Ph. D DISSERTATION

Cooperative Rao-Blackwellized Particle Filter based SLAM framework using Geometric Information and Inter-Robot Measurements

기하학적 정보와 로봇 간 측정값을 이용한 협조적 Rao-Blackwellized Particle Filter 기반 SLAM 프레임워크

BY

SEUNG-HWAN LEE

AUGUST 2015

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Abstracts

In unknown environments, multiple robots must have capabilities to sense and interpret their surroundings, and localize themselves before performing some missions such as exploring the mineral resources and rescuing people. It is usually called multi-robot simultaneous localization and mapping. To perform multi-robot SLAM more accurately, robots are required to build maps of their surroundings accurately. In addition, inter-robot measurements should be properly utilized in the SLAM process.

In this dissertation, a novel Rao-Blackwellized particle filter based SLAM framework is presented using geometric information and inter-robot measurements for accurate multi-robot SLAM. For SLAM, a Rao-Blackwellized particle filter (RBPF) is basically one of representative methods. It takes advantage of linear time-complexity which is linearly proportional to the number of features by factoring the full SLAM posterior into the product of a robot path posterior and landmark posteriors. Additionally, it deals with multi-hypothesis data association using particles with their own data association. They makes it more robust than extended Kalman filter based SLAM.

The proposed SLAM framework is divided into two major parts. First, the RBPF is improved using cooperation among particles in case of single robot SLAM, which is called Relational RBPF-SLAM. Here, the framework

basically follows the process of the factored solution to SLAM using the unscented Kalman filter (UFastSLAM), which is an accurate instance of RBPF-SLAM. A concept of particle to particle cooperation is considered in the importance weight step and the resampling step to increase the SLAM accuracy and solve some inherent problems such as the particle depletion problem and the data association problem. The particle depletion problem is almost eliminated using the formation maintenance of particles which is controlled without any rejection or replication of particles during the resampling step. In addition, to overcome the data association problem, the posterior distribution is estimated more accurately by compensating improperly assigned weights of particles. Secondly, to reduce the accumulated robot pose errors and feature errors, inter-robot measurements are utilized in the proposed RBPF-SLAM framework. They can be measured when a rendezvous between robots occurs or robots share common features. To deal with the inter-robot measurements, a Kalman consensus filter scheme is involved in the proposed RBPF-SLAM framework, which is robust than the covariance intersection method. Several simulations and experiments show significant improvements of the proposed RBPF-SLAM framework in both the accuracy of robot poses and map quality by comparing the state of the art techniques, i.e. FastSLAM 2.0, particle swarm optimization (PSO) based FastSLAM, UFastSLAM, particle fission based UFastSLAM and PSO based UFastSLAM.

Keywords: Rao-Blackwellized Particle Filter, UFastSLAM, Triangular Mesh Generation, Expectation Maximization, Kalman Consensus Filter.

Student Number: 2010-30995

Contents

List of Figures

List of Tables

Nomenclature

Notation	Description	Chapter
$X_t^{[i]}$	The *i*th particle	2
$x_t^{[i]}$	The *i*th particle's pose	2,3,4
$\mu_{N,t}^{[i]}$	The Feature mean of the *i*th particle	2
$\sum_{N,t}^{[i]}$	The Feature covariance of the *i*th particle	2
N_f	The number of features	2
N	The number of particles	2
$c_{1:t}$	The set of data associations until time *t*	2
$u_{1:t}$	The set of control input until time *t*	2
f	The motion model	2
h	The observation model	2
z_t	The measurement at *t*	2
R_t	The measurement noise covariance	2,5,6
Q_t	The control input noise covariance	2,5,6
$w_t^{[m]}$	The weight of *m*th particle at *t*	2,3
N_{eff}	The effective number of particles as an operation criteria for the resampling step	2

R_i	The *i*th robot	2,4,5
u_i	The information vector by the *i*th robot	2,4
U_i	The information matrix by the *i*th robot	2,4
$\hat{z}_{j,k}^{t,i}$	The prior estimate of the robot *t* with particle *i* to the *j*th landmark at time *k*	2
$\mu_{j,k-1}^{t,i}$	The updated *j*th landmark of the robot *t* with particle *i* until *k*-1	2
$wc_t^{[m]}$	The compensation term in the weight equation	3
$T_{[m],i}^{p}$	The *i*th expectation of a new likelihood for the *m*th particle after the *p*-th iteration	3
θ_{un}	A vector of unknown parameters	3
μ_1^L	The means of normal distribution for the first cluster after *L*-iteration	3
μ_2^L	The means of normal distribution for the second cluster after *L*-iteration	3
T_m	The triangular configuration of the *m*th particle	3
W_{T_m}	The weight of T_m	3
$x_t^{[n_1]}$	The neighbor particle who has the shortest Euclidean distance from particle $x_t^{[m]}$	3
$x_t^{[n_2]}$	The neighbor particle that is determined to minimize the sum of two distances, i.e., the distance between $x_t^{[m]}$ and $x_t^{[n_1]}$ and the distance between $x_t^{[n_1]}$ and $x_t^{[n_2]}$.	3
$Pw_t^{[k]}$	the transformed weight for the *k*th particle to operate the PFM	3
Δ_a^n	The relative transformation vector	4

R_f	The reference robot	4,5
H_i	The Jacobian of the observation model in Consensus	4
$b_{a'}^{f}$	The fused information vector for R_f	4
$B_{a'}^{f}$	The fused information matrix for R_f	4
$\hat{P}_{a'\|a'-1}^{fn,(i)}$	The merged covariance for R_f and R_n	4
$\hat{x}_{a'\|a'-1}^{fn,(i)}$	The merged state for R_f and R_n	4
$\left(wc_t^{[m]}\right)'$	The simple weight compensation term	5
η	A parameter of the simple compensation scheme	5
$\varepsilon_{Pose}(t)$	The robot pose error according to t	5
$\varepsilon_{Feature}(t)$	The feature error according to t	5
ε_{Pose}	The total errors of the robot poses	5
$\varepsilon_{Features}$	The total errors of the feature locations	5
$\varepsilon_{Avg}(t)$	The average feature errors according to t	5
$\mathbf{I}$	The line feature vector	6

Chapter 1

Introduction

1.1 Background and motivations

Localizing a robot while building a map of its surroundings, called Simultaneous Localization and Mapping (SLAM), is one of the most important capabilities that intelligent robots should have. To perform SLAM, there are three approaches, i.e. scan matching based SLAM [1-9], graph SLAM [10-22], and Bayesian filter based SLAM [23-40]. These approaches have pros and cons according to their purpose, the type of features, the type of data acquisition methods and the SLAM implementation. In case of scan matching based SLAM, it can be operated without the robot control inputs such as odometry information. To obtain all estimated robot poses and feature locations directly, called full SLAM, the graph approach is suitable for SLAM [96]. To estimate the current state and the current covariance based on the previous state and

measurements, called online SLAM, the Bayesian filter based SLAM can be operated well and we handle this SLAM framework in detail in this dissertation.

EKF-SLAM [23-25], as the oldest and the most popular approach, has served as the main approach to the SLAM problem for the last twenty years. However, EKF-SLAM is known to have two major well-known shortcomings: quadratic computational complexity and sensitivity to failures in data association. First, the computation of EKF-SLAM grows quadratically with the number of landmarks since observation-update step requires that all landmarks and the joint covariance matrix are updated every observation [33]. Second, it is especially difficult to deal with a loop closure problem, that is, when a robot returns to observe landmarks again after a large traverse. These shortcomings consequently make it difficult to apply EKF-SLAM in real and large environments.

Murphy [27] and Doucet [28] introduced Rao-Blackwellized particle filters as an effective means to solve the SLAM problem. Each particle in a RBPF represents a potential trajectory of the robot and a map of its surrounding environment. FastSLAM, which is a specific instance of SLAM based on a Rao-Blackwellized particle filter (RBPF-SLAM), has been considered as an alternative solution to overcome the above mentioned problems [35, 36]. It takes advantage of linear time-complexity which is linearly proportional to the number of features by factoring the full SLAM posterior into the product of a robot path posterior and landmark posteriors. Additionally, it induces a multi-

hypothesis data association using multiple particles with its own data association, which makes it more robust than EKF-SLAM. However, there are some limitations in spite of the advantages of RBPF-SLAM when a finite number of particles is used. One issue is that the distinctive particles are decreased over time because low-weight particles are rejected and high-weight particles are replicated in the resampling step known as the particle depletion problem [41, 42]. In this way, RBPF-SLAM typically does not maintain multiple data association hypotheses. Another limitation is related to filter convergence. The weight of the particles can be assigned according to an improper proposal distribution, which can cause a failure in the approximation of the posterior distribution. Finally, the filter can be diverged. In addition, the resampling step can accelerate the degeneration of the convergence capability through rejection and replication using the particle weights. This dissertation deals with both the filter convergence and the particle depletion problems. In addition, another problem related to data association is suggested and concerned throughout this dissertation.

