Springer

KSME

Journal of Mechanical Science and Technology 26 (3) (2012) 949~958
www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-011-1251-9

Implementation of real-time positioning system using extended Kalman filter and artificial landmark on ceiling †

Angga Rusdinar[1], Jungmin Kim[1], Junha Lee[2] and Sungshin Kim[1,*]

[1]*School of Electrical Engineering, Pusan National University, Busan, 609-735, Korea*
[2]*Department of Interdisciplinary Cooperative Course: Robot, Pusan National University, Busan 609-735, Korea*

(Manuscript Received June 15, 2011; Revised October 5, 2011; Accepted November 2, 2011)

Abstract

Most localization algorithms use a range sensor or vision in a horizontal view, which usually imparts some disruption from a dynamic or static obstacle. By using landmarks on ceiling which the vehicle position were vertically measured, the disruption from horizontal view was reduced. We propose an indoor localization and navigation system based on an extended Kalman filter (EKF) and real-time vision system. A single upward facing digital camera was mounted on an autonomous vehicle as a vision sensor to recognize the landmarks. The landmarks consisted of multiple circles that were arranged in a defined pattern. Information on a landmark's direction and its identity as a reference for an autonomous vehicle was produced by the circular arrangements. The pattern of the circles was detected using a robust image processing algorithm. To reduce the noise that came from uneven light, the process of noise reduction was separated into several regions of interest. The accumulative error caused by odometry sensors (i.e., encoders and a gyro) and the vehicle's position were calculated and estimated, respectively, using the EKF algorithm. Both algorithms were tested on a vehicle in a real environment. The image processing method could precisely recognize the landmarks, and the EKF algorithm could accurately estimate the vehicle's position. The experimental results confirmed that the proposed approaches are implementable.

Keywords: Localization; Navigation; Extended Kalman filter; Robot vision; Artificial landmarks; Region of interest; Ceiling

1. Introduction

The problem of robotic localization and navigation has been widely studied. Autonomous vehicle localization is important because when a robot moves through its environment, its actual position and orientation should be known as a reference. Numerous studies of localization and navigation systems have been performed [1-24]. Most of those studies were applied only to one location. Therefore, when a vehicle moved to a new location or different environment, the algorithms should be modified or built to identify its environment. One method to perform indoor localization is by placing landmarks at known positions and using a mounted sensor on the robot to measure their bearings [1]. Many positions for landmarks as a references have been proposed, including those on floors and walls (i.e., a horizontal views) [3-9], and ceilings (i.e., a vertical views) [10-15]. Since the use of ceiling landmarks has an advantage, which it has a tolerance of dynamic obstacle, most researchers have chosen a landmark on the ceiling as the reference. Fukuda et al. used an air conditioning device (anemo) as a landmark located on the ceiling [15]. Two charge-coupled device (CCD) cameras were used as center and front cameras to recognize the anemo, and fuzzy template matching algorithms were used to recognize the landmarks. Panzieri et al. used ceiling lamps in their experiment as a landmark because ceiling lamps have the same shape if placed in a regular way [18], and can easily be seen without obstacles between the ceiling lamps and the robot vision system. Ceiling light-based positioning methods with all lights shining were assumed; however, there exists another problem when the light is damaged or turned off. In addition, the light may not appear in the view of the camera. Xu used natural features on the ceiling [11]. The initial orientation and position of the autonomous vehicle was estimated via perspective n-point-based positioning method. With the motion of the mobile robot, its global orientation was calculated from the main and secondary lines feature when the ceiling has parallels. In other cases, its global orientation was estimated with point feature on the ceiling. Then, its position is recursively computed with the point features. Wu and Tsai used attached landmarks on the ceiling [10]. An omnidirectional camera was used to take an image of a landmark with a circular shape. The circular shape on the

† This paper was recommended for publication in revised form by Editor Keum-Shik Hong

[*] Corresponding author. Tel.: +82 51 510 23674, Fax.: +82 51 513 0212
E-mail address: sskim@pusan.ac.kr

ceiling was proposed becomes irregular (ellipse) with no mathematical shape descriptor in an omnidirectional image. The parameters of the ellipse are then used for estimating the location of the vehicle with good precision for navigation guidance. Both simulated and real images were tested and good experimental results confirm the feasibility of the proposed approach. The aforementioned methods are feasible when the area of robot operation is small and the camera can obtain the landmark from any position in the room. However, when the robot operates in a large room or in corridors, the circle on the landmark will be difficult to differentiate from the other landmarks. Another disadvantage of this approach is that the robot still needs a sensor (such as a compass) for direction knowledge, which is needed to detect the robot's position in a different direction.

The Kalman filter (KF) and extended Kalman filter (EKF) have been used in many areas including automotive, communications, and simultaneous localization and mapping (SLAM). Turnip et al. used an EKF to estimate nonlinear model parameters of hydro-mount systems [19]. Kim and Hong used an EKF and an unscented Kalman filter for vehicle tracking in an automated container terminal [20]. Park et al. used an EKF to correct a robot pose in a rough surface environment [21]. Myung et al. used a KF to estimate parameters under physical constraints using a general constrained optimization technique [22].

A vision system or a combination of a vision system and sensors has been used in localization and navigation systems. Lee et al. used a combination of vision and a range sensor for mobile robot localization [7]. Park et al. used a CCD camera to calculate the zero moment point that was measured from the reference object image (the camera was located on a robot's head) [16]. Bui et al. used sonar sensor to avoid obstacles in robots navigation [9]. We propose localization and navigation based on an extended Kalman filter and real-time image processing on a vision system. A combination of an odometry sensor and a vision system was used. The vision system was used to change laser range sensor Nav200 in our previous work [23].

The vision system was designed with a simple and implementable landmark (an arranged circle combination) and attached to a ceiling. A digital camera in the face-upward position was mounted on an autonomous vehicle to recognize the landmarks. EKF algorithms were used to estimate the vehicle's position based on image processing information and data from the odometry sensors. The contributions of this study are the following: i) A low cost digital camera replace an expensive laser range finder as a global sensor. ii) A robust and fast image processing algorithm, which has ability to detect and recognize the landmark in real-time in areas with uneven light, is proposed. iii) The incorporation of an EKF algorithm and a vision system improves the correction ability with respect to the vehicles' position when the vehicle moves from one landmark to another. We believe that the proposed system will help engineers to develop simple and cheap systems without reducing the ability of autonomous vehicle guidance (AVG) to localize and navigate in an indoor environment.

