundary region of the image, they couldn't be detected by using the box filters in Octaves 3 and 4. Therefore, the box filters in Octaves 3 and 4 are not utilized in this research.

B. Feature Description

The Haar wavelet response is the convolution of the Haar wavelet filters with the images in a square area, therefore the size of the square area will influence the computational time. In order to improve the computing speed, we reduce the 64-dimensional (64D) description vector to a 16-dimensional (16D) vector. The square area involved in feature area will be shrunk to reduce the computational time of image convolution. Only the central 2x2 sub-areas are selected from the 4x4 sub-areas. Therefore, the dimension of description vector is reduced to be 2x2x4=16. We have compared the processing times when calculating the 16D and 64D description vectors. The number of detected features varies from 100 to 800. The experimental results show that computation for the 16D description vectors is about 45% faster than that for the 64D description vectors.

B. Feature Matching

The Nearest-Neighbor (NN) search method [16] is the most popular method for matching high-dimensional description vectors. If there are two point sets named P and Q in the space of n dimensions. The query point q belongs to Q. We can find the point p in P which has minimum distance with q using NN search method. The distance between the arbitrary point p in P set and the query point q is usually defined by the norm l_s. In the case of l_2, the distance becomes Euclidean distance d. In this paper, the criterion of feature matching is to determine the smallest distance between two descriptors. We set a threshold value of the Euclidean distance, d_{match}, to be a judgment whether the matching between two descriptors is satisfied. When the Euclidean distance d is less than d_{match}, the matching is successful,

$$d = \|p - q\|_2 = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} < d_{\text{match}} \qquad (12)$$

IV. MAP MANAGEMENT

A tactic of map management is designed to manage the newly extracted features and the bad features in the system. The properties of the newly extracted features are investigated and the moving objects will be discriminated from the stationary objects by using a detection algorithm. The state variables of all the stationary landmarks are augmented in the state vector in Eqn. (1). On the other hand, those features which are not continuously detected at each time step will be treated as bad features and erased from the state vector. In the state prediction stage, only landmarks in the camera angle of view (AOV) are considered in the state estimation. Figure 3 depicts the state prediction of landmarks in a map. Landmarks located in front of the camera have the condition $h_z^L > 0$. Six landmarks (landmarks 1-6) satisfy this condition. However, the pixel projections (I_x,I_y) of landmarks no.1 and no.6 are not located on the image plane. Therefore, only four landmarks located in the camera AOV (nos. 2-5) are considered in the state estimation.

Figure 4 shows the size-variable searching window designed to search the newly extracted features for data associations after the landmarks in the camera AOV are determined. The solid squares in the figure indicate landmarks located inside the camera AOV. The smallest size searching window (dash-line squares) is first used to find the corresponding feature. If no image feature is found, the size of the searching window is enlarged and used to determine the corresponding feature. Note that, the threshold value of Euclidean distance d_{match} is reduced to ensure a correct match when the size of the searching window is increased.

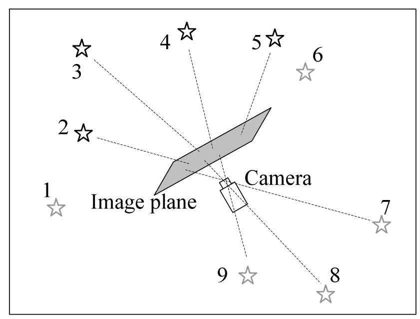

Figure 3. Prediction of the states of landmarks

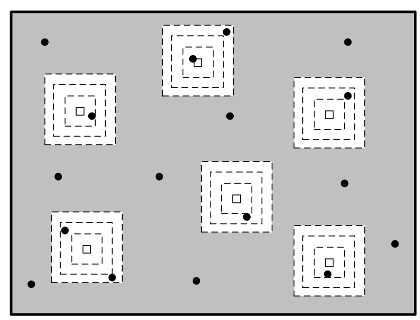

Figure 4. Size-variable searching windows

V. EXPERIMENTAL RESULTS

In this article, the experiment of an indoor SLAM has been carried out on the real system to validate the performance of the proposed algorithm. The kinematic model of camera and the monocular measurement model proposed by Davison *et al.* [2] are implemented on left camera to predict and detect image features. Since right camera of the binocular vision might not be aligned with left camera, the image depth of newly extracted features can be obtained using Equation (8). Therefore, three-dimensional state of these features can be initialized.

In this experiment, the camera is carried by a person to go along the sidewalk of bookshelf in the library. The resultant map and the camera pose estimation are plotted in Figure 5. In this figure, the estimated states of the camera and landmarks are illustrated in a 3D map plot. The ellipses in the figure indicate the uncertainty of the landmarks obtained from the extracted image features. The solid line represents the trajectory of the free-moving binocular vision. The rectangular box represents left camera of the system which is the location of camera frame {C}. The camera is carried to move from first image frame and go through a sidewalk of bookshelf. The SLAM system also starts up from first image frame and captures image features with unknown positions. These features will be initialized and stored as landmarks in the map. When the camera is carried to go along the sidewalk of bookshelf, the SLAM system builds the environment map and estimates the camera pose concurrently.

More detail image frames of the experimental results are illustrated in Figure 6. The top-view plot is depicted in the figure. The estimated states of the camera and landmarks are illustrated in a 2D plot. The red (dark) ellipses represent the uncertainty of the landmarks which are detected in current frame, while the green (light) ellipses denote the uncertainty of the landmarks which cannot be detected in current frame. The SLAM task begins from the first frame. Figure 6 depict the whole motion of camera along the sidewalk of bookshelf. During the motion, the system continuously accumulates static features and utilizes them as landmarks in the map. At the location of 1524th frame in Figure 6, the camera comes to the end of the sidewalk. Totally 318 features are accumulated in the map. The deviations of the estimated camera pose in *xyz*-axis are calculated. The average pose deviation is less than 1.5cm with highest peak no more than 3 cm during the SLAM task.