When RBPF-SLAM is applied to a multi-robot system (MRS), the filter can consider the inter-robot measurements because these inter-robot measurements allow the robot state and the features to be accurately compensated. These measurements occur when two robots meet each other (*Rendezvous event*) or multiple robots share a common place or feature (*Feature sharing event*) as shown in Fig.1. When a feature sharing event occurs, their common features

(a) Rendevous Event (b) Feature Sharing Event

Figure 1.1 Two cases for acquiring inter-robot measurements. Two robots meet each other in (a), which is called *Rendezvous event*. Multiple robots share a common feature or place, which denotes *Feature sharing event*

can be updated according to the scheme of a Kalman consensus information filtering (KCIF) [92]. The KCIF has more accurate and better convergence capability performance than covariance intersection (CI) for data fusion [93]. In case of a rendezvous event, which occurs more frequently than the feature sharing event in multi-robot systems, multi-robot poses can be estimated more accurately using inter-robot measurements between two robots at the rendezvous point. In this dissertation, the above situations are considered by combining the KCIF scheme and RBPF-SLAM framework.

1.2 Related Work

There are three categories for SLAM, i.e. scan matching based SLAM, graph SLAM and Bayesian filter based SLAM, which are described in Fig. 1.2. These are considered in Section 1.2.1, 1.2.2, and 1.2.3 in detail, respectively. In

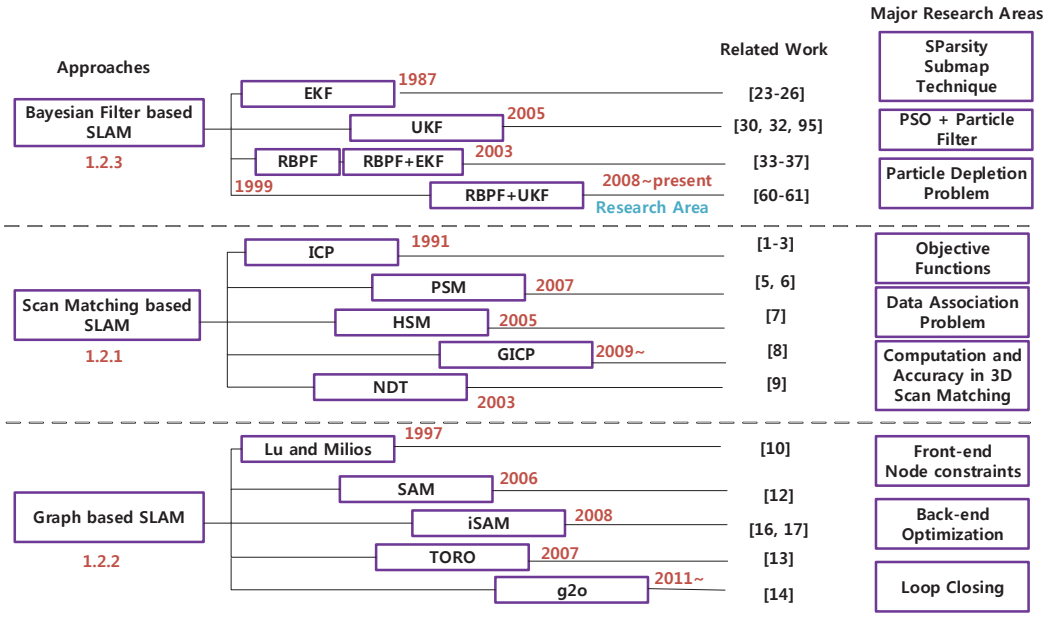


Figure 1.2 Research Areas for SLAM.

addition, unknown initialization problem and data fusion for multi-robot SLAM are handled in Section 1.2.4.

1.2.1 Scan Matching based SLAM

For SLAM, scan matching methods estimate the pose of robots by using two consecutive scans, which are measurements obtained from range sensors. Scan matching methods can be divided into point-to-point matching, point-to-line matching, point-to-plane matching and plane to plane matching. One of the most popular and the oldest scan matching methods is the iterative closest point (ICP) method [1-3]. The ICP method is a point-to-point method that find out the optimal transformation between two consecutive scans. The sum of the squared Euclidean distances between the corresponding points is minimized

iteratively according to its cost function. If the angle difference between two scans is small, the method is well operated. Otherwise, many points are not associated correctly, which results in incorrect data association and incorrect transformation. For this reason, iterative dual correspondence (IDC) alternatively and separately operates rotation and translation by minimizing the sum of the squared Euclidean distances [4]. Generalized-ICP (G-ICP) represents the cost function of the standard ICP as a single probabilistic framework [8]. It can represent the point-to-point and the point-to-plane ICP according to the projection matrix in the cost function.

The property of the Hough domain is employed in Hough scan matching (HSM) [7]. HSM extracts spectrum which is the distribution of features from the geometric points. After the extraction step, the spectrums obtained at two consecutive time steps are matched according to their correlation. HSM computes a transformation without the iteration, which finds out the maximal correlation between the spectra of a current scan and a reference scan. Diosi and Kleeman [5] developed polar scan matching (PSM) which belongs to the family of point-to-point scan matching techniques. PSM not only takes the advantage of the structure of the sensor measurements but also eliminates an expensive search for corresponding points differently from the standard ICP. In addition, PSM is generally aided by a robot odometer for the sake of obtaining a possible angle. In [6], curvature functions are considered for dynamic environments in PSM. In scan data, the amount of dynamic objects is removed

and compensated by the interpolation methods such as the linear interpolation, the nearest-neighbor interpolation, the polynomial interpolation.

Another method for scan matching is the normal distributions transform (NDT), which was initially proposed as a 2-D scan matching method and was later extended to three dimensions [9]. The 3-D NDT is not a point-to-point approach which conducts scan matching between two scans using the closest points but is a point-to-distribution approach that carries out scan matching between the reference scan and a set of distributions in grids or voxels generated from the model scan. Because the NDT does not need to search for the closest points or store the raw data from the model scan, it can greatly reduce computational burden and the amount of required memory, especially in 3-D scan matching. In addition, the gradient vector and the Hessian matrix of its score function have analytic forms, allowing the simple use of standard nonlinear optimization methods such as Gauss-Newton and Levenberg-Marquardt optimization to estimate the optimal transformation.

1.2.2 Graph SLAM

The graph-based SLAM is a full SLAM or off-line SLAM, which constructs a graph whose nodes represent robot poses or landmarks and in which an edge between two nodes encodes a sensor measurement that constrains the connected poses. The constraints denote spatial information, such as a rigid transformation and a Euclidean distance with a bearing angle.

The graph-based SLAM was first formulated as a nonlinear optimization problem by Lu and Milios [10]. But it took many years to make this formulation popular due to the comparably high complexity of solving the error minimization problem using standard techniques. Subsequently, many algorithms, such as multi-level relaxation (MLR) [11], square root smoothing and mapping (SAM) [12], tree-based network optimizer (TORO) [13], general graph optimization (g2o) [14], and so on [17, 18, 19], have been recently proposed. Furthermore, some of them were extended to incremental versions [16, 17]. These approaches can be divided by full linear solvers using several decomposition methods and stochastic relaxation using Gauss-Seidel, Stochastic gradient descent, Gauss-Newton, Levenberg-Marquardt and so on. A heuristic approach is Explicit Loop Closing Heuristics (ELCH) that is a fast and very accurate solution [18]. It also can be easily combined with other optimization tools such as LUM, TORO. However, ELCH distributes the transformation vector over the SLAM graph locally, which only depends on the last node. As shown in Fig. 1.3, it is extended in [19] for multiple robots, which is extended ELCH. When a rendezvous between robots occurs or they share a common place, a loop is constructed and compensated according to the extended ELCH scheme.

Most graph-based SLAM algorithms focus on back-end which optimizes the graph constructed in front-end composed of loop detection [20], scan matching [1], and so on. However, the back-end relies heavily on the

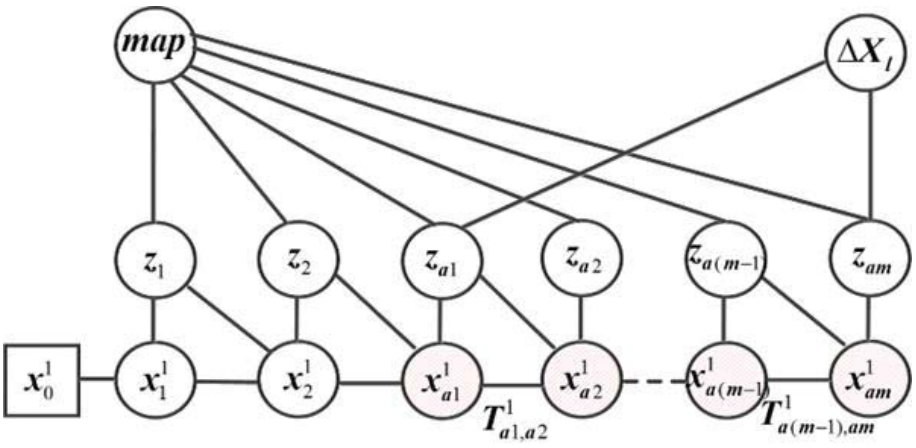


Figure 1.3 Example of graphical model for extended ELCH (EELCH)

information of the constructed graph, thus front-ends are critical for the performance of the graph-based SLAM. Hence, to overcome errors of graph from front-ends, robust graph-based SLAM approaches were proposed [21, 22]. In spite of that, because the robust algorithms still utilize the information of the edges judged as inliers, the significance of front-ends has not changed in order to guarantee the performance of the graph-based SLAM.

