2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Robust RBPF-SLAM using Sonar Sensors in Non-Static Environments

Jung-Suk Lee, Chanki Kim and Wan Kyun Chung

Abstract— **In this paper, we present a robust RBPF-SLAM algorithm for mobile robots in non-static environments. We propose an approach for sampling particles from multiple ancestor sets, not from just one prior set. This sampling method increases the robustness of SLAM algorithm, because some particles can be updated by only observations consistent with the map, even if observation at certain time step is corrupted by environmental changes. Corrupted observations are filtered out from recursive Bayesian update process by the proposed sampling method. We also present an intermediate path estimation method to use abandoned sensor information reflected from relocated objects for map update. The map can represent the changed configuration of non-static environment by the stored sensor information and the estimated path. Results of simulations and experiments in non-static environments show the robustness of proposed RBPF-SLAM algorithm using sonar sensors.**

I. INTRODUCTION

The simultaneous localization and mapping (SLAM) has been proposed to solve both pose estimation and mapping for mobile robots, concurrently. The SLAM algorithms using extended Kalman filter (EKF) [1], [2] and Rao-Blackwellized particle filter (RBPF) [3], [4] have shown remarkable results in last two decades. However, most of these SLAM algorithms assume that the environment is static.

The real environment is always changed due to the relocation of furniture and electronics while the robot executes SLAM. There are also some objects like people and vehicles moving around the environment. In real non-static environment, these environmental changes cause the inconsistency between the corrupted observations reflected from objects that have been moved and the predicted observations from the map. This inconsistency induces errors in the pose estimation of SLAM process due to the wrong data association.

Observation from changed objects can be registered as a new landmark, not associated with wrong landmarks on the map, if the accurate sensors are used or plentiful measurements are provided. The corrupted observations also can be filtered out, if they can be discriminated from the correct ones. However, it is not easy to know whether the observation is corrupted or not with sonar sensors due to their large uncertainty, low resolution and wide beam directivity, even though they have the advantage of cost competitiveness for commercial service robots. Pose estimation error caused by corrupted observation is almost inevitable for SLAM using sonar sensors, and the errors in pose estimation result in the divergence of SLAM.

There are some researches dealing with non-static environments. SLAM with detection and tracking of moving objects (DATMO) [5], [6], [7] were proposed for safe driving of outdoor vehicles equipped with laser range finders (LRFs) and vision sensors. In these works, only the observations reflected from static objects are selected and used for SLAM process, while the moving obstacles are tracked to avoid collision, independently. In [8] and [9], same problem is solved by proposing single unified framework that includes both data association and target tracking, and builds the map for stationary parts of the environment.

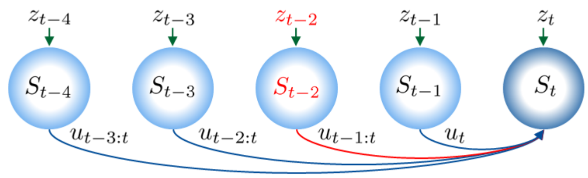
In indoor environment, on the other hand, tracking moving obstacles are not a critical problem, because they do not move so fast. Therefore, researches for indoor SLAM have focused on the mapping for non-static environments. Biswas *et al.* introduced a robot object mapping algorithm that learned the models of non-stationary objects by comparing several maps drawn at different times through EM algorithm [10]. Similarly, observations from dynamic objects were filtered out through EM, while the remaining observations reflected from the stationary parts were used for off-line SLAM in [11]. However, both algorithms cannot update the environmental changes on the map adaptively, even though they can build the map for stationary part of the environment.

For the adaptive mapping algorithm that can be of help to pose estimation of robots, all possible configurations for non-static environment are represented on several local maps [12]. Local maps for same area are updated with multiple timescales and used for long-term SLAM of mobile robots in [13]. Using vision sensors, existence and displacement of landmarks on the map are learned by appearance properties and strength states [14]. In [15], three kinds of maps are updated independently. Union of static and dynamic occupancy grid maps provides a complete description of the environment and the static corner feature map is used to estimate the pose of mobile robot.

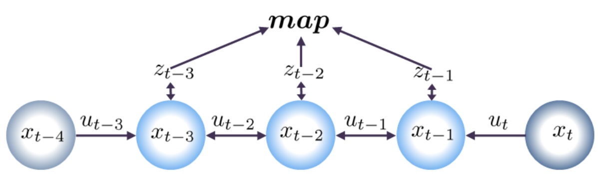
Most of the previous works for non-static environments used LRFs that are accurate enough for robust data association and can offer abundant information containing observations reflected from static parts. However, it is not easy to discriminate corrupted observations with sonar sensors. Moreover, observations are easily corrupted by small environmental changes due to their wide beam directivity. Also, pose estimation in non-static environment was not considered in most of works. Robots cannot estimate its pose robustly, if the environmental changes are not immediately

Jung-Suk Lee and Chanki Kim are with students of Robotics Lab., Dept. of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, KOREA {badol, minekiki}@postech.ac.kr

Wan Kyun Chung is with Faculty of Robotics Lab., Dept. of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, KOREA wkchung@postech.ac.kr



(a) Sampling from multiple ancestor sets.



