

Recursive Unscented Kalman Filtering based SLAM using a Large Number of Noisy Observations

Seongsoo Lee, Sukhan Lee*, and Dongsung Kim

Abstract: Simultaneous Localization and Map Building (SLAM) is one of the fundamental problems in robot navigation. The Extended Kalman Filter (EKF), which is widely adopted in SLAM approaches, requires extensive computation. The conventional particle filter also needs intense computation to cover a high dimensional state space with particles. This paper proposes an efficient SLAM method based on the recursive unscented Kalman filtering in an environment including a large number of landmarks. The posterior probability distributions of the robot pose and the landmark locations are represented by their marginal Gaussian probability distributions. In particular, the posterior probability distribution of the robot pose is calculated recursively. Each landmark location is updated with the recursively updated robot pose. The proposed method reduces filtering dimensions and computational complexity significantly, and has produced very encouraging results for navigation experiments with noisy multiple simultaneous observations.

Keywords: Real-time SLAM, recursive unscented Kalman filtering, stochastic SLAM, vision-based SLAM.

1. INTRODUCTION

Simultaneous localization and mapping (SLAM) is an essential technique that localizes a robot in an environmental map while estimating the map itself with the localized robot pose, simultaneously. It is an inherently complex problem since an error of the robot pose leads to an error of the map and vice versa [1,2]. Also, it should be operated in real-time.

One of the most well-established approaches to solve the SLAM problem is based on an assumption of Gaussian distributions for errors of the state and observations. Given this assumption, the extended Kalman filter (EKF) based approaches to SLAM [3-5] have been developed since the first introduction by Smith, Self, and Cheeseman [6]. There have been

several applications of these approaches in a number of different environments, such as indoors [7], underwater [8], and outdoors [9]. However, these approaches require heavy computation as the number of landmarks and their simultaneous observations increase because they have the complexity of $O(N^2)$, where N is the number of landmarks [3,6].

This scaling problem arises because each landmark is correlated to all the other landmarks. To overcome this problem, various research projects have been performed to optimize EKF based approaches to SLAM [10-12]. The decouple stochastic mapping (DSM) method [10] reduces the computational burden by dividing the global map into multiple overlapping sub-maps but only the single activated sub-map is updated at each sampling time. In [11], the compressed extended Kalman filter (CEKF) is applied to achieve a computation time of $O(N_L^2)$, where N_L is the number of landmarks in the local area. However, this approach still requires a computation time of $O(N_L^2 N^{2/3})$ when the robot moves from one local map to another. In [12], the constrained local sub-map filter (CLSF) is presented to periodically fuse a local sub-map into the global environment map. It can reduce the computational complexity of maintaining the global map estimation, but it still has the complexity of $O(N^2)$ from a global point of view because the number of sub-maps is still proportional to the number of landmarks.

Rao-Blackwellised particle filter (RBPF) is another

Manuscript received December 2, 2005; revised May 21, 2006; accepted July 18, 2006. Recommended by Editor Jae-Bok Song. This work is supported by the Intelligent Robotics Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea. This work is also supported in part by the Science and Technology Program of Gyeonggi Province as well as in part by Sungkyunkwan University.

Seongsoo Lee and Sukhan Lee are with the School of Information and Communication Engineering of Sungkyunkwan University, 300 ChunChun-Dong, JangAn-Gu, Suwon, KyungGi-Do 440-746, Korea (e-mails: {lss0703, lsh}@ece.skku.ac.kr).

Dongsung Kim is with the School of Electronic Engineering, Soongsil University, SangDo-Dong 511, DongJak-Gu, Seoul 156-743, Korea (e-mail: dongsung@ssu.ac.kr).

* Corresponding author.

attempt to reduce the scale problem by factoring the state variables, which is sampling over a subset of the state variables [13,14]. An application of RBPF to SLAM was introduced first by Murphy [15]. FastSLAM 1.0/2.0 [16,17], which are variants of RBPF, have a complexity of $O(K \log N)$ for integrating a single observation, where K is the number of particles used at each sampling time. Although FastSLAM 2.0 can even converge with a single particle for special cases [17], its computational complexity depends linearly on the number of particles, which is scaled with the environment size.

This paper proposes an efficient method for SLAM with a large number of landmarks and their simultaneous observations. We aim at developing a SLAM method using the recursive unscented Kalman filtering. The proposed method is based on a recursively marginalized Bayesian formula of SLAM to update the state, composed of the robot pose and landmark locations. The posterior probability distributions of the robot pose and landmark locations are represented by their marginal Gaussian probability distributions. In particular, the robot pose is updated by a recursive unscented Kalman filtering. Using this updated robot pose, the observed landmarks are individually updated. The proposed method can reduce filtering dimensionality significantly and can also reduce the computation requirements for filtering with respect to the state update.

This paper is organized as follows: Section 2 introduces the theoretical concept of the proposed method; Section 3 describes details of the method based on the recursive unscented Kalman filtering; Section 4 verifies the efficiency and localization accuracy of the experimental results in unstructured indoor environments, given multiple simultaneous observations. Section 5 concludes this paper.

2. THEORETICAL CONCEPT OF THE PROPOSED METHOD

A SLAM algorithm uses dead reckoning and relative observations to estimate the robot pose and to build and maintain an environmental map. The dead reckoning can be achieved by an odometry such as a wheel encoder while relative distances of the landmarks to the robot can be acquired by an external sensor such as a stereo camera. Those sensor data are formulated into two probabilistic models, the motion model and the observation model, to estimate the robot pose and the landmark locations. The motion model which predicts the state of the robot pose at time t , x_t , is given by the following probabilistic function:

$$p(x_t | x_{t-1}, u_t) \Leftrightarrow x_t = f(x_{t-1}, u_t) + w_t, \quad (1)$$

where u_t and w_t are robot control at time t and noise of

the robot control between time $t-1$ and time t , respectively. The observation model relating the robot pose and the observed landmark location m_{c_i} to the observation z_i is represented by the following probabilistic function:

$$p(z_i | x_t, m_{c_i}) \Leftrightarrow z_i = h_i(x_t, m_{c_i}) + v_i \quad (i=1, \dots, n), \quad (2)$$

where v_i is observation noise. For all landmark locations in a map, M , observed landmark locations $M_{c_1}^{c_n} = \{m_{c_1}, \dots, m_{c_n}\}$ are computed from multiple simultaneous observations at time t , $Z_t = \{z_1, \dots, z_n\}$, where n is the number of observations at time t . The unobserved landmark locations are denoted as M^{uob} .