1.2.3 Bayesian Filter based SLAM for Single Robot

Unlike scan matching based SLAM and graph SLAM, as online SLAM, Bayesian filter based SLAM can recursively estimate the current robot pose and a map of its surroundings in unknown data association at each time step. One of popular and simple approaches is the extended Kalman filter based SLAM (EKF-SLAM), which assumes a motion and a measurement noise are distributed according to the Gaussian distribution, and motion models of robots

and measurement models of sensors are nonlinear [23, 24]. Until recently, it has been used due to its simple implementation for SLAM, but this approach has some limitations such as computational complexity and single hypothesis data association. As the above mentioned, the first drawback of the EKF as a solution to the SLAM problem is related to computational complexity [28, 57]. Both the computation time and memory required by the EKF scale quadratically with the number of landmarks in the map. SLAM algorithms based on the full EKF generally do not scale beyond a few hundred landmarks. However, millions of features are contained in reasonably large environment models. The second one is single-hypothesis data association. If a large number of readings are incorporated incorrectly into the EKF, the filter will diverge. To overcome the above two problems, the sparse extended information filter SLAM (SEIF-SLAM) [25, 26] was suggested by solving the scalable problem by sparsification of an information matrix. They assumed that most pairs of landmarks are nearly conditional independent of each other. Even though degrading its accuracy from the assumption, the SEIF-SLAM achieves that the time complexity per step and the space complexity are $O(1)$ and $O(N)$, where N is the number of landmarks. The unscented Kalman filter (UKF) is a more reliable estimator than EKF while the system model is highly nonlinear [30]. The past of the UKF is relatively short compared to EKF. By approximating the probability density function instead of the nonlinear function itself, UKF SLAM [30, 32, 95] received a considerable attention. To reduce the

computational complexity, UKF SLAM adopts the square root of the covariance matrix [26]. In the simulations, it showed 20 percent decrease in computation time. However, it is required to improve the computational complexity for online SLAM.

In a work by Murphy [33] and Doucet [34], Rao-Blackwellized particle filters (RBPF) have been introduced as an effective means to solve the SLAM problem. It consists of a particle filter for robot pose estimation and an optimal filter for feature estimation. Each particle in a RBPF represents a possible robot trajectory and a map. An instance of RBPF-SLAM is FastSLAM [35, 36] where the particle filter is used for the mobile robot position estimation, and EKF is used for the feature update. The first version of FastSLAM is FastSLAM 1.0 where the proposal distribution only relies on the motion estimate, and the second version of FastSLAM is FastSLAM 2.0 which considers the motion estimate and the most recent sensor measurement in the proposal distribution. Unlike EKF or UKF based approaches, RBPF-SLAM has two advantages that are linear time complexity and multi-hypothesis data association [35-37]. The time complexity is linearly proportional to the number of features. Multiple particles can estimate the true target well even though some particles fail to the estimation. In [54, 55], the motion model using cameras was applied to RBPF-SLAM. They used visual odometry and vision-based sensing. Recently, keyframe based RBPF-SLAM has been applied to camera tracking and 3D mapping [56]. The keyframe is redefined if the percentage of features tracked

by the KLT tracker is smaller than a pre-defined value. The camera poses and features at the keyframe are only estimated except ones at the non-keyframe. However, RBPF-SLAM also suffers from some drawbacks, namely, the particle depletion problem and the problems caused by the derivation of the Jacobian matrices and the linear approximations of the nonlinear functions.

The particle depletion problem is that the number of distinctive particles is decreased over time which causes loss of multi-hypothesis data association property. It occurs when a finite number of particles is used in RBPF-SLAM and most of cases are involved. To overcome the problem, G. Grisetti et al. [37, 53] proposed accurate proposals using scan matching and selective resampling technique using N_{eff}, which is defined by the variance of the particle weights. If N_{eff} drops below a given threshold $N/2$, particles are resampled according to the resampling scheme. In the importance weight step, the probability of the motion model is defined as a constant, which means the reduction of effect for the motion model. Scan matching technique is also involved for minimizing odometric errors during mapping in [59]. A probabilistic model of the residual errors of scan matching process is then used for resampling steps. In [38, 39], the concept of active loop closing is used for accurate SLAM using hierarchical map representation that consists of a topological map and an occupancy grid map. If the distances between the current area and the visited area in both the occupancy grid map and the occupancy grid map are quite different, a loop can be closed. By re-entering already visited areas, the robot can reduces its

localization error and this way learns more accurate maps. For the particle depletion problem [42], several researchers considered geometric information of particles in RBPF-SLAM. The two categories are as follows: particle swap [40-42] and particle swarm optimization (PSO) [43-46]. The particles with relatively low weights and the particles with relatively high weights are highly rejected and replicated during the resampling step, respectively when a finite number of particles is used. It causes the particle depletion problem. To alleviate that problem, some particles are swapped toward area of high-weight particles. In terms of particle fission, particles whose importance weight is bigger than the mean of weights are selected and, then split up several particles. These generated particles obey normal state distribution with the mean of the original particle. The results show that the number of distinctive particles are not largely decreased overtime. The other approach, i.e. PSO, uses an idea that particles cooperate with one another to track a common target. PSO is a population-based technique, similar in some respects to evolutionary algorithms, except that potential solutions (particles) move and cooperate with one another, rather than evolve through the search space. The particle dynamics which govern the movement are inspired by models of swarming and flocking [90]. In some approaches, PSO is combined with FastSLAM in the sampling step and in the resampling step. PSO was carried out to congregate particles toward the target point and helped keep them from diverging. The H_∞ filter is considered with PSO in FastSLAM [45]. The H_∞ filter is used instead of EKF

for overcoming the inaccuracy caused by the linear approximations of nonlinear functions and tends to be more robust in the system taking additional uncertainties. In several simulations, it operated more robust than FastSLAM 2.0. To acquire the posterior distribution more precisely, particle weights are re-approximated in Monte Carlo frameworks [47, 48]. Doing this considerably reduces the number of errors occurred in the computation of proposal distribution during the sampling step. In addition, this method improves the posterior distribution by taking the continuity and smoothness of the distribution into account. It worked reasonably well, but the particle with the maximum or minimum weight was treated as a meaningless particle, which implies that the best or the worst information for a true robot pose was neglected in the re-approximation.

The problems caused by the derivation of the Jacobian matrices and the linear approximations of the nonlinear functions have been studied in [31, 60, 61, 64]. To solve these problems, Unscented FastSLAM (UFastSLAM) has been suggested in [60-61]. UFastSLAM overcomes the drawbacks caused by Jacobian matrices and the linear approximations in the FastSLAM framework. In UFastSLAM, the linearization process with Jacobian calculations is removed by applying the unscented transformation (UT) to the SLAM framework. In [60], a full version of the UFastSLAM algorithm was presented. In this work, unscented Kalman filter (UKF) is used to update the mean and covariance of the feature and to initialize new features. Also, the unscented particle filter

technique was utilized in the prediction step of the vehicle state, and the unscented particle filter provides a better proposal distribution without the accumulation of linearization errors and without the need to calculate the Jacobian matrices in the measurement updates. These approaches make the filter converge close to a real robot pose and decrease the number of rejected and replicated particles in the resampling step. For the improvement of SLAM accuracy, UKF was iteratively used in the robot pose estimation and the feature estimation, which called iterated unscented Kalman filter (IUKF) [66]. In the simulations, it showed the enhanced results in terms of SLAM accuracy and consistency. But its time cost is higher than FastSLAM 2.0 and UFastSLAM. This is because calculating Jacobians in FastSLAM 2.0 is a more computational efficient linearization method than the SUT in the simulator. L. Liu et al. [69] proposed a vision-based semantic UFastSLAM framework. The concept of semantic topological metric map that employs the semantic relationships between the landmarks was considered in the robot pose estimation and the feature estimation steps. PSO is also adapted to optimize particles in [62, 65]. In [65], PSO caused the particle set to tend to the high probability region of the posterior before the weights are updated. In addition, computational loads are reduced using Cholesky factorization for the covariance matrix. In [62], PSO was involved in the UFastSLAM framework, which was performed after the resampling step. FastSLAM 2.0, UFastSLAM and PSO based UFastSLAM were compared in several simulations and real experiments using an

autonomous underwater vehicle.

In our previous works [49-52], the weights of particles are compensated by heuristic rules. In addition, the particles constitute formation by means of the generation of an adaptive triangular mesh [70, 71] in the resampling step. In [52], in particular, these two steps for the weight compensation and the particle formation maintenance are involved in the FastSLAM framework. In some tests, the proposed approach showed better SLAM performance, i.e. robot pose errors and feature errors, than FastSLAM 2.0, PSO based FastSLAM and UFastSLAM. However, since the basic framework is FastSLAM 2.0, the proposal distribution and the observation model are linearly approximated, which requires more accurate system modeling.

1.2.4 Bayesian Filter based SLAM for Multiple Robots

Rao-Blackwellized particle filters for a single robot can be extended for multiple robots, which implies multi-robot SLAM [72, 73]. Multi-robot SLAM has some advantages relative to single robot SLAM: (a) a massive amount of the measurements acquired from several robots; (b) saving computing power; (c) a variety of external information communicating with each other.