(b) Estimation of intermediate path and map update.

Fig. 1. Process of the proposed RBPF-SLAM algorithm.

represented on the map, especially when the low performance sensors are used.

In this paper, we propose a robust RBPF-SLAM algorithm that can estimate the pose of the robot robustly and update the map correctly in non-static environment with sonar sensors. This work is started from our previous work for localization of mobile robots in non-static environments [16]. Proposed SLAM algorithm has following characteristics:

- In proposed SLAM algorithm, particles are sampled from multiple ancestor sets, as shown in Fig. 1(a). This sampling makes the particles updated by different combinations of observations, and only correctly weighted particles that are updated by non-corrupted observations can survive.
- When particles are sampled from the several steps earlier set, some sensor information is abandoned because the poses at intermediate time steps are not sampled. To update the map with all available sensor information for changed environment, intermediate path is estimated, as shown in Fig. 1(b).

The remainder of this paper is organized as follows. Section II introduces RBPF-SLAM, the foundation of proposed SLAM framework. In section III, a new sampling method for robust RBPF-SLAM is proposed. Section IV describes the estimation of intermediate path and map update method. Section V shows the experimental results and the conclusion follows in section VI.

II. BACKGROUNDS

In SLAM algorithm, the posterior of the probability distribution for robot's pose and the map is represented as follows:

$$p(x_{1:t}, m | z_{1:t}, u_{1:t}, n_{1:t}), \tag{1}$$

where $x_{1:t}$, $z_{1:t}$, $u_{1:t}$ and $n_{1:t}$ mean the pose of the robot, the sensor observation, the control input and the data association up to time t, respectively. And the m denotes the map. This SLAM posterior can be factored into the map update procedure and the pose estimation by

$$p(m|x_{1:t}, z_{1:t}, u_{1:t}, n_{1:t})\, p(x_{1:t}|z_{1:t}, u_{1:t}, n_{1:t}). \tag{2}$$

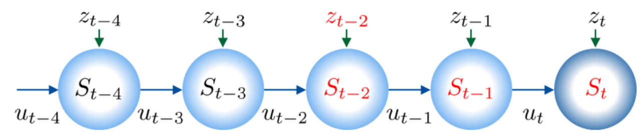


Fig. 2. Recursive Bayesian update process of SLAM: The corrupted observation z_{t-2} disturbs all following estimation results.

In RBPF-SLAM, the pose of the robot, the right term in (2), is described by a set of M weighted particles $S_t = \{x^{t,i}, w^{t,i}\}_{i=1}^{M}$, where $x^{t,i}$ $(= x^i_{1:t})$ and $w^{t,i}$ mean the path and the importance weight for ith particle, respectively. At each time step, particles for S_t is sampled from the prior set S_{t-1} with control input u_t as follows:

$$x^i_{1:t} \sim p(x_t | x^i_{1:t-1}, u_t). \tag{3}$$

Under the assumption that the S_{t-1} is distributed according to $p(x_{1:t-1}|z_{1:t-1}, u_{1:t-1}, n_{1:t-1})$, the proposal distribution for sampled particles is given by

$$p(x_{1:t}|z_{1:t-1}, u_{1:t}, n_{1:t-1}). \tag{4}$$

Then, the importance weight is calculated by

$$\begin{aligned} w^{t,i} &= \frac{\text{target distribution}}{\text{proposal distribution}} \\ &= \frac{p(x^i_{1:t}|z_{1:t}, u_{1:t}, n_{1:t})}{p(x^i_{1:t}|z_{1:t-1}, u_{1:t}, n_{1:t-1})} \\ &\propto \frac{p(z_t|x^i_{1:t}, z_{1:t-1}, u_{1:t}, n_{1:t})\, p(x^i_{1:t}|z_{1:t-1}, u_{1:t}, n_{1:t})}{p(x^i_{1:t}|z_{1:t-1}, u_{1:t}, n_{1:t-1})} \\ &= p(z_t|x^i_t, n_t), \end{aligned} \tag{5}$$

which is proportional to the measurement likelihood.

After the path of the robot is estimated, the map for each particle can be updated from the left term in (2). For the landmark map, the states of landmarks can be updated independently based on the path of the robot as follows:

$$\begin{aligned} &p(m|x_{1:t}, z_{1:t}, u_{1:t}, n_{1:t}) \\ &\quad = \prod_{n=1}^{N} p(m_n|x_{1:t}, z_{1:t}, u_{1:t}, n_{1:t}), \end{aligned} \tag{6}$$

where the map m consists of N landmarks, $m = \{m_1, \ldots, m_N\}$.

The RBPF-SLAM can estimate the path of the robot and update the map efficiently. Per-particle data association ability also can increase the robustness of SLAM [17]. However, it has a particle degeneracy problem where most particles have meaningless importance weight, when the robot's sensor is too accurate [18]. To solve this problem, improved sampling techniques that can substitute (3) for landmark map [19] by

$$x^i_{1:t} \sim p(x_t|x^i_{1:t-1}, u_{1:t}, z_{1:t}, n_{1:t}) \tag{7}$$

and for the grid map [20] were introduced. Sampling particles under consideration of the observation z_t as well as the control input u_t makes the RBPF-SLAM shows better performance.