In Bayesian sense, the posterior probability distributions of the robot pose and landmark locations for SLAM can be represented by a recursive formula with respect to time sequences [18,19]:

$$p(x_t, M | Z_t, Z^{t-1}, u^t) = \eta p(Z_t | x_t, M) \times \int p(x_t | x_{t-1}, u_t) p(x_{t-1}, M | Z^{t-1}, u^{t-1}) dx_{t-1}, \quad (3)$$

where η is a normalizing constant, $Z^t = \{Z_1, \dots, Z_t\}$ and $u^t = \{u_1, \dots, u_t\}$. In (3), the marginal probability distribution of the robot pose x_t represents its posterior probability distribution:

$$\begin{aligned} p(x_t | Z_t, Z^{t-1}, u^t) &= \int p(x_t, M_{c_1}^{c_n}, M^{uob} | Z_t, Z^{t-1}, u^t) dM \\ &= \int \left[\int p(x_t, M_{c_1}^{c_n}, M^{uob} | Z_t, Z^{t-1}, u^t) \right] dM_{c_1}^{c_n} \quad (4) \\ &= \int p(x_t, M_{c_1}^{c_n} | Z_t, Z^{t-1}, u^t) dM_{c_1}^{c_n}. \end{aligned}$$

Equation (4) is a recursive formula marginalizing (3) with respect to all individual landmark locations at specific time t , M while (3) is a recursive formula with respect to time sequences. In (4), it is not necessary to compute unobserved landmark locations, M^{uob} , because the calculation of $p(x_t | Z_t, Z^{t-1}, u^t)$ doesn't require them. Also, (4) can be rewritten by the following equations which are recursively marginalized with respect to observed landmark locations at time t , $M_{c_1}^{c_n}$:

$$\begin{aligned} &\int \left[\int p(x_t, M_{c_1}^{c_n} | Z_t, Z^{t-1}, u^t) dm_{c_1} \right] dM_{c_2}^{c_n} \\ &= \int \left[\int p(x_t, M_{c_2}^{c_n} | Z_t, Z^{t-1}, u^t) dm_{c_2} \right] dM_{c_3}^{c_n} \quad (5) \\ &\quad \vdots \end{aligned}$$

$$\begin{aligned}
&= \int \left[\int p(x_t, M_{c_{n-1}}^{c_n} | Z_t, Z^{t-1}, u^t) \right. \\
&\quad \left. dm_{c_{n-1}} \right] dm_{c_n} \\
&= \int p(x_t, m_{c_n} | Z_t, Z^{t-1}, u^t) dm_{c_n}.
\end{aligned}$$

Equation (5) allows the robot pose to be updated independently by each observation. It reduces the computation load dramatically because the filtering dimensionality is reduced from much higher dimensions composed of robot pose and all observed landmarks to the much smaller dimensions composed of the robot pose and a single landmark. For computing (5) efficiently, we make a reasonable assumption that the robot pose and all landmark locations are mutually independent. Using these assumptions, (5) is rewritten by the following equations (for details of the derivation, refer to the Appendix A):

$$\begin{aligned}
&\int p(x_t, M_{c_1}^{c_n} | Z_t, Z^{t-1}, u^t) dM_{c_1}^{c_n} \\
&= \int \left[\underbrace{\int \eta_1 p(z_1 | x_t, m_{c_1}) P_{X_1} dm_{c_1}}_{=p(x_t, m_{c_1} | z_1, B)} \right] A_1 dM_{c_2}^{c_n} \\
&= \int \left[\underbrace{\int \eta_2 p(z_2 | x_t, m_{c_2}) P_{X_2} dm_{c_2}}_{=p(x_t, m_{c_2} | z_2^1, B)} \right] A_2 dM_{c_3}^{c_n} \\
&\quad \vdots \\
&= \int \left[\underbrace{\int \eta_{n-1} p(z_{n-1} | x_t, m_{c_{n-1}}) P_{X_{n-1}} dm_{c_{n-1}}}_{=p(x_t, m_{c_{n-1}} | z_1^{n-1}, B)} \right] A_{n-1} dm_{c_n} \\
&= \int \underbrace{\eta_n p(z_n | x_t, m_{c_n}) P_{X_n} dm_{c_n}}_{=p(x_t, m_{c_n} | z_1^n, B)},
\end{aligned} \tag{6}$$

where η_i is a normalizing constant and

$$\begin{cases}
P_{X_1} = p(x_t | B) p(m_{c_1} | B) & (i=1), \\
P_{X_i} = p(x_t | z_1^{i-1}, B) p(m_{c_i} | B) & (i>1), \\
A_1 = p(z_2^n | x_t, m_{c_2}^{c_n}) p(m_{c_2}^{c_n} | B), \\
A_2 = p(z_3^n | x_t, m_{c_3}^{c_n}) p(m_{c_3}^{c_n} | B), \\
\quad \vdots \\
A_{n-1} = p(z_n | x_t, m_{c_n}) p(m_{c_n} | B), \\
B = Z^{t-1}, u^t.
\end{cases}$$

Similarly to deriving (6), the posterior probability

distribution of the observed landmark location m_{c_i} is represented by its marginal probability distribution as a recursive formula:

$$\begin{aligned}
&p(m_{c_i} | Z_t, Z^{t-1}, u^t) \\
&= \iint \left[\int p(x_t, M_{c_1}^{c_n}, M^{uob} | Z_t, Z^{t-1}, u^t) dM^{uob} \right] \\
&\quad dx_t dM' \\
&= \iint \left[\int p(x_t, M_{c_1}^{c_n} | Z_t, Z^{t-1}, u^t) dM' \right] dx_t \\
&= \int p(x_t, m_{c_i} | Z_t, Z^{t-1}, u^t) dx_t,
\end{aligned} \tag{7}$$

where $M' = M_{c_1}^{c_n} - m_{c_i}$. Equation (7) can be rewritten by the following equation, given the assumptions used in (6) (for details of the derivation, refer to Appendix B):

$$\begin{aligned}
&\int p(x_t, m_{c_i} | Z_t, Z^{t-1}, u^t) dx_t \\
&= \eta_i \int \left(\begin{array}{l} p(z_i | x_t, m_{c_i}, Z^{t-1}, u^t) \\ \times p(x_t | Z_t', Z^{t-1}, u^t) \\ \times p(m_{c_i} | Z_t', Z^{t-1}, u^t) \end{array} \right) dx_t \\
&= \eta_i \int \left(\begin{array}{l} p(z_i | x_t, m_{c_i}) \\ \times p(x_t | Z_t', Z^{t-1}, u^t) \\ \times p(m_{c_i} | Z^{t-1}, u^{t-1}) \end{array} \right) dx_t,
\end{aligned} \tag{8}$$

where $Z_t' = Z_t - z_i$ and η_i is a normalizing constant. Equation (8) permits each landmark to be updated independently by its observation with $p(x_t | Z_t', Z^{t-1}, u^t)$. To simplify the computation of (8), $p(x_t | Z_t', Z^{t-1}, u^t)$ is replaced by the posterior probability distribution of the robot pose, $p(x_t | Z_t, Z^{t-1}, u^t)$. Equation (8) is then rewritten as follows:

$$\approx \eta_i \int \left(\begin{array}{l} p(z_i | x_t, m_{c_i}) p(x_t | Z_t, Z^{t-1}, u^t) \\ \times p(m_{c_i} | Z^{t-1}, u^{t-1}) \end{array} \right) dx_t. \tag{9}$$

Given a large number of observations, $p(x_t | Z_t, Z^{t-1}, u^t)$ is almost equivalent to $p(x_t | Z_t', Z^{t-1}, u^t)$ because the robot pose can be converged without using all of the observations. Thus, (9) can be used to update the landmark locations if there are a lot of simultaneous observations.

