Monocular SLAM for a Small-Size Humanoid Robot

Yin-Tien Wang*, Duen-Yan Hung and Sheng-Hsien Cheng

Department of Mechanical and Electro-Mechanical Engineering, Tamkang University, Tamsui, Taiwan 251, R.O.C.

Abstract

The paper presents a algorithm of visual simultaneous localization and mapping (vSLAM) for a small-size humanoid robot. The algorithm includes the procedures of image feature detection, good feature selection, image depth calculation, and feature state estimation. To ensure robust feature detection and tracking, the procedure is improved by utilizing the method of Speeded Up Robust Features (SURF). Meanwhile, the procedures of image depth calculation and state estimation are integrated in an extended Kalman filter (EKF) based estimation algorithm. All the computation schemes of the visual SLAM are implemented on a small-size humanoid robot with low-cost Window-based PC. Experimentation is performed and the results show that the performance of the proposed algorithm is efficient for robot visual SLAM in the environments.

Key Words: Speeded Up Robust Features (SURF), Simultaneous Localization and Mapping (SLAM), Image Feature Initialization, Humanoid Robot

1. Introduction

Many researchers have successfully implemented robot SLAM systems and validated by supporting experimental results [1–5]. Especially, the *MonoSLAM* developed by Davison et al. [2] provides a real-time algorithm which can locates the 3D trajectory of a monocular camera and maps the beacons in the environments, simultaneously. However, the research in this paper aims at implementing SLAM on a small-size humanoid robot (51 cm in height, 3 kg in weight), as well as providing an efficient image feature initialization algorithm for a class of visual SLAM systems.

MonoSLAM [2] utilizes an extended Kalman filter (EKF) to update the estimation of the robot state and the map of beacons in the environments, recursively. Computational processes in *MonoSLAM* include image capture and feature initialization algorithm, as well as an EKF state estimation process. Both of these procedures need much effort in computation if the scale of the SLAM system increases. Therefore, it is a challenge to implement all *MonoSLAM* processes on a small-size humanoid robot with limited computational capability. In the paper, we present an implementation example on a small-size humanoid robot with a low-cost Window-based PC.

The image features in the environments can be detected and tracked by analyzing the image taken by the robot vision, and then the detected features are utilized as beacons for SLAM systems. Davison et al. [2] employed the concept by Harris and Stephens [6] to extract apparent corner features from one image and track these point features in the subsequent image. Instead of detecting point features, we detect the regional features by using an image scale-invariant method proposed by Bay et al. [7], which they dubbed Speeded Up Robust Features (SURF). In the proposed initialization algorithm, we first utilize the SURF method to detect and track image features or beacons for robot SLAM, and then the pixel coordinates of a feature in two successive images is provided for calculating the 3D spatial position for the corresponding feature.

The proposed SLAM algorithms, including feature initialization and EKF estimation, are verified through

*Corresponding author. E-mail: ytwang@mail.tku.edu.tw

experiments with real platform of a small-size humanoid robot with a Window-based controller. For image processing, we also utilize the OpenCV software library, an open source programming functions for real-time machine vision [8]. The experimental results show that the performance is efficient for supporting the small-size humanoid robot to navigate in the environments.

The contributions in this paper are two-fold. First, we develop a monocular SLAM algorithm based on Speeded-Up Robust Features (SURF). Second, the SLAM algorithm is implemented on a small-size humanoid robot system with Window-based PC controller.

2. EKF-Based SLAM

In the monocular SLAM system, the free-moving camera is presumed to be at constant velocity, and the acceleration is caused by an impulse noise from the external force [2]. We Use the EKF method to estimate the state of the system. The vector of the state is chosen as:

$$x = [x_R^T \quad Y_1^T \quad Y_2^T \quad \cdots \quad Y_n^T]^T \tag{1}$$

x_R is a 12×1 state vector of the camera including the position r^W, rotational angle ϕ^W, velocity v^W, and angular velocity ω^W, all in world frame; Y_i is the three-dimensional coordinates in world frame of i^{th} image feature; n is the number of the image features. The state transition equation of the camera is expressed as:

$$x_{Rk} = f(x_{Rk-1}, u_{k-1}, w_{k-1}) = \begin{bmatrix} r_k^W \\ \phi_k^W \\ v_k^W \\ \omega_k^W \end{bmatrix} = \begin{bmatrix} r_{k-1}^W + (v_{k-1}^W + w_{vk-1})\Delta t \\ \phi_{k-1}^W + (\omega_{k-1}^W + w_{\omega k-1})\Delta t \\ v_{k-1}^W + w_{vk-1} \\ \omega_{k-1}^W + w_{\omega k-1} \end{bmatrix} \tag{2}$$

where k is the time step; Δt is the sampling time interval; w_v and w_ω are linear and angular velocity noise caused by acceleration, respectively.

We use a monocular vision as the only sensing device. The measurement vector is given as

$$z_k = g(x_k, v_k) = [z_{1k}^T \quad z_{2k}^T \quad \cdots \quad z_{mk}^T]^T \tag{3}$$

m is the number of the observed image features in current measurement. The perspective projection method [9] is employed in this research to model the transformation from 2D image plane to 3D space coordinate system. For one observed image feature, the measurement is

$$z_i = \begin{bmatrix} I_{ix} \\ I_{iy} \end{bmatrix} = \begin{bmatrix} u_0 + f_c k_u \dfrac{h_{ix}^C}{h_{iz}^C} \\ v_0 + f_c k_v \dfrac{h_{iy}^C}{h_{iz}^C} \end{bmatrix} \tag{4}$$

where I_{ix} and I_{iy} represent the pixel coordinates of the i^{th} image feature; f_c is the focal length of the camera denoting the distance from the camera center to the image plane; (u_0, v_0) is the offset pixel vector of the pixel image plane; k_u and k_v are the image pixel correctional parameters. Assuming that there is no distortion phenomenon on the image plane and we make k_u and k_v as 1; $h_i^C = [h_{ix}^C \quad h_{iy}^C \quad h_{iz}^C]^T$ is the ray vector of the image features in the camera frame. The 3D coordinates of an image feature in world frame, as shown in Figure 1, is given as

$$Y_{ik} = r_k^W + R_C^W h_{ik}^C \tag{5}$$

R_C^W is the rotational matrix from the world frame {W} to the camera frame {C} and can be represented by using the elementary rotations [10]. We can utilize Eqn. (5) to calculate the ray vector of an image feature in the camera frame. Substituting the ray vectors into Eqn. (4), the coordinates of the image feature in the image plane are obtained. And then the elements of the Jacobian matrix H_k and V_k are determined for the purpose of calculating the matrix of the innovation covariance [11,12].

3. Image Feature Initialization

During navigating in the environments, the robot

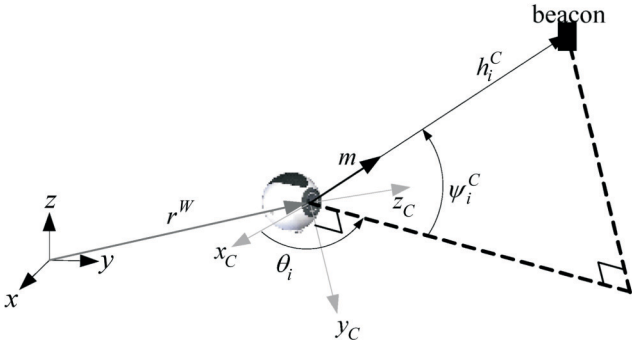

Figure 1. The camera frame and the world frame.

locates its global coordinate by consulting the positions of a group of fixed features or beacons. These features are extracted from the captured image and an initialization process is applied to select *good* features which are successively detected and tracked, as well as to determine the 3D spatial coordinates of these image features. In the paper, a novel image feature initialization algorithm is proposed to initialize the features for SLAM and moving object detection. The procedures of the algorithm include detection of image features, selection of good features, calculation of image depths, and update of feature locations. The procedures are described as the following steps:

(a) Capture an image and smooth the image by using a Gaussian smoothing operator;
(b) Utilize a detecting and tracking method to track a set of image features, called $\mathbf{C}_1$;
(c) In the image set $\mathbf{C}_1$, select *good* features which are successively detected and tracked in a series of images, and discard those *bad* features which are lost during tracking;
(d) Choose c_2 features from the remaining features in set $\mathbf{C}_1$ and compute the spatial coordinates of these features, named image set $\mathbf{C}_2$. These features are selected to be new state variables of the EKF-based SLAM. Meanwhile, the correspondent elements of the state vector and the covariance matrix are initialized.

3.1 Image Feature Detection and Tracking

In Step (b), the features are extracted from the image by using the SURF method proposed by Bay et al. [7]. SURF is a scale-invariant method for detection of image features. It detects region features from an image and obtains the location and the descriptor vector of each interest point. The basic concept of a scale-invariant method is to detect image features by investigating the determinant of Hessian matrix $\boldsymbol{H}$ [13]. Bay et al. [7] utilize a box filter to process on the image instead of calculating the Hessian matrix, and then the determinant of Hessian matrix is approximated by

$$\det(H)_{\text{approx.}} = D_{xx}D_{yy} - (wD_{xy})^2 \tag{6}$$

where D_{ij} are the images filtered by the corresponding box filters [7]; w is a weight constant. The interest points or features are extracted by examining the extreme value of determinant of Hessian matrix. Furthermore, the unique properties of the extracted features are described by using a 64 dimensional descriptor vector as depicted in Figure 2.

The extracted features from two successive images can be matched by checking the Euclidean distance d between the corresponding descriptor vectors, and using the nearest neighbor ratio matching strategy [14]. Therefore, the region features of two images can be tracked efficiently. The matching ratio r is defined as the ratio of the smallest distance d_{1st} to the second smallest distance d_{2nd}. If the ratio r closes to 0.7, we say that these two features are matched.

3.2 Inverse Depth Parameterization

In Step (d) of the image initialization algorithm, the spatial coordinates of image features are calculated. Reference [11] utilized a delay method to initialize the image feature for the SLAM system. We employ the method of inverse depth parameterization [15], an un-delay method, to initialize the image feature. Assume that there are m image features with 3D position vectors, y_i, $i = 1, \ldots, m$, which is described by the 6D state vector:

$$\hat{y}_i = [\hat{r}_{ix}^W \quad \hat{r}_{iy}^W \quad \hat{r}_{iz}^W \quad \hat{\theta}_i^W \quad \hat{\psi}_i^W \quad \hat{\rho}_i]^T \tag{7}$$

$\hat{r}^W = [\hat{r}_{ix}^W \quad \hat{r}_{iy}^W \quad \hat{r}_{iz}^W]^T$ indicates the estimated state of the camera when the feature was observed, as shown in Figure 1; $\hat{\rho}_i$ is the estimated image depth of the feature; $\hat{\theta}_i^W$ and $\hat{\psi}_i^W$ are the longitude and latitude angles of the spherical coordinate system which locates at the camera center. To compute the longitude and latitude angles, a vector η_i^W in the direction of the ray vector is constructed by using the perspective project method:

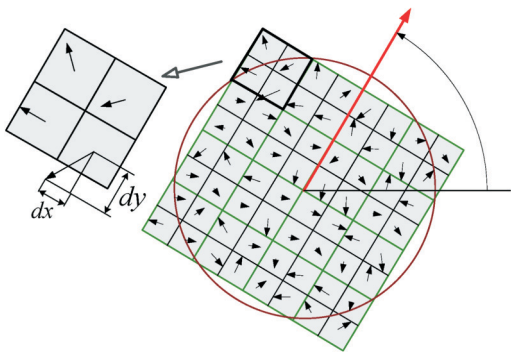

Figure 2. SURF descriptor vector.