A. Howard designs a MR-SLAM framework [72], assuming that the initial positions of all robots are unknown. In this decentralized approach, each robot's previously stored observation sequences are combined in a single environment map when rendezvous occurs. A reference robot incrementally builds a map

and localizes its poses. Other robots just accumulate their control input and observation obtained from equipped sensors. If they meet with the reference robot, they are initialized at that time, and their past and current poses and surrounding maps are estimated in the unified coordinate. Carlone et al. [73] produced a new decentralized method, based on RBPF similar to [72]. It considers limited communication taking into account the distance between robots and the original positions of the robots are unknown. The method uses cameras for robots' mutual detection and it only allows the partial maps alignment when rendezvous happens. The presented solution showed efficiency and robustness, successfully building a map of a real world environment. In [91], a cooperative decentralized SLAM system is examined, on which robots need to estimate the maps and the states of all the other robots assuming that the communication between them is limited and the connection is dynamic. It is mathematically proven that an estimation equivalent to a centralized system can be obtained by all robots in the network in a decentralized way. Besides, the robot only needs to consider its own information from the topological network in order to detect when the equivalent centralized system is obtained. Teresa A et al. concentrated the cooperation between aerial and ground robots [74, 75]. They consider some events between robots such as rendezvous, feature correspondences and absolute localization measurements for loop-closing. But they have an assumption that the robots know their pose relative to one another. In addition, a batch algorithm is performed in the optimization over the

transformations, which requires heavy computational load. C. Yunfei et al. implemented a multi-robots cooperative online FastSLAM algorithm using the leader robot and the follower robot [78]. The leader robot was defined and measured as a Robot landmark (RL) by the follower robot. The RL's posteriori estimation was used as the leader robot's priori estimation for its next pose prediction. The performance of the algorithm was verified from the simulation results.

S. M. Chen et al. in [92] presented a multi-robot FastSLAM algorithm by combining Kalman-Consensus Filter (KCF) to improve the accuracy of localization and mapping. Electromagnetism like mechanism algorithm is introduced to get a new batch of high quality particles which is similar to PSO. In the feature update part, the KCF is performed using information of multiple robots. However, they basically assumed known data association for features and the known initial condition in the simulation. The information filter and the information consensus filter are employed together in [93]. To investigate consensus effects, they compare the results from the information consensus filter and covariance intersection (CI). But the known conditions are still assumed in the simulation.

A map merging technique was proposed by assuming that robots do not know their initial poses [89]. It generally increases the accuracy of map merging using one-way observation between robots. However, it does not consider the compensation of the map itself because it just focuses on the map alignment

only. Besides, since robots cannot be suddenly emerged and manually controlled by human at the beginning, the given initial conditions are not critical issues anymore. In our previous work [76, 77, 79], we proposed a multi-robot SLAM framework. Under the unknown initial conditions, robots initialize their poses when the first rendezvous with the reference robot occurs. Subsequently, the poses and maps between the *N*-1th and the *N*th rendezvous points are compensated. In addition, current poses for two robots are fused by covariance intersection (CI) [77] or a Kalman consensus information filter [79] at the *N*th rendezvous point. However, our approaches does not consider the data consensus for common features when several robots share them.

1.3 Contributions

A cooperative RBPF-SLAM framework is proposed in this dissertation. It includes not only the improvements of RBPF-SLAM for single robot and the extension of RBPF-SLAM for multi-robot system. Especially, the proposed framework can deal with the geometric relation among particles and inter-robot measurements among robots, as shown in Fig.1.4. The conventional RBPF-SLAM framework consists of four parts, i.e. sampling, measurement update, importance weight and resampling parts. In this dissertation, the sampling and measurement update parts are enhanced using the consensus scheme when inter-robot measurements are given. The importance weight and resampling parts are also improved by the weight compensation and particle formation

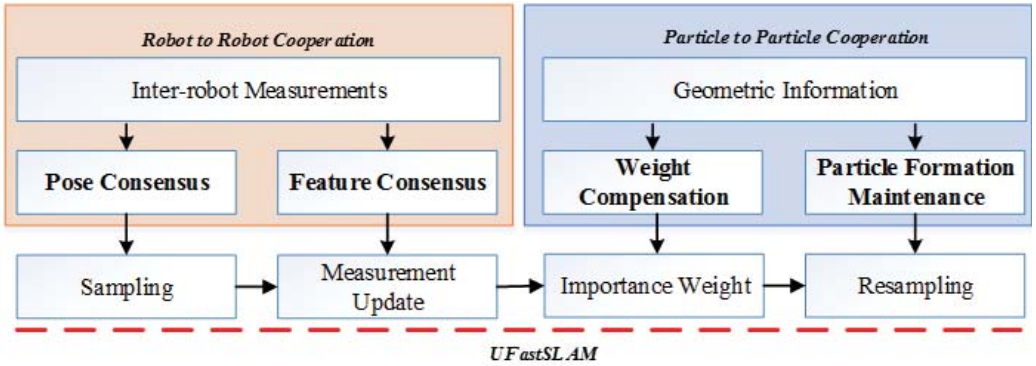


Figure 1.4 The proposed RBPF-SLAM framework. The sampling and measurement update parts are enhanced using the consensus scheme when inter-robot measurements are given. The importance weight and resampling parts are improved by the weight compensation and particle formation maintenance which employ geometric information among particles.

maintenance which employ geometric information among particles. In addition, the proposed framework seeks to cope with the inherent problems of RBPF-SLAM, i.e. a particle depletion problem and the data association problem when a finite number of particles is used in large environments. The improvements are described in detail as follows:

- It uses the geometric relation among particles in calculating the importance weight and maintaining a particle formation, which denotes *particle to particle cooperation* in this dissertation. We assign more accurate weights to particles by clustering them using the clustering algorithms such as the k-means algorithm and the expectation-maximization algorithm according to a heuristic and piecewise weighted average compensation schemes. In addition,

particles constitute an adaptive triangular mesh formation to maintain multiple data association hypotheses without the rejection and replication of the original resampling step. Its outstanding accomplishments are verified on simulations using a formal simulator and tests using the Car park dataset, the Victoria Park dataset, the indoor dataset, and the outdoor dataset by comparing the standard FastSLAM 2.0, PSO based FastSLAM, UFastSLAM, particle fission based UFastSLAM and PSO based UFastSLAM.

- The consensus scheme, i.e. the Kalman consensus information filter (KCIF) is also involved in the proposed RBPF-SLAM framework to deal with inter-robot measurements. The inter-robot measurement occurs when two robots meet each other, which is a rendezvous event. Their current poses are more correctly estimated via the KCIF, which results in an improvement of the sampling step in the proposed RBPF-SLAM framework. When two robots share a common feature, the feature can also be more correctly estimated, which is called a feature sharing event. This correct estimation results in an enhancement of the feature update step in the proposed RBPF-SLAM framework. The accumulated SLAM errors are dramatically reduced from the above mentioned two events, which is said to be *robot to robot cooperation*. In several simulations and some experiments, the SLAM performance of the proposed framework is verified and is better than one of

conventional solutions in terms of pose accuracy and feature accuracy.

1.4 Organization

This dissertation is organized as follows. The fundamental framework i.e. unscented Rao-Blackwellized particle filter based SLAM, called UFastSLAM and the required techniques such as the Kalman consensus information filter (KCIF) for data fusion are described in Chapter 2. Chapter 3 presents the proposed RBPF-SLAM framework to deal with the particle depletion problem, the data association problem and the filter convergence problem in terms of particle to particle cooperation. For the robot to robot cooperation, a consensus scheme based on the KCIF are presented and combined into the proposed framework in Chapter 4. To test the performance of core contents of the proposed framework, several simulations using the formal simulator [80] are performed in Chapter 5. The results from real indoor and outdoor experiments and some tests using a formal datasets, i.e. Car Park dataset and Victoria dataset, are represented in Chapter 6. Chapter 7 concludes this dissertation while briefly summarizing all of the chapters.

Chapter 2

Fundamental Techniques for Multi-robot SLAM

In this chapter, some fundamental issues for multi-robot SLAM are described. As a fundamental framework, a Rao-Blackwellized particle filter are introduced which takes advantages of the multi-hypothesis data association property and the linear computational complexity. In particular, unscented factored solution to SLAM (UFastSLAM) is explained in detail, which will be improved in the following chapters. To deal with the data fusion for the multiple robots, covariance intersection (CI) and the Kalman consensus filter (KCF) are described using the inter-robot measurement.