3. RECURSIVE UNSCENTED KALMAN FILTERING BASED SLAM

The posterior probability distribution for SLAM can be simplified by (6) and (9) with the assumptions of robot pose and all landmark location independences between states. Their reduced dimensionalities allow the complexity of the filtering to be reduced efficiently. The computation of the two equations is performed by unscented Kalman filters (UKFs) and (6) is computed by a recursive unscented Kalman filter because the UKF is more robust against the non-linearity of a system model than EKF [20]. Please note that the same robot pose updated by (6) is used to calculate all observed landmark locations with (9). Assuming Gaussian noise, the algorithm for computing (6) and (9) is stated as follows:

- Step 1:** Estimate $\eta_i p(z_i | x_t, m_{c_i}) p(x_t | z_1^i, Z^{t-1}, u^t) p(m_{c_i} | Z^{t-1}, u^t)$, represented by joint Gaussian distribution, for updating recursively the robot pose x_t . In case of $i = 1$, the prior Gaussian distribution of x_t , $p(x_t | Z^{t-1}, u^t)$, is propagated from the motion model represented in (1). The prior Gaussian distribution of m_{c_i} , $p(m_{c_i} | Z^{t-1}, u^t)$, is equivalent to that of the previous state.
- Step 2:** Obtain the marginal probability distribution of x_t , $p(x_t | z_1^i, Z^{t-1}, u^t)$, represented by a Gaussian distribution, from $p(x_t, m_{c_i} | z_1^i, Z^{t-1}, u^t)$ calculated in Step 1. Here, $p(x_t | z_1^i, Z^{t-1}, u^t)$ is recursively used as the prior probability distribution for calculating $p(x_t, m_{c_{i+1}} | z_1^{i+1}, Z^{t-1}, u^t)$.
- Step 3:** Repeat Steps 1~2 until $i = n$. The posterior probability distribution of the robot pose, $p(x_t | Z_t, Z^{t-1}, u^t)$, is obtained from n iterations.
- Step 4:** Estimate $\eta_i p(z_i | x_t, m_{c_i}) p(x_t | Z_t, Z^{t-1}, u^t) p(m_{c_i} | Z^{t-1}, u^{t-1})$ represented by joint Gaussian distribution for updating the observed landmark m_{c_i} . The same $p(x_t | Z_t, Z^{t-1}, u^t)$ is used for updating all observed landmarks.
- Step 5:** Obtain the marginal Gaussian probability distribution of m_{c_i} , $p(m_{c_i} | Z_t, Z^{t-1}, u^t)$, from $\eta_i p(z_i | x_t, m_{c_i}) p(x_t | Z_t, Z^{t-1}, u^t) p(m_{c_i} | Z^{t-1}, u^{t-1})$ calculated in Step 4.
- Step 6:** Repeat Steps 4~5 until all observed landmarks are individually updated.

Here, $p(z_i | x_t, m_{c_i}) p(x_t | z_1^i, Z^{t-1}, u^t) p(m_{c_i} | Z^{t-1}, u^t)$

and $p(z_i | x_t, m_{c_i}) p(x_t | Z_t, Z^{t-1}, u^t) p(m_{c_i} | Z^{t-1}, u^{t-1})$ are estimated by UKF [20,21]. Because both equations are similar except the terms $p(x_t | z_1^i, Z^{t-1}, u^t)$ and $p(x_t | Z_t, Z^{t-1}, u^t)$, the computation of them is formulated with a notation X_i denoting the random vector $[x_t, m_{c_i}]^T$. X_i is represented by uncorrelated joint Gaussian distribution with the mean \hat{X}_i and covariance P_{X_i} , as follows:

$$N(\hat{X}_i, P_{X_i}) \Leftrightarrow \begin{cases} \hat{X}_i = [\hat{x}_t, \hat{m}_{c_i}]^T, \\ P_{X_i} = \begin{bmatrix} P_{x_t x_t} & 0 \\ 0 & P_{m_{c_i} m_{c_i}} \end{bmatrix}, \end{cases} \quad (10)$$

where $N(\hat{\alpha}, P_{\alpha})$ denotes a Gaussian distribution with the mean $\hat{\alpha}$ and covariance P_{α} . The prior Gaussian distribution of x_t and m_{c_i} are represented by $N(\hat{x}_t, P_{x_t x_t})$ and $N(\hat{m}_{c_i}, P_{m_{c_i} m_{c_i}})$, respectively. To reduce complexity of the UKF computation, non-augmented X_i is used as in [21,22]. The updated X_i^+ with $N(\hat{X}_i^+, P_{X_i}^+)$ is calculated by the following equations:

$$\begin{aligned} \hat{X}_i^+ &= \hat{X}_i + K_i(z_i - \hat{z}_i), \\ P_{X_i}^+ &= P_{X_i} - K_i(P_{z_i z_i} + R_i)K_i^T, \\ K_i &= P_{X_i z_i}(P_{z_i z_i} + R_i)^{-1}, \end{aligned} \quad (11)$$

where K_i is a Kalman gain and z_i is an observation with a covariance R_i . $P_{z_i z_i}$ is the covariance of the predicted observation \hat{z}_i while $P_{X_i z_i}$ is the predicted cross-correlation matrix of X_i and z_i . In (11), $P_{z_i z_i}$ and $P_{X_i z_i}$ are calculated using the unscented transformation (UT) [21,22] with X_i and z_i represented by Gaussian distributions, through the observation model represented by (2). The UT is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation, and forms the core of the UKF algorithm. It chooses a set of points (so called *sigma* points and their weights) based on the mean and covariance of a random variable. The chosen points are fed into a nonlinear model (e.g., the measurement function $h(\cdot)$ in (2)), and the weighted statistics of the transformed points make an estimate of the nonlinearly transformed mean and covariance (for more details, refer to [21,22]). In the case of updating the robot pose recursively, the initial prior Gaussian distribution of the robot pose x_t ,

$N(\hat{x}_t, P_{x_t x_t})$, is calculated using the UT with the prior Gaussian distribution of the previous robot pose x_{t-1} and the robot control at time t , and u_t , through the motion model represented by (1). The updated X_i^+ is represented by the correlated joint Gaussian distribution, $N(\hat{X}_i^+, P_{X_i^+}^+)$, as follows:

$$\hat{X}_i^+ = [\hat{x}_t^+, \hat{m}_{c_i}^+]^T, P_{X_i^+}^+ = \begin{bmatrix} P_{x_t x_t}^+ & P_{x_t m_{c_i}}^+ \\ P_{m_{c_i} x_t}^+ & P_{m_{c_i} m_{c_i}}^+ \end{bmatrix}. \quad (12)$$

Now, the marginal probability distributions of the robot pose x_t and the landmark location m_{c_i} are computed with the joint Gaussian distribution represented by (12) and are $N(\hat{x}_t^+, P_{x_t x_t}^+)$ and $N(\hat{m}_{c_i}^+, P_{m_{c_i} m_{c_i}}^+)$, respectively.

Table 1 provides a summary of the computational complexity of the proposed method and the conventional SLAM algorithms with respect to the number of landmarks. The computation time of EKF based SLAM and CEKF based SLAM depends on the number of all landmarks in a map or the number of landmarks in a local area regardless of the number of observations at each frame. FastSLAM requires $O(K \log N)$ for integrating a measurement by the maximum likelihood data association, where K and N are number of particles and number of landmarks, respectively. If it integrates n simultaneous observations, the computation time will be $O(Kn \log N)$. Because the efficiency of FastSLAM depends crucially on K , the number of the particles needs to be kept within reason. On the contrary, the complexity of the proposed method is proportional to only the number of observed landmarks at each frame, and is $O(2n)$ for integrating n simultaneous observations.