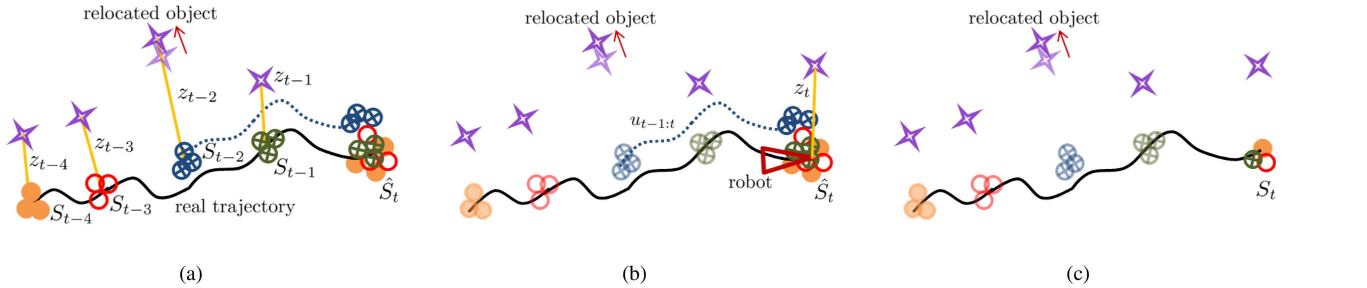


Fig. 3. Process of the sampling from multiple ancestor sets, where $K = 4$, $M = 3$ and $\Delta t_k = \{1,2,3,4\}$. (a) Particles are sampled from K ancestor sets and added to $\hat{S}_t$. (b) $M \times K$ particles are weighted by observation z_t. (c) Highly weighted M particles are added to S_t.

Aforementioned RBPF-SLAM algorithms can correctly estimate the path of the robot and update the map in static environment. However, if the environment is non-static, observations from relocated objects could be associated with the wrong landmarks on the map, which is registered before the environmental changes. Wrong data associations degrade the performance of the pose estimation in SLAM, because their effects remain in recursive Bayesian update process, as shown in Fig. 2.

To solve this problem, we propose a robust RBPF-SLAM framework with a new sampling strategy that can suppress the effects of corrupted observations caused by environmental changes.

III. ROBUST SAMPLING STRATEGY FOR RBPF-SLAM

Once the SLAM posterior is disturbed by corrupted observations, the effect of corruption remains in the SLAM process because the posterior is updated recursively from the posterior of the last time step, as shown in Fig. 2. It is not easy to overcome this problem for RBPF-SLAM using the low performance sensors like sonar. One possible solution for this problem is to make the particles, which are updated by corrupted observations, to be not used in SLAM process. For example, if the particles for S_{t-1} are sampled directly from S_{t-3} in Fig. 2, the effect of corrupted observation z_{t-2} can be removed. However, discrimination of corrupted observations is not an easy problem.

To solve this problem, we propose a new sampling method for RBPF-SLAM, which can select correct particles when the robot later observes stationary parts of the environment. In proposed SLAM algorithm, $M \times K$ particles are sampled from K ancestor sets, not from just one prior set, as follows:

$$x^i_{1:t-\Delta t_k,t} \sim p(x_t | x^i_{1:t-\Delta t_k}, u_{t-\Delta t_k+1:t}), \tag{8}$$

where $k = 1,\ldots,K$, and added to set $\hat{S}_t$.

When particles are sampled from K ancestor sets, as shown in Fig. 3(a), particles except ones sampled from S_{t-2} represent the pose of the robot correctly, not disturbed by corrupted observation z_{t-2}. After $M \times K$ particles are sampled from K sets, all particles are weighted by observation z_t (Fig. 3(b)). At this time, it is highly likely that particles sampled from S_{t-4}, S_{t-3} and S_{t-1} receive high importance weight because they are sampled on correct positions. On the other hand, particles disturbed by corrupted observation z_{t-2} cannot get high importance weight due to errors in estimated pose. And then, through the resampling process, highly weighted M particles are selected from $\hat{S}_t$ and added to set S_t, as shown in Fig. 3(c).

To select the appropriate M particles from $\hat{S}_t$, importance weights are given to particles. If we assume that the $S_{t-\Delta t_k}$ is distributed according to $p(x_{1:t-\Delta t_k} |\, z_{1:t-\Delta t_k}, u_{1:t-\Delta t_k}, n_{1:t-\Delta t_k})$, the proposal distribution of particles sampled from $S_{t-\Delta t_k}$ is given after the estimation of intermediate path $p(x_{t-\Delta t_k+1:t-1} |\, z_{1:t-1}, u_{1:t}, n_{1:t-1}, x_{1:t-\Delta t_k,t})$ that will be explained in Section IV by

$$p(x_{1:t} | z_{1:t-1}, u_{1:t}, n_{1:t-1}), \tag{9}$$

which is the same as the one in (4), and the importance weight is given by the likelihood with the observation z_t in the same manner as in (5). Using the given importance weight, M particles are resampled from $\hat{S}_t$ and added to S_t, which can represent the pose of the robot correctly, not disturbed by corrupted observations.