2.1 Rao-Blackwellized Particle Filter based SLAM

Rao-Blackwellized Particle Filter based SLAM has many advantages in terms of time-complexity and filter convergence compared to a traditional methods such as EKF-SLAM [28, 57]. Also, it was implemented as FastSLAM (Factored Solution to the SLAM problem), which has two standard versions with different the proposal distributions. In this dissertation, we consider the unscented FastSLAM (UFastSLAM) [60, 61], which is a robust and efficient solution to the SLAM problem by using unscented Kalman filters in both the sampling step and measurement update step. Like FastSLAM, it also shows a factored representation of the SLAM posterior over robot poses and maps, as follows:

$$\begin{aligned} p(x_{1:t}, M \mid z_{1:t}, u_{1:t}, c_{1:t}) &= p(x_{1:t} \mid z_{1:t}, u_{1:t}, c_{1:t})\, p(M \mid x_{1:t}, z_{1:t}, u_{1:t}, c_{1:t}) \\ &= \underbrace{p(x_{1:t} \mid z_{1:t}, u_{1:t}, c_{1:t})}_{\text{path posterior}} \underbrace{\prod_{n=1}^{N_f} p(m_n \mid x_{1:t}, z_{1:t}, u_{1:t}, c_{1:t})}_{\text{landmark estimators}}, \end{aligned} \quad (2.1)$$

Here $x_{1:t}$, $z_{1:t}$ and $u_{1:t}$ are the robot pose, sensor observation and control input up to time *t*, respectively. Also, $c_{1:t}$ denotes the set of data associations until time *t*, in which each c_t specifies the identity of the landmark observed at time *t*. *M* denotes the entire map consisting of N_f observed features. This factored representation means that if a path of the robot is given, each landmark can be independently estimated by its own EKF filter. Each particle *i* in UFastSLAM is denoted by [60],

$$X_t^{[i]} = \left\langle x_t^{[i]}, \mu_{1,t}^{[i]}, \sum_{1,t}^{[i]}, \ldots, \mu_{N,t}^{[i]}, \sum_{N,t}^{[i]} \right\rangle, \quad (2.2)$$

where [i] indicates the index of the particle, $x_t^{[i]}$ is the ith particle's pose, and $\mu_{N,t}^{[m]}$ and $\sum_{N,t}^{[m]}$ are the mean and the covariance of the Gaussian distribution representing the Nth feature location conditioned on the robot path. Altogether, these elements form the ith particle $X_t^{[i]}$. Figure 2.1 represents the set of N particles.

	Robot path	Landmark #1	Landmark #2		Landmark #N_f
Particle $X_t^{[1]}$	$x_{1:t}^{[1]} = \{(x\ y\ \theta)^T\}_{1:t}^{[1]}$	$\mu_1^{[1]}, \Sigma_1^{[1]}$	$\mu_2^{[1]}, \Sigma_2^{[1]}$	$\cdots$	$\mu_{N_f}^{[1]}, \Sigma_{N_f}^{[1]}$
Particle $X_t^{[2]}$	$x_{1:t}^{[2]} = \{(x\ y\ \theta)^T\}_{1:t}^{[2]}$	$\mu_1^{[2]}, \Sigma_1^{[2]}$	$\mu_2^{[2]}, \Sigma_2^{[2]}$	$\cdots$	$\mu_{N_f}^{[2]}, \Sigma_{N_f}^{[2]}$
	$\vdots$				
Particle $X_t^{[N]}$	$x_{1:t}^{[N]} = \{(x\ y\ \theta)^T\}_{1:t}^{[N]}$	$\mu_1^{[N]}, \Sigma_1^{[N]}$	$\mu_2^{[N]}, \Sigma_2^{[N]}$	$\cdots$	$\mu_{N_f}^{[N]}, \Sigma_{N_f}^{[N]}$

Figure 2.1 A particle set in UFastSLAM. A particle in the set consists of a path estimates and a set of estimates of individual landmark locations with associated covariances.

2.1.1 Sampling Strategy

At first, the state vector $x_t^{[m]}$ is augmented with a control input and the observation, as follows:

$$x_{t-1}^{a[m]} = \begin{bmatrix} x_{t-1}^{[m]} \\ 0 \\ 0 \end{bmatrix}, \quad P_{t-1}^{a[m]} = \begin{bmatrix} P_{t-1}^{[m]} & 0 & 0 \\ 0 & Q_t & 0 \\ 0 & 0 & R_t \end{bmatrix}, \tag{2.3}$$

where $x_{t-1}^{a[m]}$ and $P_{t-1}^{a[m]}$ are the augmented vector for the state and the augmented covariance matrix, respectively. Q_t and R_t are the control input noise covariance and the measurement noise covariance, respectively.

UFastSLAM deterministically extracts sigma points from the Gaussian and passes these points through the nonlinear function. A symmetric set of $2L + 1$ sigma points $\chi_{t-1}^{a[i][m]}$ for the augmented state vector with $L = 7$ can be computed by

$$\begin{aligned}
\chi_{t-1}^{a[0][m]} &= x_{t-1}^{a[m]} \\
\chi_{t-1}^{a[i][m]} &= x_{t-1}^{a[m]} + \left(\sqrt{(L+\lambda)P_{t-1}^{a[m]}}\right)_i \qquad (i = 1,...,L) \qquad , \qquad (2.4) \\
\chi_{t-1}^{a[i][m]} &= x_{t-1}^{a[m]} - \left(\sqrt{(L+\lambda)P_{t-1}^{a[m]}}\right)_{i-L} \qquad (i = L+1,...,2L)
\end{aligned}$$

where subscript i means the ith column of a matrix. The λ is computed by $\lambda = \alpha^2(L+\kappa) - L$. As two parameters, α and κ are determined by 0.002 and 0, respectively.

The motion model of the robot f is characterized by a nonlinear function, and the set of sigma points $\chi_{t-1}^{a[i][m]}$ is transformed by the motion model using the current control $u_t^{[m]}$ with the added control noise component $\chi_t^{u[i][m]}$ of each sigma point as follows:

$$\bar{\chi}_t^{[i][m]} = f(u_t^{[m]} + \chi_t^{u[i][m]}, \chi_{t-1}^{[i][m]}) = \begin{bmatrix} \bar{\chi}_{x,t}^{[i][m]} \\ \bar{\chi}_{y,t}^{[i][m]} \\ \bar{\chi}_{\theta,t}^{[i][m]} \end{bmatrix}. \qquad (2.5)$$

The first two moments of the predicted vehicle state are computed by a linear weighted regression of the transformed sigma points $\bar{\chi}_t^{[i][m]}$:

$$x_{t|t-1}^{[m]} = \sum_{i=0}^{2L} w_g^{[i]} \bar{\chi}_t^{[i][m]}, P_{t|t-1}^{[m]} = \sum_{i=0}^{2L} w_c^{[i]} \left(\bar{\chi}_t^{[i][m]} - x_{t|t-1}^{[m]}\right)\left(\bar{\chi}_t^{[i][m]} - x_{t|t-1}^{[m]}\right)^T , \quad (2.6)$$

where $w_g^{[i]}$ and $w_c^{[i]}$ are weights in the equation. These are computed by

$$w_g^{[0]} = \frac{\lambda}{(L+\lambda)}, \quad w_c^{[0]} = \frac{\lambda}{(L+\lambda)} + \left(1-\alpha^2+\beta\right)$$
$$w_g^{[i]} = w_c^{[i]} = \frac{1}{2(L+\lambda)} \quad (i=1,...,2L). \tag{2.7}$$

Here, the parameter β is used to incorporate the knowledge of the higher order moments of the posterior distribution. In fact, $\beta = 2$. The estimated mean and its covariance of the vehicle state at time *t* are calculated by

$$\bar{N}_t^{[i][m]} = h(\bar{\chi}_t^{[i][m]}, \mu_{\hat{k},t-1}^{[m]}) + \chi_t^{z[i][m]} \tag{2.8}$$

$$\hat{n}_t^{[m]} = \sum_{i=0}^{2L} w_g^{[i]} \bar{N}_t^{[i][m]} \tag{2.9}$$

$$K_t^{[m]} = \underbrace{\sum_{i=0}^{2L} w_c^{[i]} \left(\chi_t^{[i][m]} - x_{t|t-1}^{[m]}\right)\left(\bar{N}_t^{[i][m]} - \hat{n}_t^{[m]}\right)^T}_{\text{cross-covariance(Jacobian term)}} \left(\underbrace{\sum_{i=0}^{2L} w_c^{[i]} \left(\bar{N}_t^{[i][m]} - \hat{n}_t^{[m]}\right)\left(\bar{N}_t^{[i][m]} - \hat{n}_t^{[m]}\right)^T}_{\text{innovation covariance}} \right)^{-1} \tag{2.10}$$

The measurement sigma points $\bar{N}_t^{[i][m]}$ are calculated in (8) using the observation model *h*, characterized by a nonlinear function, with the added measurement noise component $\chi_t^{z[i][m]}$. $\hat{n}_t^{[m]}$ is the predicted measurement. $K_t^{[m]}$ is the Kalman gain in the measurement update. The estimated mean and its covariance of the vehicle state at time *t* are calculated by

$$x_t^{[m]} = x_{t|t-1}^{[m]} + K_t^{[m]}\left(z_t - \hat{n}_t^{[m]}\right) \quad , \tag{2.11}$$

$$P_t^{[m]} = P_{t|t-1}^{[m]} - K_t^{[m]} \left(\sum_{i=0}^{2L} w_c^{[i]} \left(\bar{N}_t^{[i][m]} - \hat{n}_t^{[m]}\right)\left(\bar{N}_t^{[i][m]} - \hat{n}_t^{[m]}\right)^T \right) \left(K_t^{[m]}\right)^T , \tag{2.12}$$

From the Gaussian distribution generated by the estimated mean and covariance of the vehicle, the state of each particle is sampled:

$$x_t^{[m]} \sim N(x_t^{[m]}, P_t^{[m]}) . \tag{2.13}$$

2.1.2 Feature State Estimation

The feature update defines the sigma points using the previously registered mean and covariance of the feature.