Table 1. Computational complexity of different SLAM algorithms with N landmarks in a map, N_L^2 landmarks in a specific local area, and n simultaneous observations at each frame. FastSLAM is a particle filter approach (K particles).

	Required complexity for landmark update
EKF based SLAM	$O(N^2)$
CEKF based SLAM	$O(N_L^2)$ in a local area, $O(N_L N^2)$ in a global area
FastSLAM	$O(K \log N)$ or $O(Kn \log N)$
Proposed method	$O(2n)$

The new landmark location $m_{c_i}^{new}$, first measured by the observation z_i , is initialized using the inverse observation model. The inverse observation model is represented as follows:

$$p(m_{c_i}^{new} | x_t, z_i) \Leftrightarrow \begin{cases} m_{c_i}^{new} = h_i^{-1}(x_t, z_i), \\ (i = 1, \dots, n). \end{cases} \quad (13)$$

The initial Gaussian distribution of new landmark location $m_{c_i}^{new}$ is calculated using the UT with the updated robot pose x_t and the observation z_i represented by a Gaussian distribution, through (13).

4. EXPERIMENTAL RESULTS

To verify its validity in real applications, the proposed method was tested in two different experimental settings to accommodate varying floor conditions and illuminations. Both experimental settings are highly textured environments containing numerous landmarks. The comparison between the proposed method and FastSLAM 2.0 has been performed with two different odometric noises for each experimental setting.

The robot used in the experiments is a PowerBot-AGV with a SICK laser range finder and two BumbleBee stereo cameras from Point Grey Research, as is shown in Fig. 1. Please note that only the single stereo camera on the robot platform is used for the SLAM experiments. Two experiments have been performed. The first one is a 16.3m trajectory run with 215 images, and the second one is a 96.7m trajectory run with 403 images. Both runs are driven manually to achieve the ground truth data. For the first experimental setting, the final location is measured by the motion capture device (Vicon 640, MCam2).

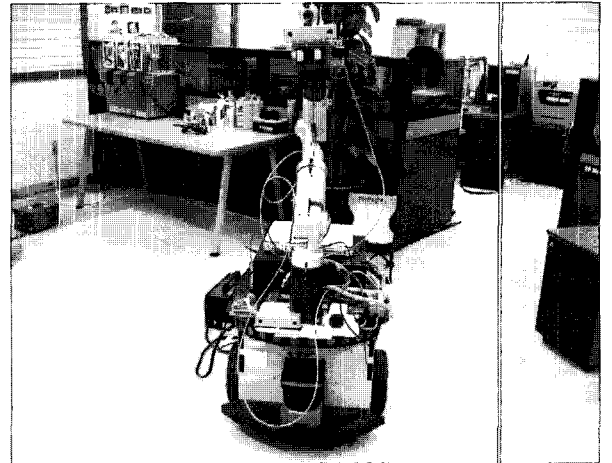
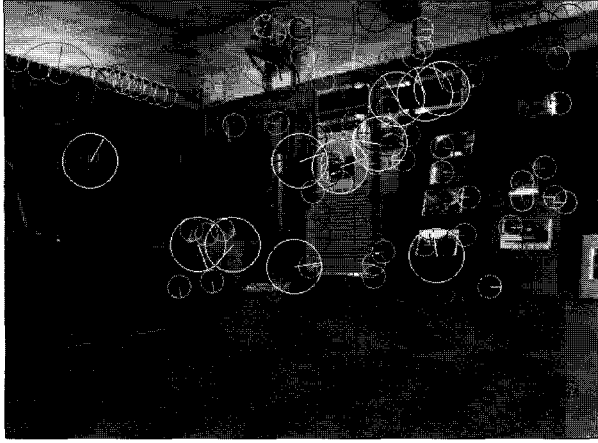
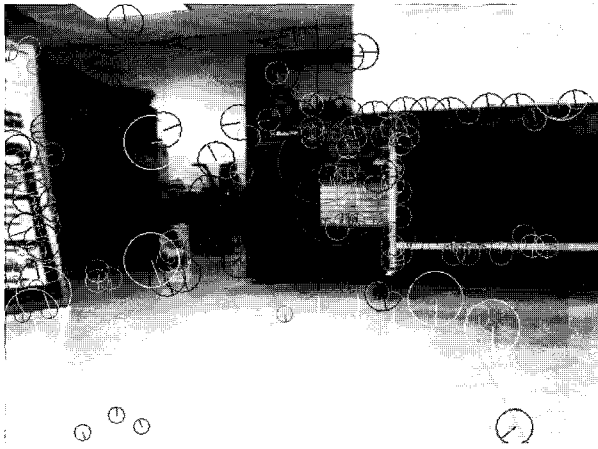


Fig. 1. PowerBot-AGV with a SICK laser range finder and two BumbleBee stereo cameras.



(a) First experimental setting.



(b) Second experimental setting.

Fig. 2. SIFT features extracted in sample images (640×480).

For the second experimental setting, the robot is returned to the initial position (0,0) as accurate as possible by an operator.

For the landmarks, SIFT features [23] are extracted as shown in Fig. 2. SIFT features are marked with circles having various sizes which represent the SIFT scale. The line inside these circles stands for the SIFT orientation and the centre of circles is representative of the SIFT location.

Fig. 3 shows a schematic diagram of the robot pose in the process of observing a landmark. Given a robot control $u_t = [\Delta d_t, \Delta \varphi_t]^T$ in terms of a translation Δd_t and a rotation $\Delta \varphi_t$, a new state for the robot pose can be updated. The following equation is used to obtain a discrete-time robot motion model:

$$\begin{bmatrix} r_{x,t} \\ r_{y,t} \\ r_{\theta,t} \end{bmatrix} = \begin{bmatrix} r_{x,t-1} + \Delta d_t \cos(r_{\theta,t-1} + 0.5\Delta \varphi_t) \\ r_{y,t-1} + \Delta d_t \sin(r_{\theta,t-1} + 0.5\Delta \varphi_t) \\ r_{\theta,t-1} + \Delta \varphi_t \end{bmatrix}, \quad (14)$$

where the state of the robot pose denotes $x_t = [r_{x,t},$

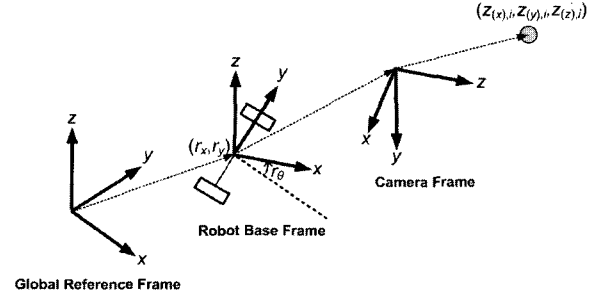


Fig. 3. Robot and observation kinematics.

$r_{y,t}, r_{\theta,t}]^T$, which comprises Cartesian coordinates $\{r_{x,t}, r_{y,t}\}$ and its orientation $r_{\theta,t}$ with respect to the global reference frame. The translation Δd_t and rotation $\Delta \varphi_t$ are estimated by an odometer, where their additive noises are assumed by uncorrelated Gaussian noises with zero mean, and variances σ_d^2 and σ_φ^2 . Thus, the covariance of u_t , σ_u^2 , can be written as follows:

$$\sigma_u^2 = \begin{bmatrix} \sigma_d^2 & 0 \\ 0 & \sigma_\varphi^2 \end{bmatrix}. \quad (15)$$

To predict σ_u^2 properly is not easy because of non-systematic errors such as slippage.