Though the proposed sampling method can increase the robustness of the RBPF-SLAM algorithm, sampling particles from ancestor sets and updating $M \times K$ particles at every time step need large computational burden. The objective of proposed sampling method is that the corrupted observations are not used to update particles. Therefore, sampling particles from multiple ancestor sets is of no use to increase the robustness of pose estimation, while the robot consecutively observes the same object. To make the proposed SLAM algorithm efficient, particles are sampled from ancestor sets only when the robot observes a new object different to the last observed one. Unfortunately, object detection is not easy with sonar sensors, so the landmark is used as a criterion for sampling timing. When the robot observes a new landmark, which is different than the ones observed by particles in ancestor sets, particles are sampled from K ancestor sets.

Here, observing new landmark can be detected by landmark tracker, which is an independent short time EKF-SLAM algorithm and tracks nearby landmarks to prevent the overlap of ancestor sets corresponding to the same landmark. This simple SLAM process can know whether the current landmark is same with the last observed one or not, and provide the criterion for the sampling method.

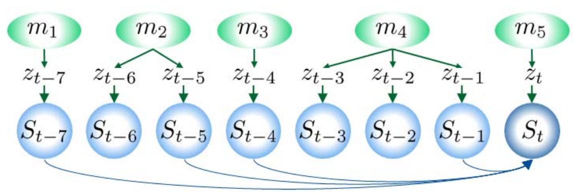


Fig. 4. Particles are sampled from multiple ancestor sets only when the robot observes a new landmark.

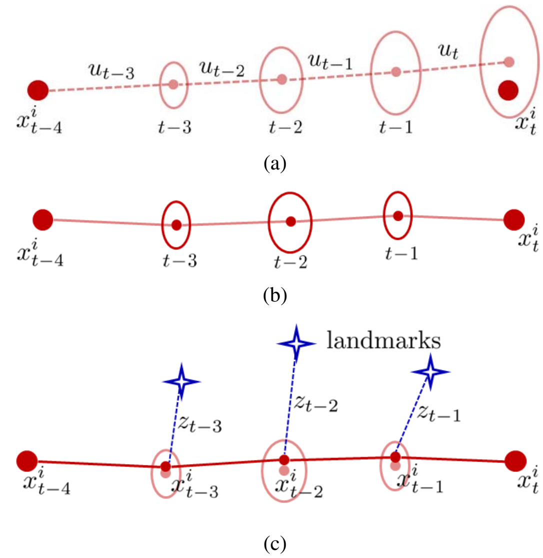


Fig. 5. Process of the intermediate path estimation and the map update. (a) Missing path information between x_{t-s} and x_t (here, $\Delta t_k = 4$). (b) Calculated mean and covariance of intermediate path. (c) Sampling intermediate path and map update.

The final form of proposed RBPF-SLAM algorithm works as shown in Fig. 4. If the robot observes landmark m_5 at time t and the landmark tracker detects that it is different to m_4 observed by the robot till $t-1$, particles are sampled from multiple particle sets that are stored at the last time step of the robot observing each landmark. On the other hand, while the robot observes same landmark, for example m_4, particles in S_{t-1} and S_{t-2} are directly sampled from S_{t-2} and S_{t-3}, respectively, same as in FastSLAM 2.0 [19].

The proposed algorithm can estimate the pose of the robot robustly, even though observations are corrupted by environmental changes. However, some observations cannot be used for map update because the particles sampled from a few time steps earlier cannot have path information at the intermediate time steps. Actually, observations that are reflected from relocated objects and discarded in resampling stage represent the state of changed environment. To use these observations for map update, estimation of path information at intermediate time steps is described in next section.

IV. ESTIMATION OF INTERMEDIATE PATH AND MAP UPDATE

In proposed sampling method, some sensor information between current and ancestor sets is abandoned. For example, if one particle in S_t is sampled from the ancestor set S_{t-4} as in Fig. 1(a), the map cannot be updated by observations $z_{t-3:t-1}$. This is because the path information at intermediate time steps $x_{t-3:t-1}$ is not sampled for that particle. Even if they are discarded in sampling process, the observations reflected from changed environment are useful in map update if the correct path of the robot is given. To update the map that can represent the current environment, first, the path information at intermediate time steps is estimated, and then the map is updated with stored sensor information based on the sampled poses as in Fig. 1(b).

For the ith particle in S_t, which is sampled from S_{t-s}, where $s = \Delta t_k$, the path information $x^i_{t-s+1:t-1}$ at intermediate time steps is sampled based on all available information as follows:

$$x^i_{t-s+1:t-1} \sim p(x_{t-s+1:t-1} | z_{1:t-1}, u_{1:t}, n_{1:t-1}, x^i_{t-s,t}). \quad (10)$$

The path is constrained by the already sampled poses just before and just after the intermediate time steps, x^i_{t-s} and x^i_t, with other data.