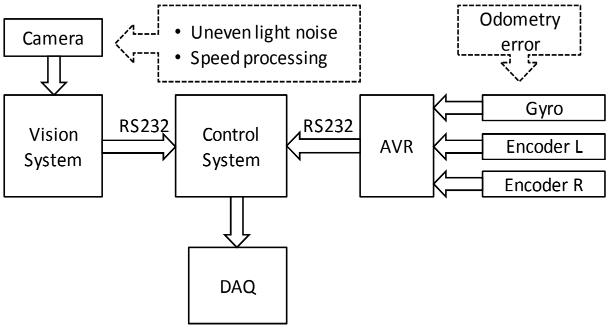


Fig. 1. Designed system configuration.

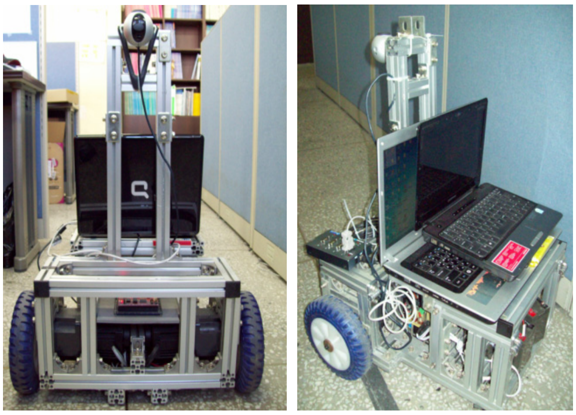


Fig. 2. Vehicle with face-upward digital camera.

The rest of the paper is organized as follows. The configuration of the odometry sensor and vision system is described in Section 2. The EKF is examined in Section 3. The landmark system and its image processing are examined in Section 4. Our experimental results are described in Section 5. Our conclusions and possible future improvements are discussed in Section 6.

2. System configuration of combined odometry sensor and vision

We built an autonomous vehicle with a digital camera mounted on the vehicle. Fig. 1 shows the system configuration. A Compaq Presario CQ20 notebook was used specifically to handle the image processing routines, with the aim that this system can work in real time. The notebook was connected to the main notebook using RS232 serial communication. The main notebook was used to control vehicle movement and estimate vehicle position by the EKF. Two encoders and a gyro were used to calculate vehicle speed and angular velocity.

An AVR ATmega128 microcontroller handled the gyro and the encoders. RS232 serial communication was used to connect the AVR and the main notebook. Two direct current (DC) motors were used to propel the vehicle. The motors were

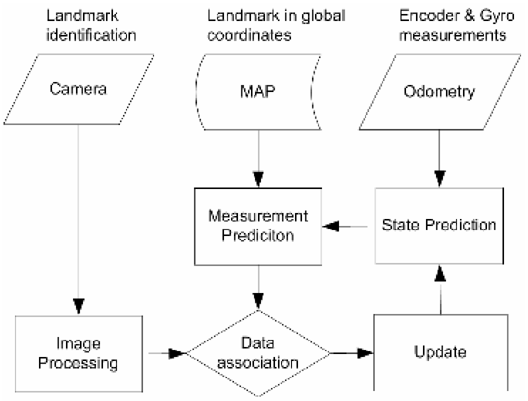


Fig. 3. EKF cycle.

controlled by the main notebook through a USB DAQ motor driver. Fig. 2 shows the vehicle with the face-upward digital camera.

3. Real-time positioning system using EKF and a ceiling landmark

There are two types of areas in the localization system: landmark-detected areas and blind areas. A landmark-detected area is an area in which the camera can capture the landmark. A blind area is an area in which the landmark is not detected by the camera. For the blind area case, the vehicle uses an encoder and a gyro (local sensors) to predict its position. The EKF algorithm was used to correct the accumulative error produced by the encoder and gyro, and the vehicle's position was estimated using the EKF.

3.1 Sensor fusion for local and global localization

The EKF is the nonlinear version of KF which linearizes the current mean and covariance. There are two states in the EKF system: the prediction state and the measurement updating state [25, 26]. The prediction state uses the state estimated from the previous time step to produce an estimate of the state at the current time step. In the updating state, the current priori prediction is combined with current observation information to refine the state estimate. Fig. 3 shows the EKF cycle of our system. The odometry sensor consists of two encoders and a gyro sensor. Data from the odometry sensors were sent to the main processor every 100 ms. In addition, data from the camera were sent to the main processor as long as the landmark is detected.

To predict the state or time update, the state and error covariance are estimated from the previous timestep *k-1* to the current timestep *k*. The prediction state is given by

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, 0)\,, \tag{1}$$

$$\boldsymbol{P}_{\bar{k}} = \boldsymbol{A}_k \boldsymbol{P}_{k-1} \boldsymbol{A}_k^T + \boldsymbol{W}_k \boldsymbol{Q}_{k-1} \boldsymbol{W}_k^T \tag{2}$$

where $\boldsymbol{x}_k$ is a state vector, $\boldsymbol{P}_k$ and $\boldsymbol{P}_{\bar{k}}$ are the a posteriori and a priori estimate of error covariance, and $\boldsymbol{V}_k$ and $\boldsymbol{W}_k$ are the Jacobian matrices of the partial derivatives of f at step k with respect to x and process noise w, respectively, $\boldsymbol{Q}$ is the process noise covariance at step k.

The measurement updating equations correct the state and covariance estimates with the measurement $\boldsymbol{z}_k$. The measurement update equations are defined by

$$\boldsymbol{K}_k = \boldsymbol{P}_{\bar{k}} \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_{\bar{k}} \boldsymbol{H}_k^T + \boldsymbol{V}_k \boldsymbol{R}_k \boldsymbol{V}_k^T)^{-1}\,, \tag{3}$$

$$\boldsymbol{x}_k = \boldsymbol{x}_k + \boldsymbol{K}_k (z_k - \boldsymbol{h}(\boldsymbol{x}_k, 0))\,, \tag{4}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{\bar{k}} \tag{5}$$

where $\boldsymbol{K}_k$ is the Kalman gain, $\boldsymbol{H}_k$ and $\boldsymbol{V}_k$ are the measurement Jacobian matrix of partial derivatives of $\boldsymbol{h}$ at step k with respect to x and measurement noise v, respectively, and $\boldsymbol{R}_k$ is the measurement noise covariance at step k.