The following equation, which transforms from the 3D landmark location of the global reference frame to that of the camera frame, is used as the observation model:

$$\begin{bmatrix} z_{(x),i} & z_{(y),i} & z_{(z),i} \end{bmatrix}^T = \begin{bmatrix} \Phi_1 \sin(r_{\theta,t}) + \Phi_2 \sin(r_{\theta,t}) + T_{(y)} + v_{(x),i} \\ {}^w m_{(z),c_i} - T_{(z)} + v_{(y),i} \\ \Phi_1 \cos(r_{\theta,t}) - \Phi_2 \sin(r_{\theta,t}) - T_{(x)} + v_{(z),i} \end{bmatrix}, \quad (16)$$

where $z_i = \{z_{(x),i}, z_{(y),i}, z_{(z),i}\}$ represents the 3D landmark location with respect to the observation (camera) frame, whereas ${}^w m_{c_i} = \{{}^w m_{(x),c_i}, {}^w m_{(y),c_i}, {}^w m_{(z),c_i}\}$ represents the 3D landmark location with respect to the global reference frame. Φ_1 is $(-{}^w m_{(x),c_i} + r_{x,t})$ and Φ_2 is $({}^w m_{(y),c_i} - r_{y,t})$. Here, $v_{(x),i}$, $v_{(y),i}$ and $v_{(z),i}$ are observation noises assumed by additive uncorrelated Gaussians with zero mean and their individual variances, denoted as $\sigma_{v_{(x),i}}^2$, $\sigma_{v_{(y),i}}^2$, and $\sigma_{v_{(z),i}}^2$ respectively. $T_{(x)}$, $T_{(y)}$, and $T_{(z)}$

specify the location of the sensor with respect to the robot base frame.

In our experiment, the variances with respect to additive noises used for filtering are given as follows:

$$\begin{cases} \sigma_d^2 = (\beta \times \Delta d_t)^2, \\ \sigma_\varphi^2 = (0.5)^2, \\ \text{if } z_{(z),i} < 4000, \begin{cases} \sigma_{v_{(x),i}}^2 = 50^2 \\ \sigma_{v_{(y),i}}^2 = 50^2 \\ \sigma_{v_{(z),i}}^2 = 100^2, \end{cases} \\ \text{otherwise,} \begin{cases} \sigma_{v_{(x),i}}^2 = (\gamma \times 50 \times \frac{z_{(z),i}}{4000})^2 \\ \sigma_{v_{(y),i}}^2 = (\gamma \times 50 \times \frac{z_{(z),i}}{4000})^2 \\ \sigma_{v_{(z),i}}^2 = (\gamma \times 100 \times \frac{z_{(z),i}}{4000})^2, \end{cases} \end{cases} \quad (17)$$

where β is the parameter for adjusting levels of a variance in terms of a translation Δd_t in the robot control.

The proposed method is compared with the FastSLAM 2.0 algorithm using simultaneous observations proposed by [24]. In the comparison experiments, the FastSLAM 2.0 uses one particle with a single map because it can produce similar performance to the FastSLAM 2.0 with multiple particles in unambiguous situations [17]. For both algorithms, the map structure is organized spatially as a hash table instead of a k-d tree used in the FastSLAM 2.0 because a hash table is more efficient than a k-d tree for keeping a single map. For each experiment run, the performance of both algorithms was evaluated with two differing levels of variance associated with robot control on the same sensor data: ($\beta = 0.05$) and ($\beta = 0.2$). Both experiments are evaluated with the following three performance indicators: 1) the root mean square error (RMSE) of the robot position, 2) the constructed map at the end of exploration, and 3) the average computational time at each step.

Figs. 4 and 5 show the root mean square error (RMSE) between the ground truth position of the robot and its estimated positions by both the proposed method and the FastSLAM 2.0 for two experiments. Fig. 4(a) indicates an estimated position error when ($\beta = 0.05$) while Fig. 4(b) demonstrates an estimated position error when ($\beta = 0.2$). For the proposed method, the robot position error is clearly bounded and does not diverge in both experiments. Although both algorithms can estimate the robot pose accurately when odometric noise is low, the FastSLAM 2.0 may

fail catastrophically with a high odometric error. In experiment 2, the maximum error of the proposed method was about 22cm as shown in Fig. 5(b), but that of the FastSLAM 2.0 was 70 cm. Moreover, the proposed method shows better performance than the FastSLAM 2.0 for all four cases. These results prove that the proposed method is more robust against errors on variances. Although the variance parameter is a kind of gain which should be adjusted in the performance tuning process, it is not easy to tune the noise parameters suitably in various experiments. Thus, robustness to a variance error can be an advantage of an algorithm.

Both algorithms have better performance in the first experimental setting because SLAM was performed there with a shorter sampling time than that of the second experimental setting. The longer sampling time is, the larger the robot pose error estimated only with an odometer becomes. Generally, a correctly matched rate between landmarks and observations is smaller in the longer sampling time than the shorter one. This reduced number of valid observations may make filtering performance worse. It should be noted that there were mismatched observations in our experiments but not completely mismatched ones because the SIFT is a local feature. In spite of such false observations, the proposed method can estimate the robot position without diverging robot position error because the false ones are relatively a small population compared to the correct ones. Moreover, the mismatched landmarks don't affect the others because all landmarks are independently and individually updated in the proposed method. If a certain landmark within the field of view is not observed more than the predefined number, it is removed from a map.

Figs. 6 and 7 illustrate the maps generated by both algorithms at the end of exploration of the second experimental place with a quadrangle shape. 3D SIFT landmarks are designated by black points and estimated robot trajectories are drawn with red lines. All maps are not post-processed to remove noise. Fig. 6 shows the maps constructed by the FastSLAM 2.0 while Fig. 7 depicts the maps built by the proposed method. The map generated by the FastSLAM 2.0 for a large odometric error, as shown in Fig. 6(b), has large distortions, and is difficult to use for navigation. More than 14000 and 20000 landmarks were built in the first and the second experiments, respectively.