$$\eta_i^W = R_C^W(\hat{\phi}^W)\begin{bmatrix} \frac{I_{ix}-u_0}{f_c} & \frac{I_{iy}-v_0}{f_c} & 1 \end{bmatrix}^T \tag{8}$$

Therefore, from Figure 1, the longitude and latitude angles of the spherical coordinate system can be obtained by

$$\hat{\theta}_i^W = \tan^{-1}\left(\frac{\eta_{iz}^W}{\eta_{ix}^W}\right);\ \hat{\psi}_i^W = \tan^{-1}\left(\frac{\eta_y^W}{\sqrt{\eta_{ix}^{W\,2}+\eta_{iz}^{W\,2}}}\right) \tag{9}$$

In Step (d) of the image initialization algorithm, c_2 features are selected to be new state variables of the EKF-based SLAM. Meanwhile, for each new state variable υ_i, the correspondent elements of the state vector and the covariance matrix are initialized according to the criterion of the linearity index [15]:

$$\upsilon_i = \begin{cases} y_i & if\ \ L_d > L_{d0} \\ Y_i & if\ \ L_d \le L_{d0} \end{cases} \tag{10}$$

where L_d is the linearity index, and L_{d0} is the threshold value of the index defined in [15]. The linearity index is utilized to define a threshold for switching from 6D (y_i) to 3D (Y_i) encoding at the point when the latter can be considered linear. When $L_d \approx 0$ is hold, the function of Y_i encoding can be considered linear in the interval. If $L_d > L_{d0}$, the state vector in Eqn. (7) is utilized. The longitude and latitude angles are calculated by using Eqn. (9). The initial value for $\hat{\rho}_i$ and its standard deviation are set $\hat{\rho}_0 = 0.1$, $\sigma_\rho = 0.5$. On the other hand, if $L_d \le L_{d0}$, the state vector in Eqn. (5) is selected and modified as:

$$\hat{Y}_i = \begin{bmatrix} \hat{Y}_{ix} \\ \hat{Y}_{iy} \\ \hat{Y}_{iz} \end{bmatrix} = \begin{bmatrix} \hat{r}_{ix}^W \\ \hat{r}_{iy}^W \\ \hat{r}_{iz}^W \end{bmatrix} + \frac{1}{\hat{\rho}_i} m(\hat{\theta}_i^W, \hat{\psi}_i^W) \tag{11}$$

$$m(\hat{\theta}_i^W, \hat{\psi}_i^W) = \begin{bmatrix} \cos(\hat{\theta}_i^W)\cos(\hat{\psi}_i^W) \\ \sin(\hat{\psi}_i^W) \\ \sin(\hat{\theta}_i^W)\cos(\hat{\psi}_i^W) \end{bmatrix} \tag{12}$$

Meanwhile, for each new state variable υ_i, the correspondent elements of the covariance matrix are initialized. For the case of $L_d > L_{d0}$, the covariance matrix is

$$P_{k|k}^{new} = J\begin{bmatrix} P_{k|k} & 0 & 0 \\ 0 & R_i & 0 \\ 0 & 0 & \sigma_\rho^2 \end{bmatrix} J^T \tag{13}$$

$$J = \left[\begin{array}{cccccc|cc} \multicolumn{6}{c|}{I} & \multicolumn{2}{c}{0} \\ \hline \frac{\partial y}{\partial r^W} & \frac{\partial y}{\partial \phi^C} & 0 & \cdots & 0 & \frac{\partial y}{\partial z_i} & \frac{\partial y}{\partial \rho_i} \end{array}\right] \tag{14}$$

If $L_d \le L_{d0}$, then the covariance matrix is

$$P_{new} = JPJ^T \tag{15}$$

$$J = \begin{bmatrix} I & 0 & 0 \\ 0 & \frac{\partial Y_i}{\partial y_i} & 0 \\ 0 & 0 & I \end{bmatrix} \tag{16}$$