3.2 Movement model

To implement the EKF to our system, the proposed motion model is defined by

$$\boldsymbol{x}_k = \begin{bmatrix} x_k^p \\ y_k^p \\ \theta_k^p \end{bmatrix} = \begin{bmatrix} x_{k-1}^p + v\cos(\theta_{k-1} + \omega_k) \\ y_{k-1}^p + v\sin(\theta_{k-1} + \omega_k) \\ \theta_{k-1} + \omega_k \end{bmatrix} \tag{6}$$

where x_k^p, y_k^p and θ_k^p are the (x, y) vehicle position and the heading at step respectively; p indicates that x, y, and θ come from odometry sensor; θ_{k-1} is the robot heading at step *k-1*; ω_k is the angular velocity produced by gyro sensor; and v is the vehicle speed as follows:

$$v = \frac{v_l + v_r}{2} \tag{7}$$

where v_l and v_r are the speed of the left and right vehicle wheel produced by the left and right encoders, respectively.

The time update of vehicle movement in the EKF model is defined by

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \theta) = \boldsymbol{A}_k \boldsymbol{x}_{k-1}^T + \boldsymbol{W}_k \boldsymbol{u}_k^T \tag{8}$$

where A_k is the Jacobian matrix of partial derivatives of f with respect to x, given as

$$A_k = \frac{\partial f(x_{k-1}, u_k, 0)}{\partial x_{k-1}} = \begin{bmatrix} \dfrac{\partial x_k^p}{\partial x_{k-1}^p} & \dfrac{\partial x_k^p}{\partial y_{k-1}^p} & \dfrac{\partial x_k^p}{\partial \theta_{k-1}^p} \\ \dfrac{\partial y_k^p}{\partial x_{k-1}^p} & \dfrac{\partial y_k^p}{\partial y_{k-1}^p} & \dfrac{\partial y_k^p}{\partial \theta_{k-1}^p} \\ \dfrac{\partial \theta_k^p}{\partial x_{k-1}^p} & \dfrac{\partial \theta_k^p}{\partial y_{k-1}^p} & \dfrac{\partial \theta_k^p}{\partial \theta_{k-1}^p} \end{bmatrix}$$

$$=\begin{bmatrix} 1 & 0 & -v_k \sin(\theta_k + \omega_k) \\ 0 & 1 & v_k \cos(\theta_k + \omega_k) \\ 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

$\boldsymbol{W}_k$ is the Jacobian matrix of the partial derivative of f with respect to driving function $\boldsymbol{u}$, given as

$$\boldsymbol{W}_k = \frac{\partial f(\boldsymbol{x}_{k-1}^p, \boldsymbol{u}_k, 0)}{\partial u_k} = \begin{bmatrix} \frac{\partial x_k^p}{\partial v_k^p} & \frac{\partial x_k^p}{\partial \omega_k^p} \\ \frac{\partial y_k^p}{\partial v_k^p} & \frac{\partial y_k^p}{\partial \omega_k^p} \\ \frac{\partial \theta_k^p}{\partial v_k^p} & \frac{\partial \theta_k^p}{\partial \omega_k^p} \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_{k-1} + \omega_k) & 0 \\ \sin(\theta_{k-1} + \omega_k) & 0 \\ 0 & 1 \end{bmatrix}. \tag{10}$$

To support the measurement update, the measurements z_k and $\boldsymbol{H}_k$ are defined as

$$z_k = \boldsymbol{h}(\boldsymbol{x}_k, X_{vision}, v) = \begin{bmatrix} z_{dist} \\ z_{theta} \end{bmatrix}$$

$$= \begin{bmatrix} ((x_{vision} - x_k^p)^2 + (y_{vision} - y_k^p)^2)^{1/2} \\ \tan^{-1}\left(\frac{y_{vision} - y_k^p}{x_{vision} - x_k^p} \right) - \theta_k + \frac{\pi}{2} \end{bmatrix}, \tag{11}$$

$$\boldsymbol{H}_k = \begin{bmatrix} \frac{\partial h(x_k, x_{vision}, 0)}{\partial x_k} & \frac{\partial h(x_k, x_{vision}, 0)}{\partial x_{vision}} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} \frac{\partial z_{dist}}{\partial x_k^p} & \frac{\partial z_{dist}}{\partial y_k^p} & \frac{\partial z_{dist}}{\partial \theta_k^p} \\ \frac{\partial z_{theta}}{\partial x_k^p} & \frac{\partial z_{theta}}{\partial y_k^p} & \frac{\partial z_{theta}}{\partial \theta_k^p} \end{bmatrix} & \begin{bmatrix} \frac{\partial z_{dist}}{\partial x_{vision}} & \frac{\partial z_{dist}}{\partial y_{vision}} \\ \frac{\partial z_{theta}}{\partial x_{vision}} & \frac{\partial z_{theta}}{\partial y_{vision}} \end{bmatrix} \end{bmatrix}$$

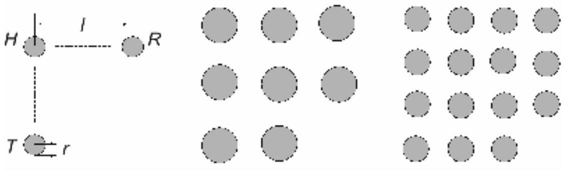
$$= \begin{bmatrix} -\frac{x_{vision} - x_k^p}{dist} & -\frac{x_{vision} - x_k^p}{dist} & 0 & \frac{x_{vision} - x_k^p}{dist} & -\frac{y_{vision} - y_k^p}{dist} \\ \frac{y_{vision} - y_k^p}{dist^2} & \frac{x_{vision} - x_k^p}{dist^2} & -1 & -\frac{y_{vision} - y_k^p}{dist^2} & \frac{x_{vision} - x_k^p}{dist^2} \end{bmatrix}, \tag{12}$$

$$dist = \left((x_{vision} - x_k^p)^2 + (y_{vision} - y_k^p)^2 \right)^{1/2} \tag{13}$$

where x_{vision} and y_{vision} are the vehicle positions produced by the camera, and *dist* is the measured position error between the estimated position from the odometry sensor and the camera.

4. Landmarks

Landmarks are images that consist of some circles arranged to form the code information representing identity and direction [27]. Identity and direction are used to mark the position of landmarks and determine the robot's head, and act as a guide to position the next landmark so the vehicle can follow the direction of the landmarks.



(a) H, R and T position (b) 3X3 type (c) 4 X 4 types

Fig. 4. Landmark schemes.

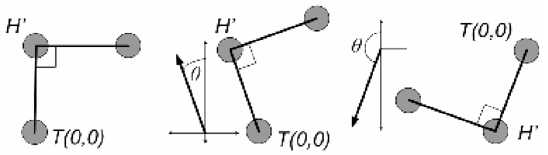


Fig. 5. Landmark direction.