Table 2 shows the average computation time for each step in experiment 2 of the high odometric error. The computation time is measured on a 3.0 GHz Pentium IV computer with 1GBytes RAM. The matched observations are the average number of simultaneous observations matched to landmarks in a map at each frame. The landmark processing time represents average computing time for extracting and

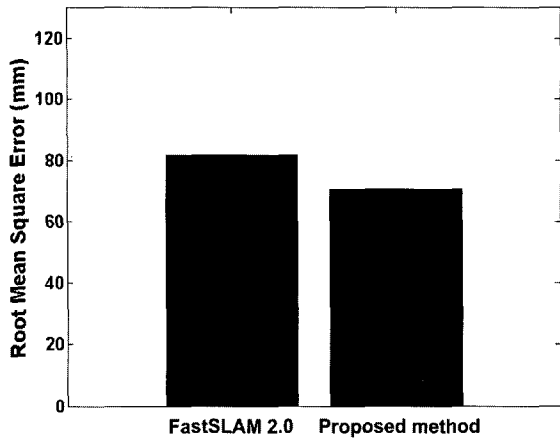
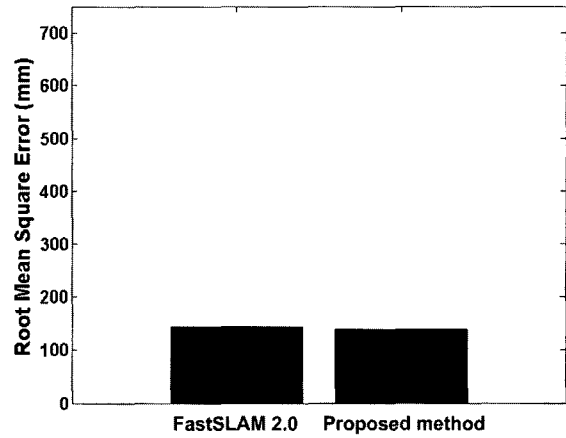
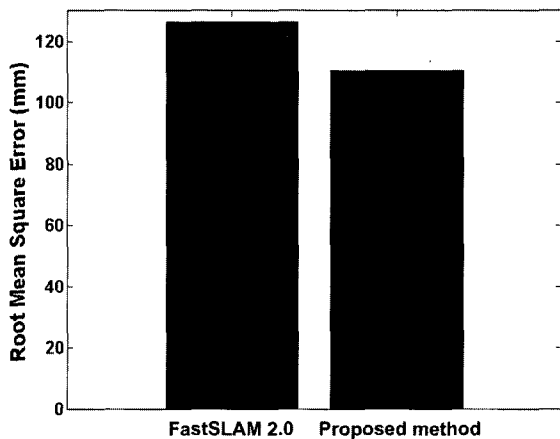
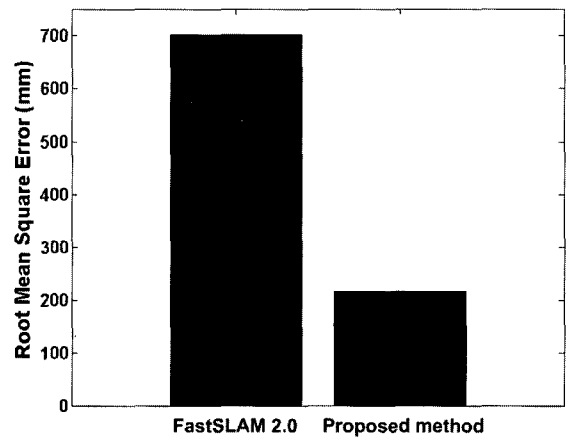
(a) Low odometric error ($\beta = 0.05$).(a) Low odometry error ($\beta = 0.05$).(b) High odometric error ($\beta = 0.2$).(b) High odometry error ($\beta = 0.2$).

Fig. 4. Experiment 1: Estimation error of the FastSLAM 2.0 and the proposed method with varying levels of odometric noise.

Fig. 5. Experiment 2: Estimation error of the FastSLAM 2.0 and the proposed method with varying levels of odometric noise.

matching of the SIFT features at each frame. The landmark initialization and filtering time indicate average computing time for the sum of initialization time for new landmarks and filtering time with matched observations at each step. The filtering time per observation means computing time for integrating a matched observation in a state update. In this experiment, both the FastSLAM 2.0 and the proposed method take similar computation time for extracting and matching of the SIFT features while the proposed method takes about twice as long for filtering. As seen in Table II, the filtering time of the proposed method is small enough so that the method can be used as a real-time SLAM algorithm if feature processing time is reduced by either optimizing the SIFT or using other features that can be extracted efficiently.

5. CONCLUSIONS

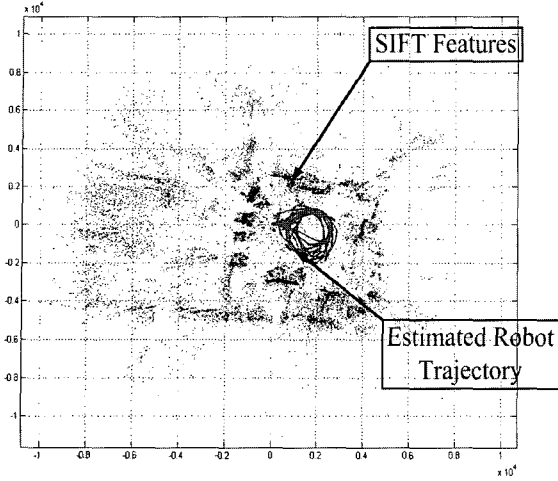
This paper has presented an efficient SLAM method with a large number of simultaneous observations. The proposed method estimates the

robot pose with a recursive unscented Kalman filter. The estimated robot pose is utilized to update landmark locations. The reduction in the filtering dimensionality enables the SLAM method to be performed in real-time, even though there are enormous landmarks in a map. The proposed method is evaluated with the two experiments in terms of efficiency and localization accuracy. For 3D landmarks, 14000 SIFT features and 20000 SIFT features have been extracted and estimated for the two experiments. The experiments produce robust results against the erroneous levels of noise. Also, the results prove that it is possible to build an accurate map without keeping the correlation between landmarks.

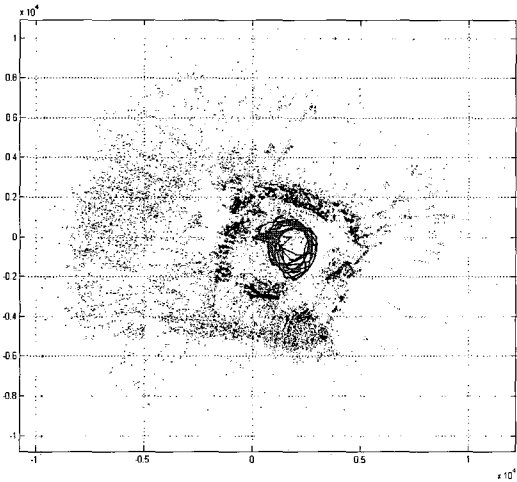
For future research, more efficient and robust features will be developed to reduce computing time because most of the computing time is consumed by the feature extraction and matching.

APPENDIX A

Equation (6) can be derived directly from (5). Using Bayes rule and the definition of conditional



(a) Low odometric error.

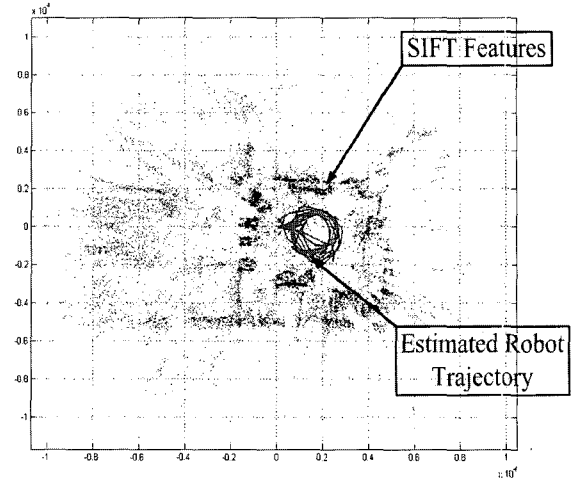


(b) High odometric error.