Now, the probability distribution for path at intermediate time steps in (10) is rewritten by Bayes' rule and the Markov assumption as follows:

$$\begin{aligned} & p(x_{t-s+1:t-1} | z_{1:t-1}, u_{1:t}, n_{1:t-1}, x^i_{t-s,t}) \\ & = \eta\, p(z_{t-s+1:t-1} | x_{t-s+1:t-1}, u_{t-s+1:t}, n_{t-s+1:t-1}, x^i_{t-s,t}) \\ & \quad \times p(x_{t-s+1:t-1} | u_{t-s+1:t}, x^i_{t-s,t}), \end{aligned} \quad (11)$$

where η is the normalizing constant.

The first term in the right-hand of the equal in (11) can be factorized by

$$\begin{aligned} & p(z_{t-s+1:t-1} | x_{t-s+1:t-1}, u_{t-s+1:t}, n_{t-s+1:t-1}, x^i_{t-s,t}) \\ & = p(z_{t-s+1} | z_{t-s+2:t-1}, x_{t-s+1:t-1}, u_{t-s+1:t}, n_{t-s+1:t-1}, x^i_{t-s,t}) \\ & \times \cdots \times p(z_{t-1} | x_{t-s+1:t-1}, u_{t-s+1:t}, n_{t-s+1:t-1}, x^i_{t-s,t}), \end{aligned} \quad (12)$$

and the typical term is rewritten as follows:

$$\begin{aligned} & p(z_l | z_{l+1:t-1}, x_{t-s+1:t-1}, u_{t-s+1:t}, n_{t-s+1:t-1}, x^i_{t-s,t}) \\ & = \int p(z_l | x_l, n_l, m_{n_l}) \\ & \times p(m_{n_l} | z_{l+1:t-1}, x_{l+1:t-1}, u_{t-s+1:t}, n_{l+1:t-1}, x^i_{t-s,t}) dm_{n_l}, \end{aligned} \quad (13)$$

where $t-s+1 \le l \le t-1$, and m_{n_l} is the observed landmark at time l.

Now the final form of the probability distribution for the intermediate path from (10) can be obtained as follows:

$$\begin{aligned} & p(x_{t-s+1:t-1} | z_{1:t-1}, u_{1:t}, n_{1:t-1}, x^i_{t-s,t}) \\ & \propto p(x_{t-s+1:t-1} | u_{t-s+1:t}, x^i_{t-s,t}) \times \sum_{l=t-s+1}^{t-1} \int p(z_l | x_l, n_l, m_{n_l}) \\ & \times p(m_{n_l} | z_{l+1:t-1}, x_{l+1:t-1}, u_{t-s+1:t}, n_{l+1:t-1}, x^i_{t-s,t}) dm_{n_l}. \end{aligned} \quad (14)$$

The term $p(x_{t-s+1:t-1} | u_{t-s+1:t}, x^i_{t-s,t})$ represents the probability distribution for the intermediate path $x_{t-s+1:t-1}$, which

TABLE I
NUMBER OF PARTICLES (× SETS) USED FOR EACH METHOD

	FastSLAM 2.0	Proposed RBPF-SLAM
For simulations	200	100 (× 5)
For experiment	200	50 (× 5)

is calculated from the control input for intermediate time steps and is also constrained by sampled poses at t and $t-s$. The remaining term inside of the integral can update the probability distribution for x_l with the corresponding observation and the associated landmark on the map.

The implementation for the estimation of intermediate path is as follows. First, to calculate the mean and covariance for each pose $p(x_{t-s+1:t-1}|u_{t-s+1:t}, x^i_{t-s,t})$, the loop-closing algorithm [21] is applied. Based on the control input at intermediated time steps and already sampled poses x^i_{t-s} and x^i_t, as shown in Fig. 5(a), the consecutive relative transformations from $\mathbf{x}^{t-s+1}_{t-s}$ to $\mathbf{x}^t_{t-1}$ and corresponding covariance matrix are optimized by iterated extended Kalman filtering. From the result of this nonlinear constrained least-squares optimization problem, the mean and covariance for each pose at intermediate time step is calculated, as shown in Fig. 5(b).

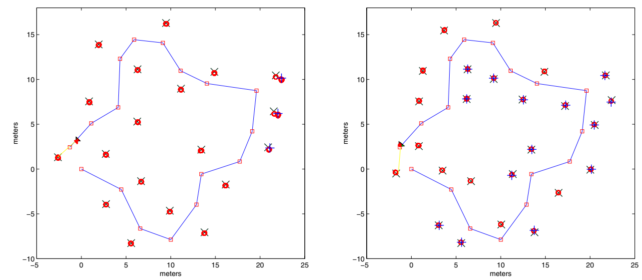
Next, the intermediate path $x^i_{t-s+1:t-1}$ is sampled from the mean and covariance with the stored sensor observations, as shown in Fig. 5(c). Simultaneously, the map is also updated based on the sampled path and sensor observations. This process is same with the sampling stage of FastSLAM 2.0, and applied from the time $t-1$ to $t-s+1$ because the observed landmark m_{n_l} is updated by sensor observation at subsequent time steps $z_{l+1:t-1}$ as in (14). If the observed landmarks at intermediate time steps are already registered on the map, the states of the landmarks are updated. For the observation reflected from the relocated object, corresponding landmark is registered as a new one on the correct position, not associated with wrong landmark by the correctly estimated intermediate path.