$$\begin{aligned}
\chi_{t-1}^{[0][m]} &= \mu_{n_t,t-1}^{[m]} \\
\chi_{t-1}^{[i][m]} &= \mu_{n_t,t-1}^{[m]} + \left(\sqrt{(n+\lambda)\sum_{n_t,t-1}^{[m]}}\right)_i \qquad (i=1,...,n) \\
\chi_{t-1}^{[i][m]} &= \mu_{n_t,t-1}^{[m]} - \left(\sqrt{(n+\lambda)\sum_{n_t,t-1}^{[m]}}\right)_{i-n} \qquad (i=L+1,...,2n)
\end{aligned} \quad , \tag{2.14}$$

where $\mu_{n_t,t-1}^{[m]}$ is the mean of the n_t th feature that is registered in feature initialization step. $\sum_{n_t,t-1}^{[m]}$ is the covariance matrix of the n_t th feature. In this case, $\lambda=\alpha^2(n+\kappa)-n$, and $\lambda=\alpha^2(n+\kappa)-n$ and $\alpha=0.001$ and $\kappa=0$ are appropriate for estimating the feature state. The predicted measurement $\hat{z}_t^{[m]}$ and the Kalman gain $K_t^{[m]}$ are computed by

$$Z_t^{[i][m]} = h\left(\chi^{[i][m]}, x_t^{[m]}\right) \quad (i=0,...,2n) \ , \tag{2.15}$$

$$\hat{z}_t^{[m]} = \sum_{i=0}^{2n} w_g^{[i]} \bar{Z}_t^{[i][m]} \ , \tag{2.16}$$

$$\bar{S}_t^{[m]} = \sum_{i=0}^{2n} w_c^{[i]} \left(\bar{Z}_t^{[i][m]} - \hat{z}_t^{[m]}\right)\left(\bar{Z}_t^{[i][m]} - \hat{z}_t^{[m]}\right)^T + R_t \, , \tag{2.17}$$

where $Z_t^{[i][m]}$ is the transformed sigma points of the *m*th particle. $\sum_t^{[m]}$ determines the cross-covariance between state and observation, which is used to compute the Kalman gain $\bar{K}_t^{[m]}$ as follows:

$$\bar{\sum}_t^{[m]} = \sum_{i=0}^{2L} w_c^{[i]} \left(\chi_t^{[i][m]} - \mu_{n_t,t-1}^{[m]}\right)\left(\bar{Z}_t^{[i][m]} - \hat{z}_t^{[m]}\right)^T , \tag{2.18}$$

$$\bar{K}_t^{[m]} = \bar{\sum}_t^m \left(\bar{S}_t^m\right)^{-1} \ , \tag{2.19}$$

Finally, the mean $\mu_{n_t,t}^{[m]}$ and the covariance $\sum_{n_t,t}^{[m]}$ of the n_t th feature are updated as follows:

$$\mu_{n_t,t}^{[m]} = \mu_{n_t,t-1}^{[m]} + \bar{K}_t^{[m]}\left(z_t - \hat{z}_t^{[m]}\right), \tag{2.20}$$

$$\sum_{n_t,t}^{[m]} = \sum_{n_t,t-1}^{[m]} - \bar{K}_t^{[m]}\bar{S}_t^{[m]}\left(\bar{K}_t^{[m]}\right)^T, \tag{2.21}$$

where z_t is the true measurement. The Cholesky factorization is used in this feature update to make the algorithm more stable numerically.

2.1.3 Calculating Importance Weight and Resampling Strategy

To assign the weight to each particle, the importance weight $w_t^{[m]}$ can be denoted by

$$w_t^{[m]} = \frac{\text{target distribution}}{\text{proposal distribution}}$$

$$= \frac{p(s^{t,[m]} \mid z^t, u^t, c^t)}{p(s^{t-1,[m]} \mid z^{t-1}, u^{t-1}, c^{t-1})\, p(s_t^{[m]} \mid s^{t-1,[m]},\ z^t, u^t, c^t)}$$

Like FastSLAM 2.0, the importance weight of UFastSLAM can be computed by considering the most recent observations, and it is given as follows:

$$w_t^{[m]} = \left|2\pi L_t^{[m]}\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z - \hat{z}_t^{[m]})^T \left(L_t^{[m]}\right)^{-1}(z - \hat{z}_t^{[m]})\right\}, \tag{2.22}$$

$$L_t^{[m]} = \left(\sum_t^{x,n[m]}\right)^T \left(P_t^{[m]}\right)^{-1} \sum_t^{x,n[m]} + \bar{S}_t^{[m]}, \tag{2.23}$$

where $\sum_t^{x,n[m]}$, $P_t^{[m]}$ and $\bar{S}_t^{[m]}$ are the cross-covariance, the covariance and the innovation covariance of the *m*th particle, respectively. $z - \hat{z}_t^{[m]}$ is the innovation vector. Suppose that there is a feature in a map. Each particle should estimate the feature. However, if data association is failed, some particles can

estimate two or more wrong features, which is the data association problem. It may cause that wrong weights are assigned to the particles according to their innovation vector and even lead to filter divergence.

In the resampling step, the death or life of particles is up to the score of the importance factor. Some particles with relatively large mismatches with their target, called bad particles, are rejected and another particles with relatively small mismatches with the target, called good particles, are replicated according to the resampling scheme. Most resampling techniques used the effective number of particles as an operation criteria, which is computed by,

$$N_{eff} = \frac{1}{\sum_{m=1}^{N} \left(\hat{w}_t^{[m]} \right)^2}, \tag{2.24}$$

where N is the total number of particles and $\hat{w}_t^{[m]}$ is the normalized weight of the mth particle. If the variance of the importance weights increases, the N_{eff} decreases. However, when a finite number of particles is used for SLAM, distinctive particles are decreased over time because this resampling scheme consists of the rejection and replication of particles, which is called the particle depletion problem. The variation of distinct particles over time was experimentally represented by various resampling methods such as RSR - residual systematic resampling, RBR - rank-based resampling and PR - partial resampling [42]. Though the depletion rates of those approaches were different, they could undergo the particle depletion problem. The multiple data association hypotheses, representing one of the powerful capabilities of

UFastSLAM, becomes weaker due to this problem, which is particularly remarkable in a large map, in particular. To maintain multiple data association hypotheses, UFastSLAM should eliminate the rejection and replication drawbacks. In addition, if a particle with a relatively low weight which represents a target well is rejected during the resampling process, the filter can be wrongly converged. The problem regarding convergence also worsens the SLAM performance of UFastSLAM, and even it can be diverged. To deal with the filter divergence, the importance weight which is approximately and individually calculated, should be more correctly estimated.

2.2 Covariance Intersection (CI) and Kalman Consensus Information Filter (KCIF) in RBPF-SLAM

In multi-robot systems (MRSs), multiple robots can be sharing their information one another and measure the other robot's information. Covariance intersection (CI) and the Kalman consensus information filter (KCIF) are algorithms for combining two or more estimates of state variables. Suppose that two robots, R_1 and R_2, detect a common feature ' a '. The estimated feature state obtained from the robots are $\mu_{R_1}^a$ and $\mu_{R_2}^a$, respectively. In addition, its covariance information can be represented by $\sum_{R_1}^a$ and $\sum_{R_2}^a$. Here, the CI method is applied to the information fusion as follows:

$$\left(\sum_{new}\right)^{-1} = w\left(\sum_{R_1}^a\right)^{-1} + (1-w)\left(\sum_{R_2}^a\right)^{-1}, \tag{2.25}$$

$$\left(\mu_{new}\right)^{-1} = \sum_{new}\left(w\left(\sum_{R_1}^a\right)^{-1}\mu_{R_1}^a + (1-w)\left(\sum_{R2}^a\right)^{-1}\mu_{R_2}^a\right), \tag{2.26}$$

where μ_{new} and $\sum_{new}$ are the fused state and covariance. These variables are used in the next step. If we adopt the KCIF, the information fusion is operated according to Algorithm 1. As we can see, the information vector and information matrix as well as state variables are required in the KCIF. The information vector and information matrix depends on the measurement. In MRSs, N_i is the set of neighbors around the ith robot R_i , which is defined according to the communication range in practice. In terms of the optimality, stability, and performance, the KCIF is better than the CI method. In addition, CI does not consider the covariance and sets weights heuristically. The comparison was studied in [93]. In RBPF-SLAM, the Kalman consensus filter that is not the information form can be applied for landmark consistency correction as follows:

$$\hat{z}_{j,k}^{t,i} = h(\mu_{j,k-1}^{t,i}, x_k^{t,i}), \tag{2.27}$$

where $h(\cdot)$ is the observation model. $\hat{z}_{j,k}^{t,i}$ is the prior estimate of the robot t with particle i to the jth landmark at time k. $\mu_{j,k-1}^{t,i}$ is the updated jth landmark of the robot t with particle i until k-1. It is assumed that the detected landmarks are stationary; then their prior estimate values are themselves at time k. Landmark location update step by consensus-based filtering is thus given by

$$\mu_{j,k}^{t,i} = \mu_{j,k-1}^{t,i} + K_{j,k}^{t,j}(z_{j,k}^{t,i} - \hat{z}_{j,k}^{t,i}) + \varepsilon \sum_{n \in N_{j,k-1}} \left(\mu_{j,k-1}^{n,\max} - \mu_{j,k-1}^{t,i} \right), \tag{2.28}$$

$$\sum_{j,k}^{t,i} = \sum_{j,k-1}^{t,i} - \sum_{j,k-1}^{t,i} \left(H_j^{t,i} \right)^T \left(H_j^{t,i} \sum_{j,k-1}^{t,i} \left(H_j^{t,i} \right)^T + R^i \right)^{-1} H_j^{t,i} \sum_{j,k-1}^{t,i}, \tag{2.29}$$

where ε is the step size, $N_{j,k-1}$ is the number of robots to take the same

observed landmark j at time k-1, and $\mu_{j,k-1}^{n,\max}$ denotes the feature by the jth feature with the maximum weight of the nth robot at time k-1. Notice that, in (2.28), this method only requires prior estimates exchange between robots which observe the same landmarks. The Kalman gain $K_{j,k}^{t,j}$ is given by

$$K_{j,k}^{t,j} = \sum_{j,k-1}^{t,j} \left(H_j^{t,i} \right)^T \left(H_j^{t,i} \sum_{j,k-1}^{t,j} \left(H_j^{t,i} \right)^T + R_k^t \right)^{-1} \tag{2.30}$$

In practical applications, by considering the limited network communication broadband, packet-dropping, delay, and other sorts of interference, the algorithm may be of weak anti-interference and poor robustness, if there was too much information exchange between robots. In addition, rendezvous situations can also be dealt with by the Kalman consensus filter because inter-robot measurements occur more often in this case than in case of common feature detection.