4.1 Landmark direction and identity

Landmark identity is defined by the circle arrangement in the image. To recognize the identity, the circle coordinates of head (*H*), tail (*T*), and right (*R*) must first be defined. *H* is defined as a meeting point between two lines formed from the coordinates of *T* and *R*, where the two lines have equal length and a line perpendicular to each other. Fig. 4 shows the landmark schemes. Which are; (a) show *H*, *R*, and *T* positions, where *l* is the distance between *H* and *R*, (b) and (c) show the dimension of the landmarks in 3×3 and 4×4 types of size. The landmark direction is determined by the position of *T* and *H*. If we assumed that the line formed between *T* and *H* is a line of arrows, then the heads of the arrows are *H* and the tails are *T*. The direction of the landmark is obtained by the angle formed between the arrows and the y-axis in the positive direction. To calculate the direction, *T* is shifted into the center of the landmark $T'(0, 0)$. The coordinate of *H* is also shifted into the new coordinate H'. The new coordinate of H' is defined by Eq. (14), and the direction is defined by Eq. (15)

$$H'_{xy} = H_{xy} - T_{xy} \ , \tag{14}$$

$$\theta = \tan^{-1}\left(\frac{H'_x}{H'_y}\right). \tag{15}$$

The landmark identity is defined by the arrangement of a circle in the image. Fig. 6 shows the bit position in landmark. Bit positions are defined by circle coordinates. To define the coordinates, the *n*-th coordinates must be determined; this point is called E' and is defined by

$$E'_{xy} = T_{xy} + (R_{xy} - H_{xy}) \tag{16}$$

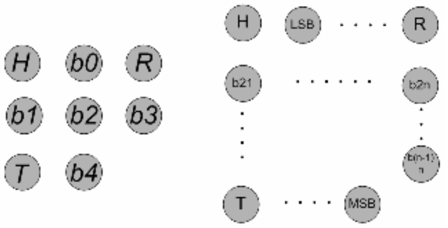


Fig. 6. Bit position in landmark.

where E'_{xy}, T_{xy}, R_{xy}, and H_{xy} are the coordinates of E', T, R, and H respectively. Then, the coordinate of each circle is defined by

$$b_{1j} = \frac{j}{(n-1)}(R_{xy} - H_{xy}) \quad j = 2 \cdots n-1, \tag{17}$$

$$b_{i1} = \frac{i}{(n-1)}(T_{xy} - H_{xy}) \quad i = 2 \cdots n-1, \tag{18}$$

$$b_{nj} = \frac{j}{(n-1)}(E'_{xy} - T_{xy}) \quad j = 2 \cdots n-1, \tag{19}$$

$$b_{in} = \frac{i}{(n-1)}(E'_{xy} - R_{xy}) \quad i = 2 \cdots n-1, \tag{20}$$

$$b_{ij} = \frac{j}{(n-1)}(b_{in} - b_{i1}) \quad i = 2 \cdots n-1, \; j = 2 \cdots n-1. \tag{21}$$

Each coordinate of bit is defined by the circles position, which its coordinate were searched by using image processing. The bits were defined by "1" for any position indicated by a black circle, and by "0" for the empty position. Then, the result is arranged into a binary code format. $\boldsymbol{B}_l = \left[bit_{(b-1)} \;\; bit_{(b-2)} \;\; \cdots \;\; bit_1 \;\; bit_0 \right]$ is an obtained binary code from the image processing. The binary code is changed into decimal code by

$$id = \boldsymbol{B}_l . \boldsymbol{d}^T \tag{22}$$

where *id* is the identity number of a landmark and *d* is defined by

$$\boldsymbol{d} = \left[2^{(b-1)}{}_{MSB} \;\; 2^{(b-2)} \;\; \cdots\cdots \;\; 2^1 \;\; 2^0{}_{LSB} \right] \tag{23}$$

where *b* is the number of the bits, it defined by $b = n^2 - 4$, then the *id* is used to mark every coordinate point in the navigation.

4.2 Image processing of the landmark

Speed and noise are the main problems in image processing. Complicated algorithms make image processing run slowly, and uneven light during robot motion in the hallway causes much noise. Thus, the image recognition process becomes difficult. To solve this problem, we propose a simple and fast algorithm to recognize the circle and its arrangement. To make the recognition process work perfectly, the color image is changed into a black and white image. By changing color, the threshold search algorithms should have the ability to adapt to uneven light.

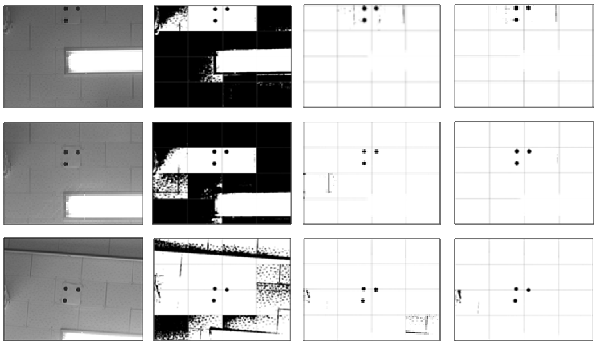


Fig. 7. Landmark on ceiling in light condition.

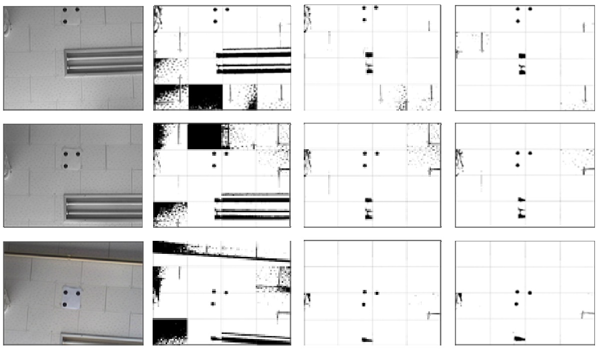


Fig. 8. Landmark on ceiling in dim light condition.