V. SIMULATIONS AND EXPERIMENTS

In this section, the performance of proposed RBPF-SLAM algorithm is evaluated by comparing with FastSLAM 2.0 [19] under similar computational burden for each method. The number of particles and sets for each method is shown in Table I. First, the robustness of proposed algorithm was verified with simulations in static and non-static environments. Then, the algorithm was applied to the real robot equipped with sonar sensors in non-static residential-like environment.

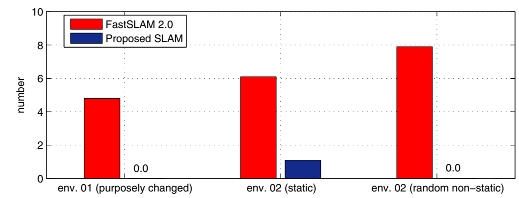
A. Simulations

Performance of each method was verified by simulations in two different environments. In all simulations, the robot traveled the environment 3 times following the given way-points, as shown in Fig. 6, with 3% of motion noise. The robot was equipped with a range-bearing sensor with a maximum range of 20 m, 180° frontal field of view and the standard deviation of 0.2 m in range and 2° in bearing. All simulation results were obtained from 10 independent runs for each algorithm.

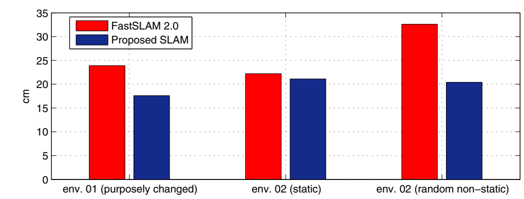


(a) Environment 01 (b) Environment 02

Fig. 6. Simulation environments (30 m × 30 m): Landmarks (red circles) are registered on the map while the robot (red triangle) travels the environment following given way-points (red squares). The environment has initial landmarks (green Xs) and the position of some landmarks are changed (blue plus signs) after the robot's first lap.



(a) Number of spurious landmarks.



(b) Landmark position error.

Fig. 7. Simulation results.

For simulations in the first environment, the rightmost 3 of 19 landmarks were relocated purposely with increasing orders to disturb the data association, after the robot traveled its first lap, as shown in Fig. 6(a). The intentional displacement of the adjacent landmarks makes the observations from them wrongly associated with landmarks that are removed. After the robot observes relocated landmarks, other stationary landmarks are registered as new ones on wrong position, if the pose estimation has errors caused by wrong data association.

In second environment (Fig. 6(b)), performances of both algorithms were also evaluated under random changes of landmarks. 7 $\sim$ 18 of 25 initial landmarks were selected randomly and displaced to new position under the Gaussian random with mean of initial position and variance given by the standard deviation of sensor's range error model. The performance of SLAM in non-static environments was also compared with the one in static environment of which no landmarks were moved.

In Fig. 7, results of simulations are presented. There is small number of spurious landmarks on the map updated by the proposed SLAM algorithm. That means the pose estimation is not disturbed by corrupted observations and the

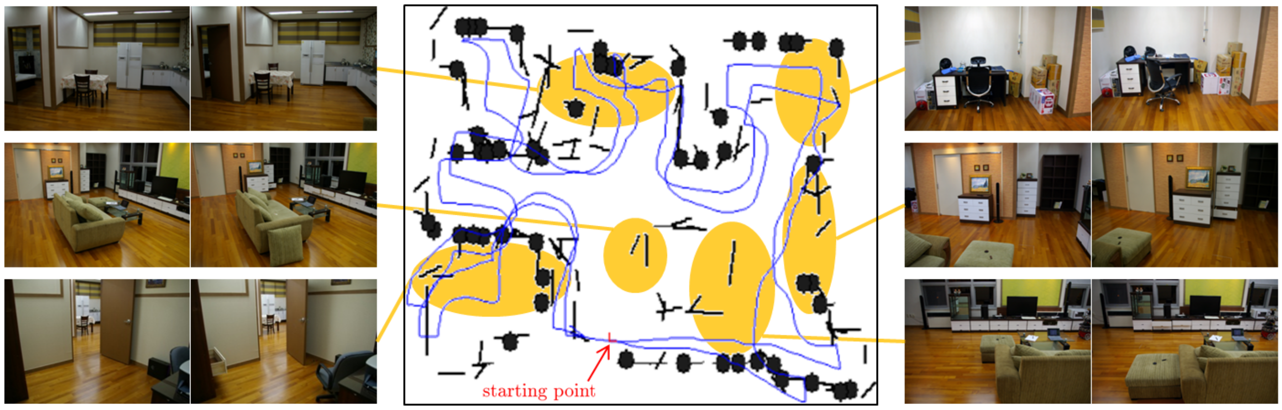


Fig. 8. Residential-like environment (12 m $\times$ 10 m): The landmark map at the center was obtained from proposed SLAM algorithm when the robot finished its first lap. Black circles and lines mean the point and line landmarks, respectively. Blue line denotes the odometry path of the robot. Six pairs of small pictures show the environment at robot's first lap (left in each pair) and the changed environment at second lap (right in each pair).