TABLE 2.1

ALGORITHM FOR KALMAN-CONSENSUS INFORMATION FILTER

Algorithm 1. Kalman-Consensus Information Filter

Given $\sum_i = \sum_0$, $\mu_i = \mu_0$ at time $k = 0$, and messages

$m_j = \{u_j, U_j, \bar{x}_j\}; j \in J_i = N_i \cup \{i\}$

1: Obtain measurement z_i with covariance R_i

2: Compute information vector and matrix of node *i*

$$u_i = H_i^T R_i^{-1} z_i, \quad U_i = H_i^T R_i^{-1} H_i$$

3: Broadcast message $m_j = \{\mu_j, U_j, \bar{x}_j\}$ to neighbors.

4: Receive messages from all neighbors

5: Fuse information matrices and vectors

$$y_i = \sum_{j \in J_i} u_j, S_i = \sum_{j \in J_i} U_j$$

6: compute the Kalman-Consensus state estimate

$$\hat{\mu}_i = \bar{\mu}_i + M_i(y_i - S_i \bar{\mu}_i) + \gamma \sum_{j \in N_i} (\bar{\mu}_j - \bar{\mu}_i)$$

$$M_i = \left((\sum_i)^{-1} + S_i \right)^{-1}$$

$$\gamma = \varepsilon / \left(\|\sum_i\| + 1 \right), \ \|X\| = tr(X^T X)^{\frac{1}{2}}$$

7: Update the state of the local information filter

$\sum_i \leftarrow AM_i A^T + BQB^T$, $x_i \leftarrow A\hat{x}_i$

Chapter 3

Particle-to-Particle Cooperation in RBPF-SLAM

To overcome the particle depletion problem and data association problem as mentioned in the previous section, the particle cooperation can be geometrically and statistically considered in the Rao-Blackwellized particle filter based SLAM framework, which is called relational Rao-Blackwellized particle filter based SLAM [52]. The main process is divided into two parts. One is to improve the accuracy of individually calculated weights such that the posterior distribution is approximated more precisely. In this part, geometric information and the weights of particles are merged probabilistically. In addition, weights are adaptively computed under the merged information. Based on the enhanced weights of particles, they constitute a particle formation and keep the formation without the rejection or replication existing in the

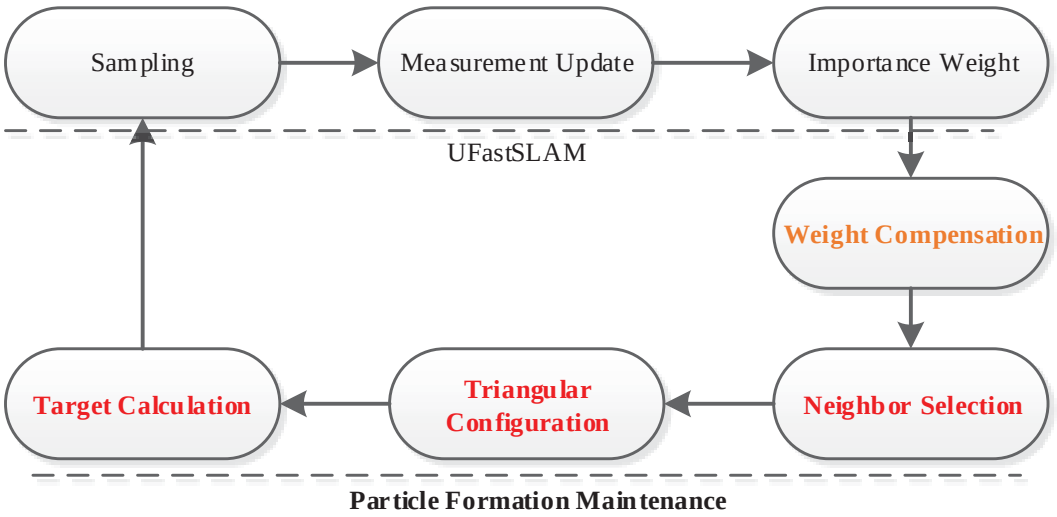


Figure 3.1 Block diagram for the proposed RBPF-SLAM using particle to particle cooperation. The resampling step is removed, and weight compensation and particle formation maintenance consisting of four steps are added to UFastSLAM.

traditional resampling step. Here, an adaptive triangular mesh structure is used for the particle formation [70, 71]. It makes relational RBPF-SLAM converge more correctly and retain the multiple data association hypotheses, which is strong property for RBPF-SLAM estimation. In the following sections, we will discuss the structure of relational RBPF-SLAM in detail.

3.1 Weight Compensation using Particle Cooperation

RBPF-SLAM produces the weights of particles using the ratio between target distribution and the proposal distribution, as determined by (2.22). Although the measurement and the control input are also involved in the proposal distribution, it is impossible to calculate the weight precisely due to the unknown target distribution, measurement noise, and especially unknown

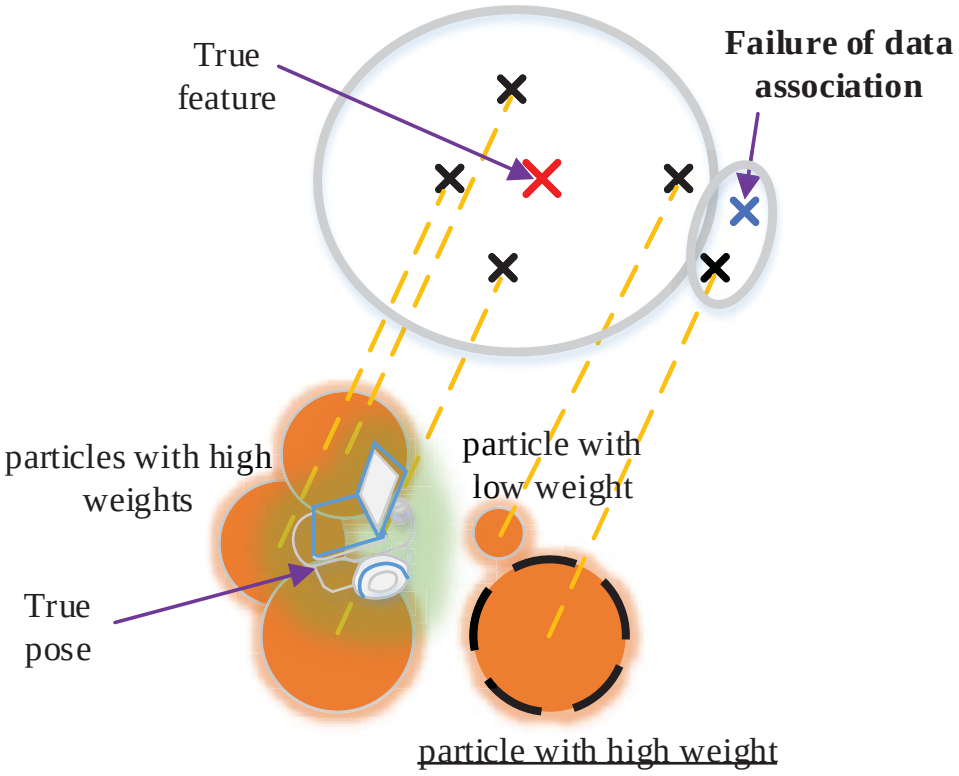


Figure 3.2 Relation between unknown correspondence and the weight assignment.

correspondence. The problem of unknown correspondence is serious because it is related to data association as shown in Fig. 3.2. If some particles estimate a feature accurately, relatively high weights are assigned to them. However, if the *m*th particle detects a feature that is generated from the failure of data association, relatively high weight is also assigned to the particle by the innovation term, $z - \hat{z}_t^{[m]}$, in the weight computation equation, which is the problem of unknown data association. Thus, the weight $w_t^{[m]}$ should be formulated as follows:

$$w_t^{[m]} = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(s^{t,[m]} \mid z^t, u^t, c^{t,[m]})}{p(s^{t-1,[m]} \mid z^{t-1}, u^{t-1}, c^{t-1,[m]})\, p(s_t^{[m]} \mid s^{t-1,[m]},\ z^t, u^t, c^{t,[m]})}, \quad (3.1)$$

where $c^{t,[m]}$ is the correspondence of the *m*th particle. Until now, this term has not been considered. To reduce the data association error from $c^{t,[m]}$, we add compensation term $wc_t^{[m]}$ to (2.22) as follows:

$$w_t'^{[m]} = \underbrace{\left|2\pi L_t^{[m]}\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z-\hat{z}_t^{[m]})^T \left(L_t^{[m]}\right)^{-1}(z-\hat{z}_t^{[m]})\right\}}_{w_t^{[m]}} + wc_t^{[m]}, \quad (3.2)$$

where the added term is used to assign more accurate weights to particles; it is hard to compute this term as a closed form. But it can be obtained from the post-processing step after the importance weight step. The geometric relationship between the particles can be considered to alleviate the problem because the conventional weight for each particle is individually calculated in RBPF-SLAM.