4.2.1 Threshold searching

In preprocessing, the image was divided into 16 regions, and the processes of threshold searching and noise removal were performed in each region. Noise was removed in three stages. First, the minimum and maximum value of pixels were searched. Then, the threshold was calculated as

$$thr = \tfrac{1}{2}(RGB_{\max} + RGB_{\min}) . \tag{24}$$

Using Eq. (24), the threshold in each region was defined, where *thr* is the threshold in every region, and $RGB_{\max}$ and $RGB_{\min}$ are the maximum and minimum RGB pixel values in every region, respectively. The RGB pixel was changed to black or white based on the threshold. Second, the black area calculation was performed in every region. The black area, which was more than thirty percent of the region size, was assumed as noise. Third, we performed a black pixel spreading calculation. On this stage, a grid was built in each region with a distance of five pixels, and the number of black pixels was calculated. If the spreading of a black pixel by more than half of the region size occurred, then the pixel was assumed to be noise. Fig. 7 (when the light is on) and Fig. 8 (when the light is off) show the results of preprocessing.

4.2.2 Landmark recognition

A region of interest (ROI) was built to limit each circle distance. In this ROI, the circle and landmark shape recognition processes were performed. The position of the ROI changed in every frame of the image if the landmark was not detected, otherwise, the ROI moved to follow the landmark position. To recognize the circle, the searching method of Ho and Chen was used [28]. The center of the recognized circles and their radii were saved into the memory array. Then the positions of the circles were classified into four categories: highest (top), lowest (bottom), left-most, and right-most positions. By calculating and comparing the distance between these positions, the circle coordinates of *H*, *R*, and *T* were defined. As explained in Section 3.1, after the head of the landmark was found, the *H* position was used as a landmark coordinate. The *H* position was defined by H_{xy}, where H_{xy} is the (*x*, *y*) position of the landmark head. Since the circle positions were in the ROI, the landmark position in the image can be described as

$$l_{xy} = H_{xy} + f_{xy} \tag{25}$$

where *fxy*=(*fx*, *fy*) is the ROI position. The ROI position was used to track the landmark image when the vehicle was moving, and the ROI position was updated at every time step as long as the landmarks were captured and recognized by the camera. The new region of interest position is given by

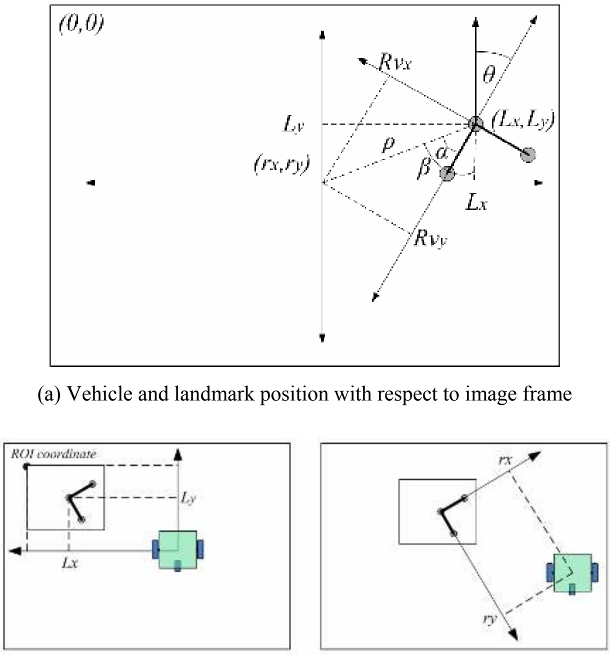
$$f_{xy} = ((f_{x-1} + H_x) - wd/2, (f_{y-1} + H_y) - hg/2) \tag{26}$$

where (f_{x-1}, f_{y-1}) is the landmark's position in the previous frame, and wd and hg are the width and height of the ROI, respectively.

The tracking speed of the landmark position increased using the small window processing methods. When the vehicle moved, the head landmark position was held in the center of a small window and updated every time a new frame was recorded. This makes the window position follow the landmark image in the frame. Fig. 9(a) indicates the position of the vehicle and the landmarks with respect to the image frame (0, 0). Let (r_x, r_y) be the center of the vehicle position in the image frame (i.e., $r_x = fc/2$ and $r_y = fr/2$, where fc and fr are the column and row dimensions of the image). (L_x, L_y) is the landmark's position with respect to the center of the vehicle, and its coordinates were calculated as $(L_x, L_y) = ((l_x - r_x),\ (l_y - r_y))$, $\rho = (L_x^2 + L_y^2)^{1/2}$, $\beta = \tan^{-1}(L_x/L_y)$, $\alpha = \beta - \theta$. Then, the vehicle's position with respect to the landmark is calculated as $Rv_x = \rho\cos\alpha$, $Rv_y = \rho\sin\alpha$. Fig. 9(b) shows ROI frame coordinate and landmark pose in frame.

Fig. 9(c) shows the robot pose.

The results of the recognition process were verified by placing the vehicle under both light and dim lighting conditions. Fig. 10 shows the image processing that was performed to detect and recognize the landmark under the light condition. The rectangular line that connected the top, bottom, left-most, and right-most of circles indicated that the landmark was recognized. The empty circle indicates the position of the bit in the landmark. Fig. 11 shows that when the lights are turned off, the image processing algorithm directly detected the landmarks. The changing of the intensity of the light can disrupt the threshold value that was used to change the color image to black and white. By separated threshold searching in regions, the threshold was able to adapt according to the intensity of the light. Thus, the recognition processing was successful.



(a) Vehicle and landmark position with respect to image frame

(b) ROI frame coordinate

(c) Robot pose

Fig. 9. Vehicle and landmark coordinate system.

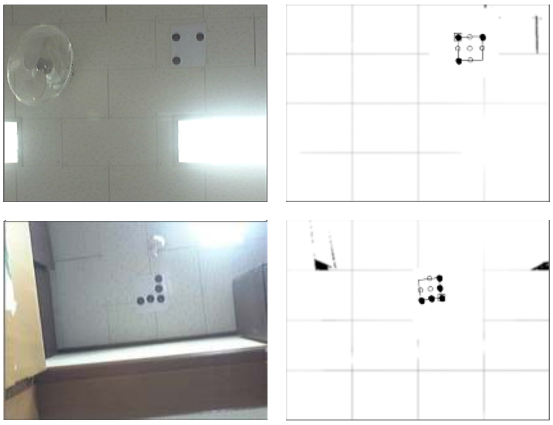


Fig. 10. Landmark recognition result in light condition.