Furthermore, for each new state variable υ_i, the correspondent elements of the Jacobian matrix H_k are modified as:

$$H_k = \frac{\partial g}{\partial x}(x_k, v_k) = \begin{bmatrix} \frac{\partial z_1}{\partial r^W} & \frac{\partial z_1}{\partial \phi^C} & \frac{\partial z_1}{\partial v^W} & \frac{\partial z_1}{\partial \omega^C} & \frac{\partial z_1}{\partial \upsilon_1} & \cdots & \frac{\partial z_1}{\partial \upsilon_n} \\ \frac{\partial z_2}{\partial r^W} & \frac{\partial z_2}{\partial \phi^C} & \frac{\partial z_2}{\partial v^W} & \frac{\partial z_2}{\partial \omega^C} & \frac{\partial z_2}{\partial \upsilon_1} & \cdots & \frac{\partial z_2}{\partial \upsilon_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial z_m}{\partial r^W} & \frac{\partial z_m}{\partial \phi^C} & \frac{\partial z_m}{\partial v^W} & \frac{\partial z_m}{\partial \omega^C} & \frac{\partial z_m}{\partial \upsilon_1} & \cdots & \frac{\partial z_m}{\partial \upsilon_n} \end{bmatrix} \tag{17}$$

The derivative is taken at $x_k = \hat{x}_{k/k-1}$ and $v_k = 0$.

4. Humanoid Robot System

A small-size humanoid robot is designed and fabricated for demonstration, which is equipped with an industrial PC, PCM-3372F, provided by a local vendor. Figure 3 depicts the appearance of the designed robot. The robot is 51 cm in height and 3 kg in weight, as well as has 20 degrees of freedom (DOF), including 6 DOF for each leg, 3 DOF for each arm, and 2 DOF for the head. The mechatronic system comprises three subsystems, including a vision sensor system, a control system, and a motor drive system, as shown in Figure 4. Each subsystem is able to work independently and also in co-

ordination with each other. The camera system, LifeCam VX-6000, is the only sensor to provide the robot position information in the environments. The control system is developed by utilizing an industrial PC running embedded Window XP and Microsoft Visual Studio 2008. All the EKF-based SLAM and image processes are implemented by employing the *OpenCV* (**Open** Source **C**omputer **V**ision), which is a library of programming functions for real time computer vision originated with Intel [8]. The motor drive system is composed of two AGB65-RSC circuit boards and is responsible for driving all the servo motors. Meanwhile, two 12V/2.5AH Li batteries are provided for the power system. One battery supplies the power for 20 servo motors and the other provides for the control system.

5. Experimental Results

Two experiments are presented in this section: monocular SLAM and humanoid robot SLAM. The robot SLAM is performed to demonstrate the proposed algorithms.

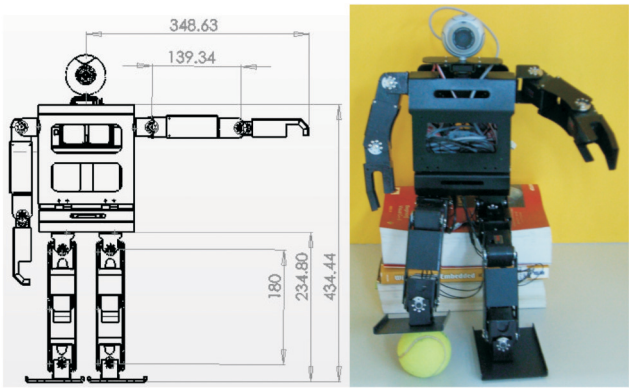

Figure 3. The appearance of the humanoid robot.

5.1 Monocular SLAM

The Monocular SLAM is implemented on a Window-based notebook with a free-moving web camera. We integrate the procedures including the image feature detection and tracking method, feature initialization, system startup procedure, and EKF-based state estimation. In this experiment, the camera moves from the left- to right-side of the field and the estimate state and image features are depicted in Figures 5 and 6. In the plots, the dots indicate the landmarks obtained from the initialized