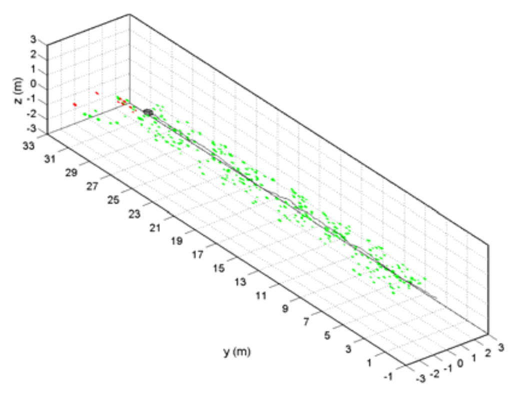

Figure 5. The results of robot SLAM along the sidewalk of bookshelf in the library

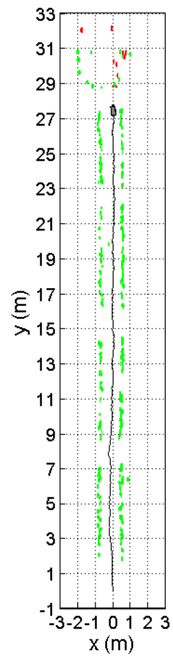

Figure 6. 1524th frame: Less features are available, therefore $D_{\text{threshold}}$ decreases to be 2550. 10 features are selected. The map size increases to 318.

V. Conclusions

In this research, we developed an algorithm for building a persistent map to improve the robustness of robot visual SLAM system. The SURF method, a scale- and orientation-invariant feature detector, is modified to provide a robust detection of image features and a stable description of the features. The modified SURF algorithms are used to construct a sparse-representation map for describing the real environment. Furthermore, the effective procedures of data association and map management for SURF features in SLAM are also designed to improve the accuracy of robot state estimation. Two experimental works have been carried out on an actual system with binocular vision sensors to evaluate the performance of proposed algorithm. First experiment validates the ground truth using the modified SURF algorithm in EKF SLAM. The results showed that the modified SURF with 16D descriptors has the best performance in state estimation of ground truth points. Second experiment is the implementation of SURF-based EKF SLAM on a binocular system in real environment. The results of this experiment showed that the binocular SLAM system with the modified SURF algorithms has the capability to support robot systems simultaneously navigating and mapping in the environment.

ACKNOWLEDGMENT

This paper was partially supported by the National Science Council in Taiwan under grants no. NSC101-2221-E-032-017 and NSC101-2632-E-032-001-MY3.

REFERENCES

[1] C. Harris and Stephens, M., “A combined corner and edge detector,” in *Proceedings of the 4th Alvey Vision Conference*, pp. 147-151, 1988.

[2] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse, “MonoSLAM: real time single camera SLAM,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29 No. 6, pp.1052-1067, 2007.

[3] L.M. Paz, Pinies, P., Tardos, J.D. and Neira, J. “Large-scale 6-DOF SLAM with stereo-in-hand,” *IEEE Transactions on Robotics*, Vol. 24 No. 5, pp. 946-957, 2008.

[4] T. Lindeberg, “Feature detection with automatic scale selection,” *International Journal of Computer Vision*, Vol. 30 No. 2, pp.79-116, 1998.

[5] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, Vol. 60 No. 2, pp. 91-110, 2004.

[6] N. Karlsson, di Bernardo, E., Ostrowski, J., Goncalves, L., Pirjanian, P. and Munich, M.E., “The vSLAM algorithm for robust localization and mapping,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp.24-29, 2005.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “SURF: speeded-up robust features,” *Computer Vision and Image Understanding*, Vol. 110, pp.346-359, 2008.

[8] P.A. Viola and Jones, M.J. “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 511-518, 2001.

[9] A. Gil, O.M. Mozos, M. Ballesta, and O. Reinoso, “A comparative evaluation of interest point detectors and local descriptors for visual SLAM,” *Machine Vision and Applications*, Vol. 21, pp.905-920, 2010.

[10]A.C. Murillo, Guerrero, J.J. and Sagues, C. “SURF features for efficient robot localization with omnidirectional images,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3901-3907, 2007.

[11]C.Y. Wu and Fu, L.C., “An integrated robotic vSLAM system to realize exploration in large indoor environment,” in *Proceedings of the CACS International Automatic Control Conference*, 2007.

[12]S. Hutchinson, Hager, G.D., and Corke, P.I., “A tutorial on visual servo control,” *IEEE Transactions on Robotics and Automation*, Vol. 12, pp.651-670, 1996.

[13]Y.T. Wang, Chi, C.T. and Hung, S.K., “Robot Visual SLAM in Dynamic Environments,” *Advanced Science Letters*, Vol. 8, pp. 229-234, 2012.

[14]L. Sciavicco and Siciliano, B., Modelling and Control of Robot Manipulators, 2nd ed., Springer-Verlag, London, 2000.

[15]Y.T. Wang, Chi, C.T. and Feng, Y.C. “Robot Simultaneous Localization and Mapping Using Speeded-Up Robust Features,” *Applied Mechanics and Materials*, Vol.284-287, pp.2142-2146, 2013.

[16]G. Shakhnarovich, Darrell, T. and Indyk, P. Nearest-neighbor methods in learning and vision, The MIT Press., MA, 2005.