5. Experimental results

Two experiments were performed. First, the vehicle was

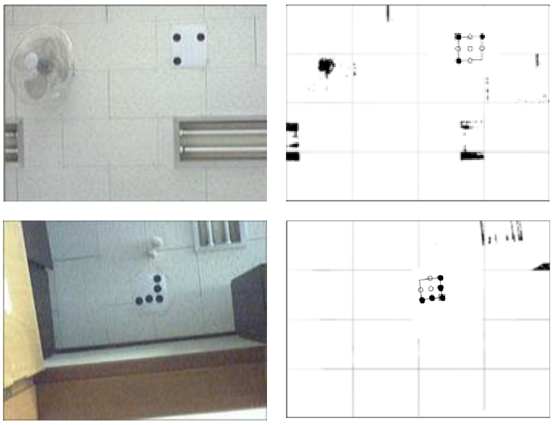


Fig. 11. Landmark recognition result in dim light condition.

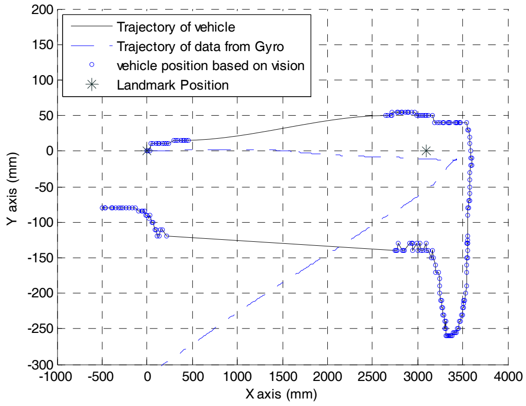


Fig. 12. One round trip trajectory of data from gyro.

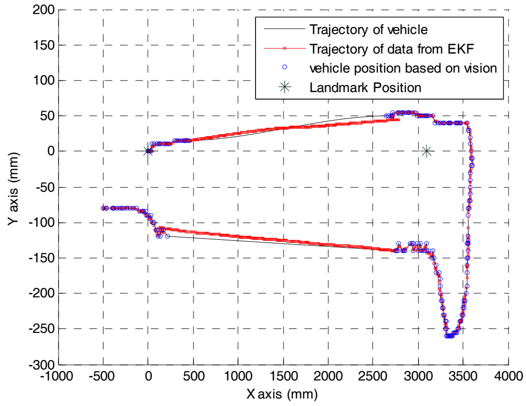


Fig. 13. One round trip trajectory of data from EKF.

moved between two landmarks at a distance of 3100 mm. Second, the vehicle was moved in the corridors of our building. In the first experiment, the vehicle moves from the first landmark to the second landmark. When it reached the second landmark, it made 180 degree turn and move back to the first landmark. The vehicle move in one and two round trips, and the results were analyzed. Figs. 12 and 13 show vehicle movements for one round trip. Fig. 12 shows the vehicle trajectory with vehicle position based on image processing and data trajectory from the gyro. Because of the gyro error, the distance of the real position from the vehicles position became large. Fig. 13 shows the trajectory result from the EKF algorithms, and vehicle position based on camera processing. The EKF had an ability to correct the accumulative error produced by the gyro. Figs. 14 and 15 show the vehicles trajectory in two round trips from the gyro. Fig. 14 indicates that the farther the vehicle moved, the greater the resulting error obtained. Fig. 15 shows the corrected position error of the vehicle using the EKF algorithm.

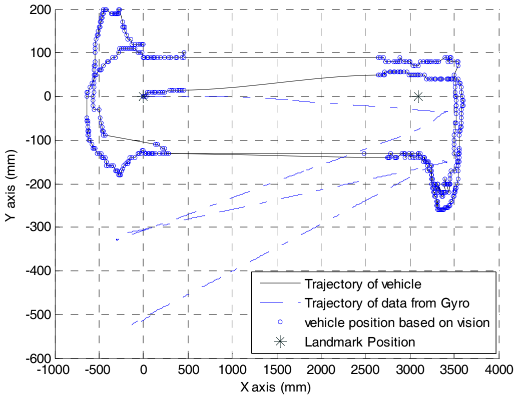


Fig. 14. Two round trips trajectory of data from gyro.

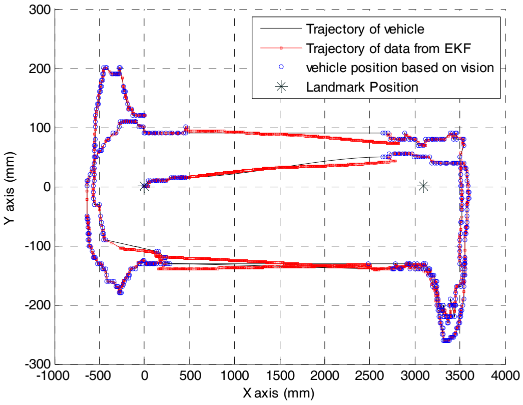


Fig. 15. Two round trip trajectory of data from EKF.

The second experiment was performed in the corridors of our building with the five landmark position map is given in Fig. 16. In the first trial, the vehicle was moved from landmarks 1-2-3-4-5-2-1. While the vehicle moved, the position data determined by the gyro, EKF algorithms was saved, and the EKF position data was used for vehicle movement control.

Fig. 17 shows the vehicle trajectory obtained by ground truth and the data from the gyro sensor. It shows that the accumulative gyro error made the vehicle trajectory far away from the real vehicle position. The vehicle trajectory based on EKF algorithm is given in Fig. 18. It shows that the EKF is able to correct the vehicle position.

In the second trial, the vehicle was moved through landmarks 1-2-3-4-5-2-3-4-5-2-1. Fig. 19 shows the vehicle trajectory based on ground truth and gyro sensor. It shows that the

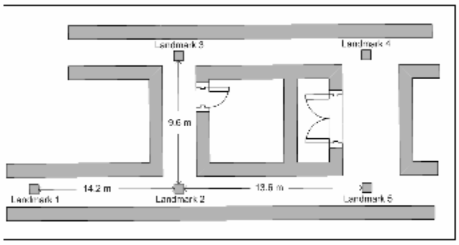


Fig. 16. Experimental environment map.

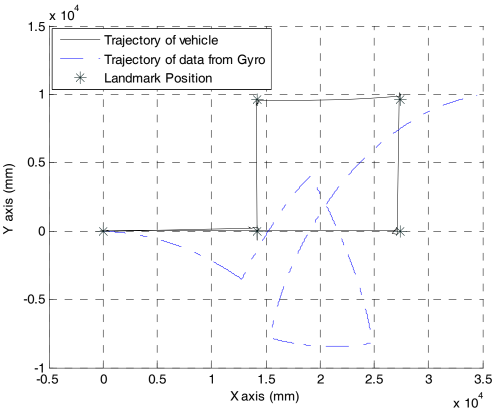


Fig. 17. Plot of trajectory of data from gyro and vehicle in one lap.