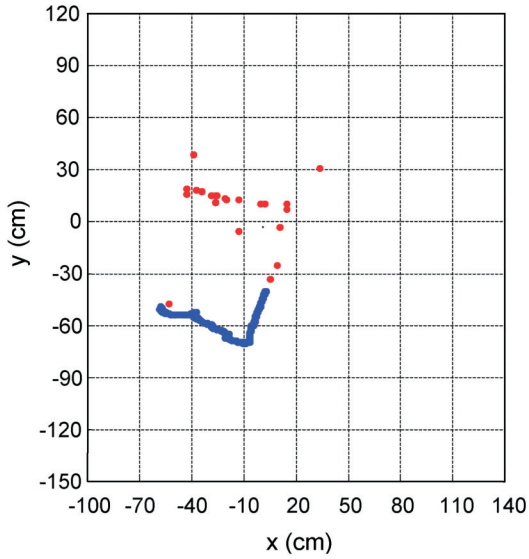

Figure 5. XY-plane plot of the estimate state and image features.

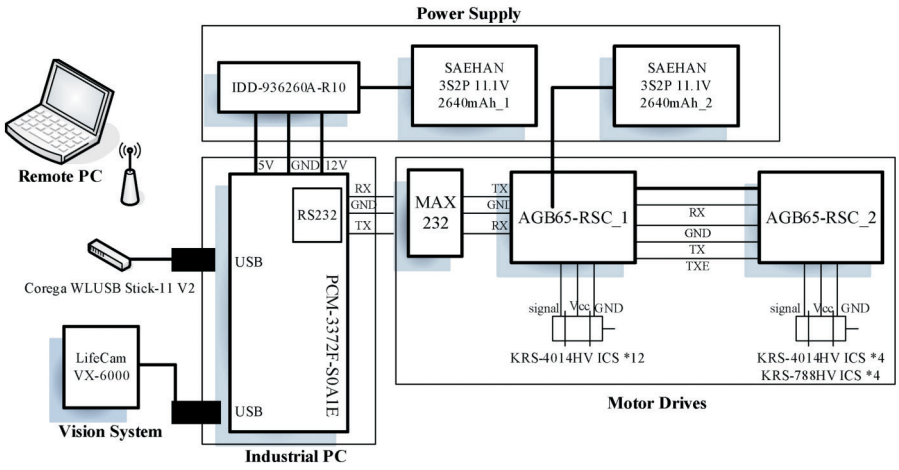

Figure 4. The control system architecture.

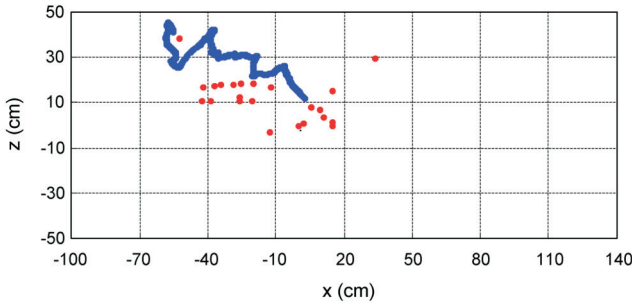

Figure 6. XZ-plane plot of the estimate state and image features.

image features and the dark line represents the motion of the camera.

5.2 Humanoid Robot SLAM

The EKF-based visual SLAM is also implemented on the small-size humanoid robot. The procedures including the image feature detection and tracking method, feature initialization, system startup procedure, and EKF-based state estimation are integrated. In this experiment, the robot moves from the left- to right-side of the field, as shown in Figure 7 and the estimate state and image features are depicted as a 3D map shown in Figure 8. In the plot, the dots indicate the landmarks obtained from the initialized image features and the asterisks represent the state of the camera equipped on the robot. Therefore, the small-size humanoid robot performs the self-localization and mapping procedures simultaneously.

6. Conclusion

In this research, we developed a monocular SLAM for mobile robots. The tasks were implemented on a Window-based PC controller for a small-size humanoid robot. The contribution of this research is in two aspects: firstly, an improved algorithm for image feature initialization was developed for the EKF-based SLAM by using SURF method, which is a scale-invariant feature extraction method. Secondly, all the computational burdens of SLAM and SURF were implemented on a small-size and low-cost industrial PC board. Experimental works were also performed in this paper and the results showed that the SLAM with the proposed feature initialization algorithm has the capability to support the humanoid robot simultaneously navigating and detecting beacons in the environments. In the future, we will extend the SLAM method to detection of moving objects with different moving velocities, or non-rigid moving objects like human body with articulated arms and legs [11,12].