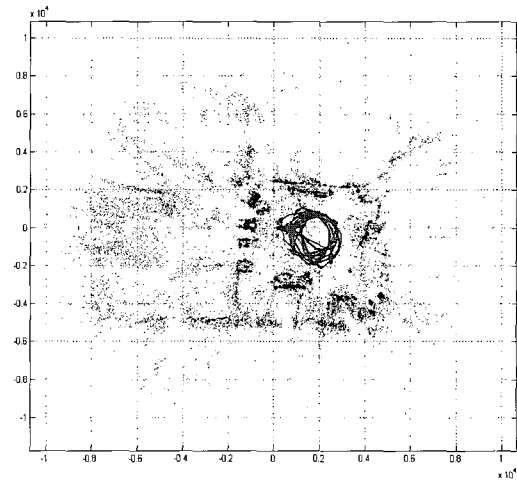
Fig. 6. Constructed maps with the FastSLAM 2.0 at the end of exploration of the second experimental place.

probability, (5) can be rewritten as the following equations:

$$\begin{aligned}
 & \int p(x_t, M_{c_1}^{c_n} | Z_t, Z^{t-1}, u^t) dM_{c_1}^{c_n} \\
 &= \int p(x_t, M_{c_1}^{c_n} | z_1, z_2^n, Z^{t-1}, u^t) dM_{c_1}^{c_n} \\
 & \text{(Bayes rule)} \\
 &= \eta \int \left(\frac{p(z_1, z_2^n | x_t, M_{c_1}^{c_n}, B)}{\times p(x_t, M_{c_1}^{c_n} | B)} \right) dM_{c_1}^{c_n} \\
 &= \eta \int \left(\frac{p(z_1, z_2^n | x_t, m_{c_1}, M_{c_2}^{c_n}, B)}{\times p(x_t, m_{c_1}, M_{c_2}^{c_n} | B)} \right) dM_{c_1}^{c_n} \\
 & \text{(Conditional probability definition)} \\
 &= \eta \int \left(\frac{p(z_1 | z_2^n, x_t, C, B) p(x_t | C, B)}{\times p(m_{c_1} | M_{c_2}^{c_n}, B) \times D} \right) dM_{c_1}^{c_n},
 \end{aligned} \tag{18}$$



(a) Low odometric error.



(b) High odometric error.

Fig. 7. Constructed maps with the proposed method at the end of exploration of the second experimental place.

Table 2. Computation time for each step in the experiment 2 (high odometric error).

	FastSLAM 2.0	Proposed method
Matched observations	77.65	119.55
Landmark processing time	682 ms	689 ms
Landmark initialization & Filtering time	17 ms	42 ms
Filtering time per observation	0.112 ms	0.239 ms

where $B = Z^{t-1}, u^t, C = m_{c_1}, M_{c_2}^{c_n}, Z_t = z_1^n = \{z_1, \dots, z_n\}$, and $D = p(z_2^n | x_t, m_{c_1}, M_{c_2}^{c_n}, B) p(M_{c_2}^{c_n} | B)$. Here, $M_{c_1}^{c_n} = \{m_{c_1}, \dots, m_{c_n}\}$ represent a set of observed landmarks

at time t while η means a normalizing constant. Assuming that all landmarks are independent with the previous states, $p(M_{c_2}^{c_n} | B)$ of D in (18) can be rewritten as the following equation:

$$\begin{aligned} p(M_{c_2}^{c_n} | B) &= p(m_{c_2} | B)p(M_{c_3}^{c_n} | B) \\ &\quad \vdots \\ &= p(m_{c_2} | B)p(m_{c_3} | B) \cdots p(m_{c_n} | B). \end{aligned} \quad (19)$$

The last equation of (18) can be simplified by assuming that z_1 is associated with x_t and m_{c_1} whereas x_t , m_{c_1} , and $M_{c_2}^{c_n}$ are independent.

$$\eta \int \left(\frac{p(z_1 | x_t, m_{c_1})p(x_t | B)}{\times p(m_{c_1} | B) \times D} \right) dM_{c_1}^{c_n}. \quad (20)$$

Here, (20) is represented as two integrals:

$$\int \left[\int E_1 dm_{c_1} \right] \times D dM_{c_2}^{c_n}, \quad (21)$$

where $E_1 = p(z_1 | x_t, m_{c_1})p(x_t | B)p(m_{c_1} | B) = \frac{1}{\eta_1} p(x_t, m_{c_1} | z_1, B)$ and η_1 is a normalizing constant.

Also, (21) is rewritten by noting that the multiple observations at time t , $z_1^n = \{z_1, \dots, z_n\}$ are generally independent.

$$\int \left(\frac{\left[\int E_1 dm_{c_1} \right] p(z_2^n | x_t, M_{c_2}^{c_n}, B)}{\times p(M_{c_2}^{c_n} | B)} \right) dM_{c_2}^{c_n}. \quad (22)$$

Now, $p(x_t | z_1, B)$ is calculated by marginalizing E_1 with respect to the landmark location, m_{c_1} .

$$\begin{aligned} p(x_t | z_1, B) &= \int \eta_1 E_1 dm_{c_1} \\ &= \int p(x_t, m_{c_1} | z_1, B) dm_{c_1}. \end{aligned} \quad (23)$$

Assuming that $z_1^n = \{z_1, \dots, z_n\}$ are generally independent, $p(z_2^n | x_t, M_{c_2}^{c_n}, B)$ of (22) can be rewritten as:

$$p(z_2 | x_t, M_{c_2}^{c_n}, B)p(z_3^n | x_t, M_{c_2}^{c_n}, B). \quad (24)$$

Here, (24) is simplified by noting that z_2 is associated with x_t and m_{c_2} whereas z_3^n is associated with x_t and $M_{c_3}^{c_n}$.

$$p(z_2 | x_t, m_{c_2})p(z_3^n | x_t, M_{c_3}^{c_n}, B). \quad (25)$$

Using (23) and (25), (22) is rewritten as:

$$\int \left(\frac{\left[\int E_2 dm_{c_2} \right] p(z_3^n | x_t, M_{c_3}^{c_n})}{\times p(M_{c_3}^{c_n} | B)} \right) dM_{c_3}^{c_n}, \quad (26)$$

where $E_2 = p(z_2 | x_t, m_{c_2})p(x_t | z_1, B)p(m_{c_2} | B) = \frac{1}{\eta_2}$

$p(x_t, m_{c_2} | z_1^2, B)$ and η_2 is a normalizing constant.

Also, $p(x_t | z_1^2, B)$ can be calculated by marginalizing E_2 with respect to the landmark location, m_{c_2} .

$$\begin{aligned} p(x_t | z_1^2, B) &= \int \eta_2 E_2 dm_{c_2} \\ &= \int p(x_t, m_{c_2} | z_1^2, B) dm_{c_2}. \end{aligned} \quad (27)$$

In the same manner, $p(x_t | z_1^n, B)$ is obtained from calculating (28).

$$\begin{aligned} p(x_t | z_1^n, B) &= \int \eta_n E_n dm_{c_n} \\ &= \int p(x_t, m_{c_n} | z_1^n, B) dm_{c_n}, \end{aligned} \quad (28)$$

where $E_n = p(z_n | x_t, m_{c_n})p(x_t | z_1^{n-1}, B)p(m_{c_n} | B) =$

$\frac{1}{\eta_n} p(x_t, m_{c_n} | z_1^n, B)$ and η_n is a normalizing constant.