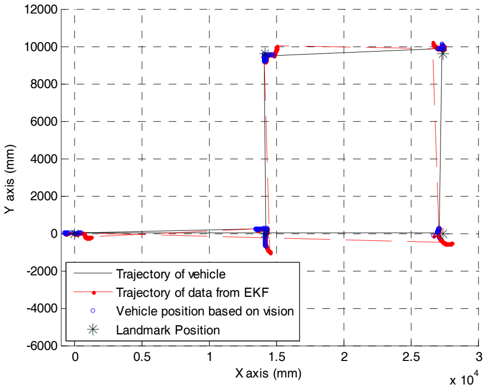


Fig. 18. Plot of trajectory of data from EKF and vehicle in one lap.

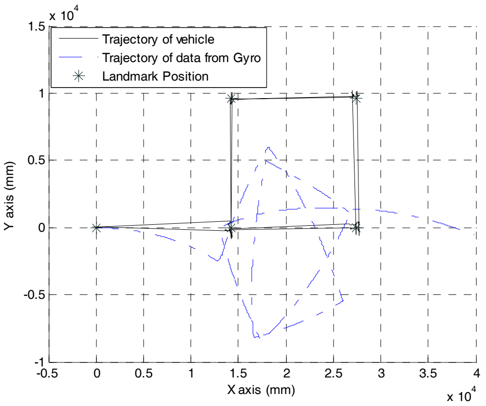


Fig. 19. Plot of trajectory of vehicle and data from gyro in two laps.

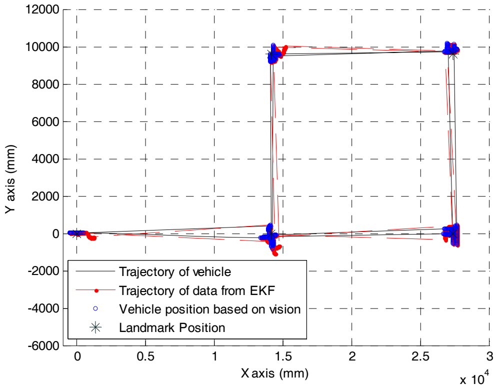
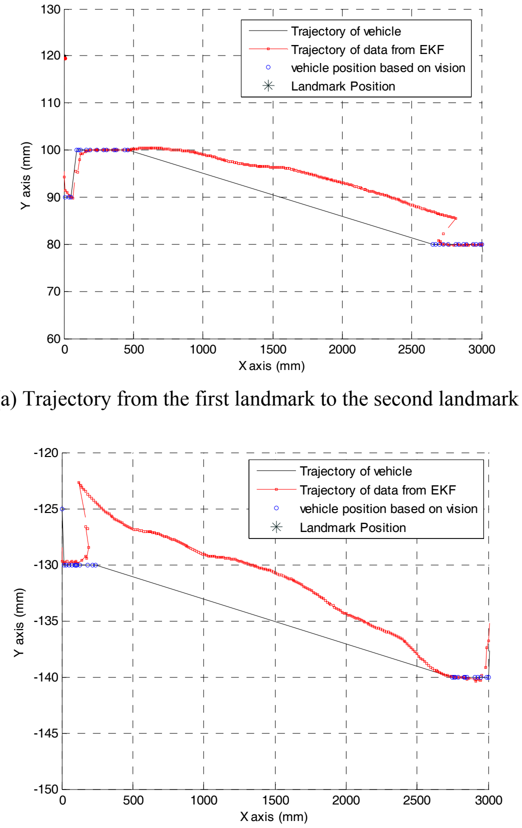


Fig. 20. Plot of trajectory of vehicle and EKF in two laps.



(a) Trajectory from the first landmark to the second landmark

(b) Trajectory from the second landmark to the first landmark

Fig. 21. Zoom in image of the vehicle's trajectory between two landmarks.

position errors based on gyro compared to the real position are large. Fig. 20 shows the result of the EKF algorithm. The results show that although the vehicle moved two laps and the error position from the gyro continued to increase, the EKF algorithms were able to correct the position. Fig. 21 shows the zoom in image of vehicle's trajectory between two landmarks. It shows that the error of vehicle position estimation can be tolerated. Fig. 21(a) shows the trajectory of vehicle and data position from EKF, when vehicle move from the first landmark to the second landmark. Fig. 21(b) shows the trajectory when the vehicle move from the second landmark to the first landmark.

6. Conclusions

We proposed localization and navigation based on an extended Kalman filter and real-time image processing on a vision system. According to the results, it has been confirmed that the combination of an odometry sensor, a vision sensor, and an EKF system can localize and navigate a vehicle in indoor environments.

We proposed an identifiable landmark recognition by image processing and navigation system algorithms using only single low-cost digital cameras and odometry sensors. The landmark recognition method makes the system easy to implement in any new indoor environment otherwise without many algorithms changing the vehicle can operate in other operation area. Using ceiling landmarks, every disruption from static or dynamic obstacles was eliminated. The system and the proposed algorithms movies can be viewed at Ref. [26].

In future work, natural landmark detection will be implemented to improve the prediction of position and to calculate the vehicle's position when vehicle unable to detect and select the artificial landmark in blind area.

Acknowledgment

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the Human Resources Development Program for Specialized Navigation / Localization Technology Research Center support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-C7000-1001-0004).

Nomenclature

EKF	: Extended Kalman filter
P_k	: Posteriori estimate error covariance
Q	: Process noise covariance
A_k	: Process Jacobian of partial derivatives of *f* with respect to *x*
W_k	: Process Jacobian of partial derivatives of *f* with respect to *w*
Z_k	: Measurement noise at time *k*
K_k	: Kalman gain
Rk	: Measurement noise covariance at step *k*
h	: Non linear measurement function
Hk	: Measurement matrix
ROI	: Region of interest
H	: Head of landmark
T	: Tail of landmark
R	: Right side of landmark
Id	: Identity of landmark
r	: Radius of circles
l	: Distance between H and R
Bl	: Binary code of landmark
θ	: Heading angle of vehicle to landmark
thr	: Threshold of pixel
fc	: Frame column
fr	: Frame rows
RGB	: Pixel number of red, green blue
L_{xy}	: Landmark position when vehicle as reference
Rv_{xy}	: Vehicle position when landmark as reference
wd	: Width of ROI
hg	: Height of ROI

References

[1] I. Loevsky and I. Shimshoni, Reliable and efficient landmark based localization for mobile robot, *Robotics and Autonomous System*, 58 (5) (2010) 520-528.