TABLE II

EXPERIMENTAL RESULTS

Algorithms	Final position error	Final heading error
FastSLAM 2.0	611.3 mm	6.714°
Proposed SLAM	365.3 mm	3.150°

map is correctly updated. Proposed SLAM algorithm failed in only one trial in static environment (Fig. 7(a)). On the contrary, FastSLAM 2.0 registered 5 $\sim$ 8 spurious landmarks even in static environment.

The position error of registered landmarks also shows the performance of proposed SLAM algorithm (Fig. 7(b)). Proposed SLAM registered the landmarks on correct positions in non-static environment as well as in static environment, while the errors of FastSLAM 2.0 in non-static environment were larger than the one in static environment.

B. Experiments in Real Environment

The performance of proposed algorithm was also verified in residential-like environment with Pioneer 3-DX differential-drive robot equipped with 12 sonar sensors. Line and point landmarks were extracted from sonar sensors by [22]. The standard deviations in range (bearing) for two kinds of landmarks are 0.2 m (5°) for line and 0.15 m (4°) for point. The motion noise is 3% of odometry information. The robot traveled the environment two times with wall following algorithm and arrived at the starting point. The average speed of the robot was about 0.1 m/s. After the robot traveled its first lap, some objects were displaced to new positions, as shown in Fig. 8. All following results were obtained from 10 independent runs for each method with stored sensor and odometry data that were sampled with 4 Hz.

The final position and heading errors from each SLAM method are shown in Table. II. Final pose error from proposed SLAM algorithm is less than the one from FastSLAM 2.0, which means the proposed algorithm can estimate the pose of the robot robustly under environmental changes. The map also shows the performance of the proposed algorithm. The map from the proposed SLAM represents the environment correctly, as shown in Fig. 9. Line landmarks on the map from the proposed algorithm are well aligned. On the contrary, the entire configuration of the map from FastSLAM 2.0 is distorted compare with the line landmarks at the center bottom of the map, which were observed right after the robot started its travel. The proposed algorithm also successfully registered the landmarks for relocated objects on the final map as compared with the map (note the areas emphasized with ellipses in Fig. 8 compared with the same areas in Fig. 9(a)) obtained when the robot finished its first lap.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we presented a robust RBPF-SLAM algorithm that can estimate the pose of the robot and update the map robustly with sonar sensors in non-static environments. The advantages of this algorithm are as follows:

- Sampling from multiple ancestor sets makes some particles not updated by corrupted observations from the environmental changes. Correctly updated particles that survive through sampling process can estimate the pose of the robot.
- Survived particles can update the map, which represent the changed environment correctly, by the estimation of intermediate path.

Our algorithm has been implemented and evaluated via simulations and experiments in non-static environments. The results showed that the proposed SLAM algorithm is robust and accurate with low performance sensors compared with the FastSLAM 2.0 under similar computational burden.

Future work will include the map management method that updates the reliability of landmark using negative information for the elimination of spurious landmarks due to the environmental changes or the errors from the landmark extraction process.

VII. ACKNOWLEDGMENTS

This work was supported in part by the Acceleration Research Program of the Ministry of Education, Science and

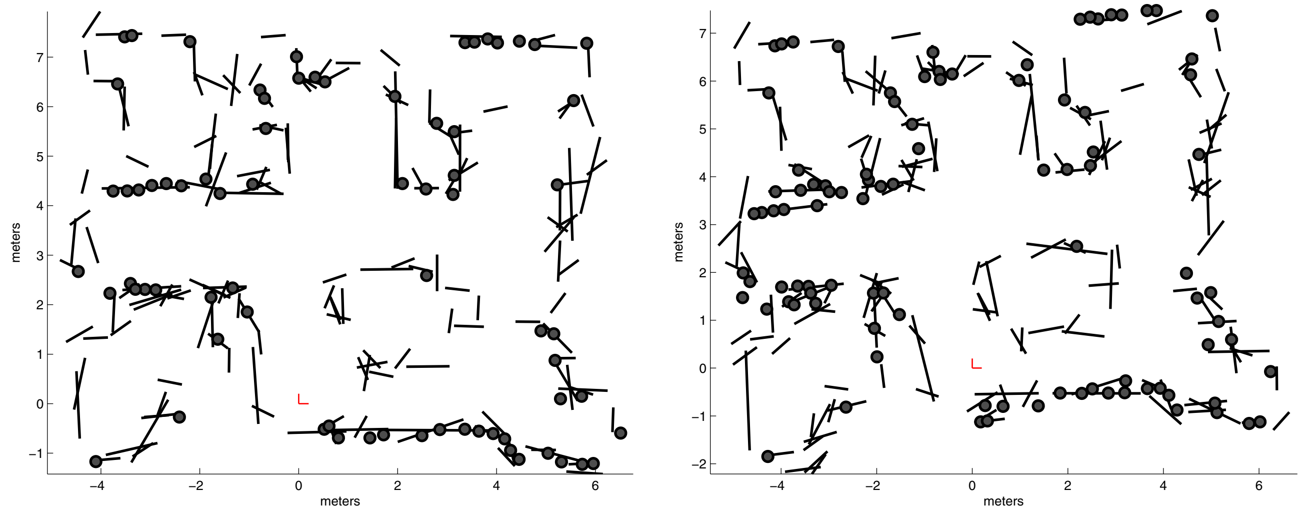


Fig. 9. Landmark map obtained after the robot finished its second lap for each SLAM algorithm: Black circles and lines mean the point and line landmarks.