Equation (28) is equivalent to the marginal or posterior probability distribution of the robot pose represented by the last equation of (6).

APPENDIX B

Using Bayes rule, the last equation of (7) is rewritten as:

$$\begin{aligned} &\int p(x_t, m_{c_i} | Z_t, Z^{t-1}, u^t) dx_t \\ &= \eta_i \int \left(\frac{p(z_i | x_t, m_{c_i}, Z^{t-1}, u^t)}{\times p(x_t, m_{c_i} | Z_t', Z^{t-1}, u^t)} \right) dx_t, \end{aligned} \quad (29)$$

where $Z_t' = Z_t - z_i$. η_i is a normalizing constant.

Assuming that the state, composed of the robot pose x_t and all landmark locations $M = \{M_{c_1}^{c_n}, M^{uob}\}$, is independent with the previous states, and that the observation z_t is associated with x_t and m_{c_i} , (29) is simplified as:

$$\eta_i \int \left(\frac{p(z_i | x_t, m_{c_i})p(x_t | Z_t', Z^{t-1}, u^t)}{\times p(m_{c_i} | Z_t', Z^{t-1}, u^t)} \right) dx_t. \quad (30)$$

Equation (30) can be rewritten as the following equation because the robot control u_t doesn't provide new information about m_{c_i} without the latest observation z_i , and the other latest observations Z_t' don't observe m_{c_i} :

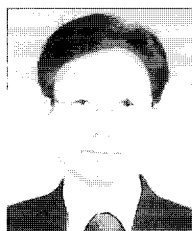
$$\eta_i \int \left(\begin{array}{l} p(z_i | x_t, m_{c_i}) p(x_t | Z_t', Z^{t-1}, u^t) \\ \times p(m_{c_i} | Z^{t-1}, u^{t-1}) \end{array} \right) dx_t. \quad (31)$$

Here, (31) is equivalent to the last equation of (8).

REFERENCES

- [1] D. Kortenkamp, R. P. Bonasso, and R. Murphy, *AI-based Mobile Robots: Case Studies of Successful Robot Systems*, MIT Press, 1998.
- [2] C. Thorpe and H. F. Durrant-Whyte, "Field robots," *Proc. of the 10th Int. Symp. of Robotics Research*, Lome, Victoria, Australia, Springer-Verlag, November 2001.
- [3] M. W. M. Gamin Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 229-241, June 2001.
- [4] J. K. Uhlmann, S. J. Julier, and M. Csorba, "Nondivergent simultaneous map building and localization using covariance intersection," *Proc. of SPIE*, Orlando, FL, vol. 3087, pp. 2-11, April 1997.
- [5] W. D. Renken, "Concurrent localization and map building for mobile robots using ultrasonic sensors," *Proc. of IEEE Int. Conf. Intell. Robots Syst.*, vol. 3, pp. 2192-2197, July 1993.
- [6] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Autonomous Robot Vehicles*, Springer-Verlag, pp. 167-193, 1990.
- [7] J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and Localization for an autonomous mobile robot," *Proc. of IEEE Int. Wkshp. Intell. Robots Syst.*, Osaka, Japan, vol. 3, pp. 1442-1447, November 1991.
- [8] S. Williams, P. Newman, M. Dissanayake, J. Rosenblatt, and H. Durrant-Whyte, "A decoupled, distributed AUV control architecture," *Proc. of the 31st Int. Symp. Robot.*, vol. 1, pp. 246-251, May 2000.
- [9] J. Guivant, E. Nebot, and S. Baiker, "Localization and map building using laser range sensors in outdoor applications," *Journal of Robotic Systems*, vol. 17, pp. 565-583, October 2000.
- [10] J. L. Leonard and H. J. S. Feder, "A computationally efficient method for large-scale concurrent mapping and localization," *Proc. of the 9th Int. Symp. on Robotics Research (ISRR)*, Utah, USA, pp. 316-321, 1999.
- [11] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map building algorithm for real-time implementation," *IEEE Trans. on Robotics and Automation*, vol. 17, pp. 242-257, June 2001.
- [12] S. B. Williams, G. Dissanayake, and H. F. Durrant-Whyte, "An efficient approach to the simultaneous localization and mapping problem," *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 406-411, May 2002.
- [13] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," *Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence*, pp. 176-183, 2000.
- [14] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197-208, 2000.
- [15] K. Murphy, "Bayesian map learning in dynamic environments," *Proc. of Neural Information Processing Systems*, pp. 1015-1021, 1999.
- [16] D. Hähnel, D. Fox, W. Burgard, and S. Thrun, "An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 206-211, October 2003.
- [17] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably con-verges," *Proc. of IJCAI*, pp. 1151-1156, 2003.
- [18] S. Thrun, D. Fox, and W. Burgard, "Monte Carlo localization with mixture proposal distribution," *Proc. of the AAAI National Conference on Artificial Intelligence*, Austin, pp. 859-865, 2000.
- [19] W. Chieh-Chih, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas," *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 842-849, September 2003.
- [20] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," *Proc. of Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153-158, October 2000.
- [21] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering non-linear systems," *Proc. of the American Control Conference*, vol. 3, pp. 1628-1632, June 1995.

- [22] W. Yuanxin, H. Dwen, and W. Meiping, "Unscented Kalman filtering for additive noise case: Augmented versus nonaugmented," *IEEE Signal Processing Letters*, vol. 12, pp. 357-360, May 2005.
- [23] D. G. Lowe, "Object recognition from local scale-invariant features," *Proc. of the 7th IEEE Int. Conf. on Computer Vision*, vol. 2, pp. 1150-1157, September 1999.
- [24] M. Montemerlo, *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*, PhD thesis, Carnegie Mellon University, Pittsburgh, PA 15213, July 2003.



Sukhan Lee received the Ph.D. degree in Electrical Engineering from Purdue University, West Lafayette, IN, in 1982 and the B.S. and M.S. degrees in Electrical Engineering from Seoul National University, Seoul, Korea. He has been awarded several patents and a dozen certificates of recognition and awards from NASA for his technical innovations. Dr. Lee has been an Associate Editor for the *IEEE Transactions on Robotics and Automation*, the *International Journal of AI Tools*, and the *International Journal of Intelligent Systems and Control*. He is currently a Professor of Sungkyunkwan University with the School of Information and Communication and a Chief of the Intelligent Systems Research Center. His research interests are in the areas of robotics and automation, computer integrated manufacturing, neural networks, intelligent systems, sensor fusion and planning, skill learning, and neural networks for robotics.



Seongsoo Lee is a Doctoral student in the School of Information and Communication Engineering at Sungkyunkwan University of Korea. He received the B.S. and M.S. degrees in the Department of Information and Communication Engineering from Soongsil University of Korea in 2003. His research interests include multisensor fusion, vision based SLAM, global localization and robust object recognition based on multiple features.



Dongsung Kim received the Ph.D. degree in Computer Engineering from the University of Southern California, in 1994 and the B.S. and M.S. degrees in Electronic Engineering from Seoul National University, Seoul, Korea. He was a Postdoctoral Researcher at the University of California, Riverside during 1994-1996. He is currently a Professor of Soongsil University with the School of Electronic Engineering. His research interests are in the areas of medical image processing and computer vision.