[2] T. A. Tamba, B. Hong and K.-S. Hong, A path following control of an unmanned autonomous forklift, *International Journal of Control, Automation, and Systems*, 7 (1) (2009) 113-122.

[3] I. Shimshoni, On mobile robot localization from landmark bearings, *IEEE Transactions on Robotics and Automation,* 18 (3) (2002) 971-976.

[4] M. Betke and L. Gurvits, Mobile robot localization using landmarks, *IEEE Transactions on Robotics and Automation,* 13 (2) (1997) 251-263.

[5] A. Widyotriatmo, B. Hong and K.-S. Hong, Predictive navigation of an autonomous vehicle with nonholonomic and minimum turning radius constrains, *Journal of Mechanical Science and Technology*, 23 (2) (2009) 381-388.

[6] N. M. Kwok, Q. P. Ha, S. Huang, G. Dissanayake and G. Fang, Mobile robot localization and mapping using a Gausian sum filter, *International Journal of Control, Automation, and System*, 5 (3) (2007) 251-268.

[7] Y. J. Lee, B. D. Yim and J. B. Song, Mobile robot localization based on effective combination of vision and range sensor, *International Journal of Control, Automation, and System*, 7 (1) (2009) 97-104.

[8] K. Lee, N. L. Doh and W. K. Chung, An exploration strategy using sonar sensor in corridor environments*, Intelligent Service Robotics*, 3 (2) (2010) 89-98.

[9] T. T. Q. Bui and K.-S. Hong, Sonar based obstacle avoidance using region partition scheme, *Journal of Mechanical Science and Technology*, 24 (1) (2010) 365-372.

[10] C. J. Wu and W. H. Tsai, Location estimation for indoor autonomous vehicle navigation by omni-directional vision using circular landmarks on ceilings, *Robotics and Autonomous System,* 57 (5) (2009) 546-555.

[11] D. Xu, L. Han, M. Tan and Y. F. Li, Ceiling based visual position for an indoor mobile robot with monocular vision, *IEEE Transactions on Industrial Electronics*, 56 (5) (2009) 1617-1628.

[12] H. Chen, D. Sun and J. Yang, Global localization of multi robot formation using ceiling vision SLAM strategy, *Mechatronics*, 19 (5) (2009) 617-628.

[13] E. Menegatti, T. Maeda and H. Ishiguro, Image based memory for robot navigation using properties of omnidirectional images, *Robotics and Autonomous System*, 47 (4) (2004) 251-267.

[14] H. Wang, H. Yu and L. Kong, Ceiling light landmarks based localization and motion control for a mobile robot, *Proc. IEEE Int. Conf. Network Sens. Control*, London, U.K. (2007) 285-290.

[15] T. Fukuda, S. Ito, F. Arai and Y. Yokohama, Navigation system based on ceiling landmark recognition for autonomous mobile robot – Landmark detection based on Fuzzy Template Matching, *Proc. of The IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Pittsburgh, Pennsylvania, USA (1995) 150-155.

[16] S. Park, Y. Han and H. Hahn, Balance control of biped robot using camera image of reference object, *International Journal of Control, Automation, and System*, 7 (1) (2009) 75-84.

[17] A. Widyotriatmo and K.-S. Hong, Navigation function based control of multiple wheeled vehicles, *IEEE Transactions on Industrial Electronics*, 58 (5) (2011) 1896-1906.

[18] S. Panzieri, F. Pascucci, R. Setola and G. Ulivi, A low cost vision based localization system for mobile robots. *Proc. of the 9th Mediterranean Conf. Control Automatic*, Dubrovnik, Croatia (2001).

[19] A. Turnip, K.-S. Hong and S. Park, Modeling of a hydraulic engine mount for active pneumatic engine vibration control using the extended Kalman filter, *Journal of Mechanical Science and Technology,* 23 (1) (2009) 229-236.

[20] Y. S. Kim and K.-S. Hong, A tracking algorithm for autonomous navigation of AGVs in an automated container terminal, *Journal of Mechanical Science and Technology*, 19 (1) (2005) 72-86.

[21] J. Park, S. Lee and J. Park, Correction robot pose for SLAM based on EKF in a rough surface environment, *International Journal of Advanced Robotic System*, 6 (2) (2009) 67-72.

[22] H. Myung, H. K. Lee, K. Choi and S. Bang, Mobile robot localization with gyroscope and constrained Kalman filter, *International Journal of Control, Automation, and System*, 8 (3) (2010) 667-676.

[23] A. Rusdinar, J. Kim and S. Kim, Error pose correction of mobile robot for SLAM problem using laser range finder based on particle filter, *International Conference on Control, Automation, and Systems*, Gyeonggi-do, Korea (2010) 52-55.

[24] A. Widyotriatmo, K.-S. Hong and L. H. Prayudhi, Robust stabilization of a wheeled vehicle: Hybrid feedback control design and experimental validation, *Journal of Mechanical Science and Technology*, 24 (2) (2010) 513-520.

[25] G. Welch and G. Bishop, *An introduction to the Kalman filter*, Course-8, Department of computer science, University of North Carolina, USA (2001).

[26] W. Burgard, C. Stachniss, M. Bennewitz, G. Grisetti and K. Arras, *Introduction to mobile robotic EKF localization*, Course-2010, Uni. Freiburg, Germany (2010).

[27] http://www.robotshop.com/hagisonic-stargazer- localization - system - 3. html.

[28] C. T. Ho and L. H. Chen, A fast ellipse/circle detector using geometric symmetry, *Pattern Recognition*, 28 (1) (1995) 117-124.

[29] http://www.youtube.com/watch?v=qM1mcRejils.

Angga Rusdinar received his B. Eng. degree in Electrical Engineering from Sepuluh Nopember Institute of Technology, Indonesia, in 2001 and M. Eng. degree from the School of Electrical Engineering and Informatics, Bandung Institute of Technology, Indonesia in 2006. He is currently a Ph.D. program student in the School of Electrical Engineering, Pusan National University, Korea. His research interests include robotics, robot vision, localization and navigation systems.

Sungshin Kim received the B.S. and M.S degree in Electrical engineering from Yonsei University, Korea in 1984 and 1986 respectively, and the Ph.D. degree in Electrical engineering from Georgia Institute of Technology, USA in 1996. He is currently a professor in Electrical Engineering Department, Pusan National University. His research interests include fuzzy logic controls, neuro fuzzy systems, neural networks, robotics, signal analysis, and intelligent systems.