Figure 7. The robot stands in front of the pre-stored image features.

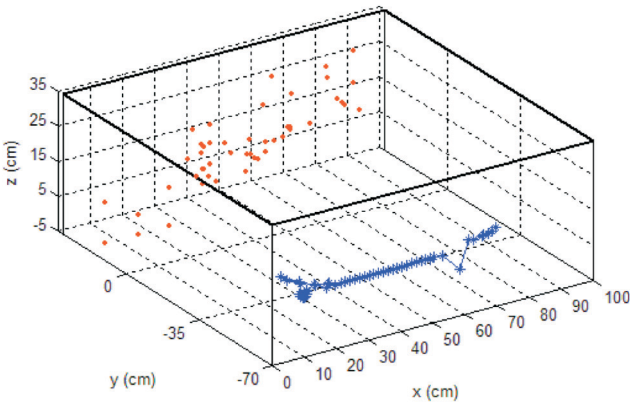

Figure 8. Three-dimensional plot of the estimate state and image features.

Acknowledgment

This work was supported in part by the National Science Council in Taiwan under grant no. NSC99-2221-E-032-064 to Y. T. Wang.

References

[1] Dissanayake, M. W. M. G., Newman, P., Clark, S., Durrant-Whyte, H. and Csorba, M., "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," *IEEE Transactions on Robotics and Automation*, Vol.17, pp. 229–241 (2001).

[2] Davison, A. J., Reid, I. D., Molton, N. D. and Stasse, O., "Mono SLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.29, pp. 1052−1067 (2007).

[3] Montemerlo, M. and Thrun, S., *FastSLAM*, Springer-Verlag (2007).

[4] Wang, C. C., Thorpe, C., Thrun, S., Hebert, M. and Durrant-Whyte, H., "Simultaneous Localization, Mapping and Moving Object Tracking," *International Journal of Robotics Research*, Vol. 26, pp. 889−916 (2007).

[5] Tanaka, M., "Reformation of Particle Filters in Simultaneous Localization and Mapping Problems," *International Journal of Innovative Computing, Information and Control*, Vol. 5, pp. 119−128 (2009).

[6] Harris, C. and Stephens, M., "A Combined Corner and Edge Detector," *Proceedings of the 4th Alvey Vision Conference*, pp. 147−151 (1988).

[7] Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L., "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding*, Vol. 110, pp. 346−359 (2008).

[8] OpenCV. (October 1, 2009). Intel [Online]. Available: http://opencv.willowgarage.com/wiki/.

[9] Hutchinson, S., 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).

[10] Sciavicco, L. and Siciliano, B., *Modelling and Control of Robot Manipulators*, McGraw-Hill, New York, NY (1996).

[11] Wang, Y. T., Lin, M. C., Ju, R. C. and Huang, Y. W., "Image Feature Initialization for SLAM and Moving Object Detection," *Innovative Computing, Information and Control -- Express Letters*, Vol. 3, pp. 477−482 (2009).

[12] Wang, Y. T., Lin, M. C. and Ju, R. C., "Visual SLAM and Moving Object Detection for a Small-Size Humanoid Robot," *International Journal of Advanced Robotic Systems*, Vol. 7, pp. 133−138 (2010).

[13] Lindeberg, T., "Feature Detection with Automatic Scale Selection," *International Journal of Computer Vision*, Vol.30, pp. 79−116 (1998).

[14] Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, pp. 91−110 (2004).

[15] Civera, J., Davison, A. J. and Montiel, J. M. M., "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics*, Vol. 24, pp. 932−945 (2008).

Manuscript Received: Oct. 25, 2009
Accepted: Oct. 15, 2010