Technology of the Republic of Korea and the Korea Science and Engineering Foundation [R17-2008-021-01000-0], in part by the Korea Health 21 R&D Project, Ministry of Health & Welfare, Republic of Korea under Grant A020603, in part by the Agency for Defence Development and by Unmanned Technology Research Center(UTRC), Korea Advanced Institute of Science and Technology, and in part by the IT R&D program of MKE/IITA [2008-F-038-1, Development of Context Adaptive Cognition Technology].

References

[1] R.C. Smith and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty," *The Int. Journal of Robotics Research*, vol.5, no. 4, 1986, pp.56-68.

[2] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte and M. Csorba, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," *IEEE Tran. on Robotics and Automation*, vol.17, no. 3, 2001, pp.229-241.

[3] K. Murphy, "Bayesian Map Learning in Dynamic Environments," *in Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 1999, pp.1015-1021.

[4] M. Montemerlo, S. Thrun, K. Koller and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneously Localization and Mapping Problem," *in Proc. of the National Conf. on Artificial Intelligence (AAAI)*, 2002, pp.593-598.

[5] CC. Wang, 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," *in Proc. of IEEE Int. Conf. on Robotics and Automation*, vol.1, 2003, pp.841-849.

[6] H. Zhao, M. Chiba, R. Shibasaki, X. Shao, J. Cui and H. Zha, "SLAM in a Dynamic Large Outdoor Environment using a Laser Scanner," *in Proc. of IEEE Int. Conf. on Robotics and Automation*, 2008, pp.1455-1462.

[7] T. Vu, O. Aycard and N. Appenrodt, "Online Localization and Mapping with Moving Object Tracking in Dynamic Outdoor Environments," *in IEEE Intelligent Vehicles Symposium*, 2007, pp.190-195.

[8] D. Hähnel, D. Schulz and W. Burgard, "Map Building with Mobile Robots in Populated Environment," *in Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol.1, 2002, pp.496-501.

[9] I. Miller and M. Campbell, "Rao-Blackwellized Particle Filtering for Mapping Dynamic Environments," *in Proc. of IEEE Int. Conf. on Robotics and Automation*, 2007, pp.3862-3869.

[10] R. Biswas, B. Limketkai, S. Sanner and S. Thrun, "Towards Object Mapping in Non-Stationary Environments with Mobile Robots," *in Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol.1, 2002, pp.1014-1019.

[11] D. Hähnel, R. Triebel, W. Burgard and S. Thrun, "Map Building with Mobile Robots in Dynamic Environment," *in Proc. of IEEE Int. Conf. on Robotics and Automation*, vol.2, 2003, pp.1557-1563.

[12] C. Stachniss and W. Burgard, "Mobile Robot Mapping and Localization in Non-Static Environment," *in American Association for Artificial Intelligence*, 2005, pp.1324-1329.

[13] P. Biber and T. Duckett, "Experimental Analysis of Sample-Based Maps for Long-Term SLAM," *The Int. Journal of Robotics Research*, vol.28, no. 1, 2009, pp.20-33.

[14] J. Andrade-Cetto and A. Sanfeliu, "Concurrent Map Building and Localization on Indoor Dynamic Environments," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol.16, no. 3, 2002, pp.361-374.

[15] D.F. Wolf and G.S. Sukhatme, "Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments," *Autonomous Robots*, vol.19, no. 1, 2005, pp.53-65.

[16] J. Lee and W. K. Chung, "Robust Particle Filter Localization by Sampling from Non-corrupted Window with Incomplete Map," *in Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp.1133-1139.

[17] M. Montemerlo and S. Thrun, "Simultaneous Localization and Mapping with Unknown Data Association using FastSLAM," *in Proc. of IEEE Int. Conf. on Robotics and Automation*, vol.2, 2003, pp.1985-1991.

[18] A. Doucet, S. Godsill and C. Andrieu, "On sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing*, vol.10, 2000, pp.197-208.

[19] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, "FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges," *in Proc. of the Sixteenth Int. Joint Conf. on Artificial Intelligence*, 2003, pp.1151-1156.

[20] G. Grisetti, C. Stachniss and W. Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," *IEEE Tran. on Robotics*, vol.23, no. 1, 2007, pp.34-46.

[21] C. Estrada, J. Neira and J.D. Tardós, "Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments," *IEEE Tran. on Robotics*, vol.21, no. 4, 2005, pp.588-596.

[22] S. Lee, J. Lim and D. Cho, "Feature-based Map Building using Sparse Sonar Data in a Home-like Environment," *Journal of Mechanical Science and Technology*, vol.21, no. 1, 2007, pp.74-82.