For the compensation, the weight was recalculated using support vector regression which resulted in a smoothed posterior density [47, 48]. During the regression, the particle with the maximum weight or the minimum weight was regarded as a hindrance to the approximation of the true posterior distribution. The particles, however, has a very important meaning. It can determine whether or not the target exists nearby, which implies that the particle demonstrates a higher or lower possibility for the target to exist than others. By reflecting this

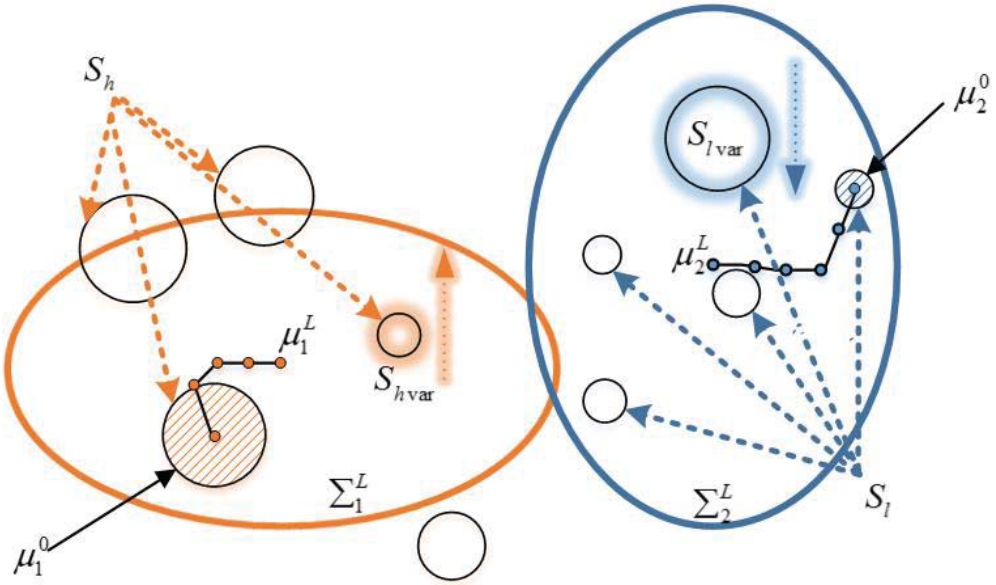


Figure 3.3 Illustration of the weight compensation. In two clusters, improperly assigned particle's weights are compensated for according to the adaptive weight compensation scheme.

fact, we construct two clusters consisting of an area in which the target is highly involved and another area in which the target most likely does not belong. For clustering, there are two algorithms, i.e. expectation maximization (EM) and standard k-means algorithms. In case of EM, the expected value of the likelihood is calculated at the expectation step as follows:

$$T_{[m],i}^{p} = \frac{\pi_i^p f(x_t^{[m]}; \mu_i^p, \Sigma_i^p)}{\sum_{j=1}^{k} \pi_j^p f(x_t^{[m]}; \mu_j^p, \Sigma_j^p)} \qquad m = 1, \ldots, N, \tag{3.3}$$

where $T_{[m],i}^{p}$ is the *i*th expectation of a new likelihood for the *m*th particle after the *p*th iteration, and *k* denotes the number of multivariate normal distributions. In fact, $k = 2$. *N* is the number of sampled particles. $f(\cdot)$ is the probability

density function of a multivariate normal. In addition, a set of unobserved latent data is a set of clusters. A vector of unknown parameters θ is defined by

$$\theta_{un} = \{\mu_1, \mu_2, \Sigma_1, \Sigma_2, \pi\}, \tag{3.4}$$

where μ_1 and μ_2 are the means of normal distributions. Their covariances are represented by Σ_1 and Σ_2. If the *i*th particle is involved in a cluster, then the probability is can be represented as π. All elements of the parameter vector θ are updated during the maximization step as follows:

$$\mu_1^{(t+1)} = \frac{\sum_{i=1}^{N} T_{[m],1}^{p} x_{t1}^{[m]}}{\sum_{i=1}^{N} T_{[m],1}^{p}}, \qquad \mu_2^{(t+1)} = \frac{\sum_{i=1}^{N} T_{[m],2}^{p} x_{t2}^{[m]}}{\sum_{i=1}^{N} T_{[m],2}^{p}}, \tag{3.5}$$

$$\Sigma_1^{(t+1)} = \frac{\sum_{i=1}^{N} T_{[m],1}^{p} (x_i - \mu_1^{(t+1)})(x_i - \mu_1^{(t+1)})^T}{\sum_{i=1}^{N} T_{[m],1}^{p}}, \qquad \Sigma_2^{(t+1)} = \frac{\sum_{i=1}^{N} \tau_i^{2,t} (x_i - \mu_2^{t+1})(x_i - \mu_2^{t+1})^T}{\sum_{i=1}^{N} T_{[m],2}^{p}}, \tag{3.6}$$

$$\pi_1^{t+1} = \frac{\sum_{i=1}^{N} T_{[m],1}^{p}}{N}, \qquad \pi_2^{t+1} = \frac{\sum_{i=1}^{N} T_{[m],2}^{p}}{N}, \tag{3.7}$$

These maximized values affects the expectation step, and these steps are performed iteratively until the parameters are converged. After the iteration, particles constitute two clusters using GMM $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$, as we know that EM may converge to a local maximum of the observed data likelihood function because it depends strongly on the initial values. In this dissertation, π consists of constant values and covariance matrices are defined

by the 2x2 identity matrix. Also, the initial points μ_1^0 and μ_2^0 are designed as follows:

$$\mu_1^0 = x_t^{[m_1]}, \quad \mu_2^0 = x_t^{[m_2]}, \tag{3.8}$$

where $m_1 = \arg\max_{m'} w_t^{[m']}$ and $m_2 = \arg\min_{m'} w_t^{[m']}$, which implies that each mean of the clusters is highly affected by the particle with the maximum or minimum weight. They move according to the geometric distribution of the particles during EM iterations. After the *L*th iteration, the most recently updated parameter θ^L which consists of $\mu_1^L, \mu_2^L, \sum_1^L, \sum_2^L$ and $\pi_{1,2}^L$ are generated and all particles are included in two clusters constructed by the parameter. If the above mentioned process should be performed quickly, k-means clustering can be adopted, which tends to find clusters of comparable spatial extent. It is usually similar to the EM algorithm for mixtures of Gaussian distributions via an iterative refinement approach. Additionally, it uses cluster centers to model the data. It has two steps, i.e. assignment step and update step. In the assignment step, each particle is assigned to exactly one S^p as follows:

$$S_i^p = \{x_t^{[m]} : \left\| x_t^{[m]} - \mu_i^p \right\|^2 \le \left\| x_t^{[m]} - \mu_j^p \right\|^2 \ \forall j, 1 \le j \le k\}, \tag{3.9}$$

where, $k = 2$, $1 \le i \le 2$. μ_i^p is the mean of the *i*th set after the *p*th iteration. Using the set S^p, the new means to be the centroids of the observations in the new clusters are calculated by

$$\mu_i^{p+1} = \frac{1}{\left| S_i^p \right|} \sum_{x_t^{[m]} \in \left| S_i^p \right|} x_t^{[m]}, \tag{3.10}$$

After the L-iteration, μ_1^L, μ_2^L, S_1^L and S_2^L can be obtained. These variables

work just like results of the EM algorithm, i.e. μ_1^L , μ_2^L , $\sum_1^L$ and $\sum_2^L$. Figure 3.3 illustrates grouped particles using the mean and variance of clusters. One cluster has a higher probability for containing a robot and the other cluster has a higher probability for not containing the robot. Two sets of particles which are involved in those clusters, are represented as S_h and S_l. Based on these sets, the particles to be compensated are selected. If any particles have relatively low weights in S_h, those are chosen for the compensation. Likewise, in S_l, those with relatively high weights are selected. S_{hvar} and S_{lvar} denote the sets of selected particles in each cluster. These are also illustrated in Fig.3.3. Because particles for the weight compensation are taken from S_{hvar} and S_{lvar}, the compensation term in the weight calculation is updated by the following adaptive scheme:

$$wc_t^{[m]} = \begin{cases} 0 & \text{,where } x_t^{[m]} \notin S_{h\,\mathrm{var}} \text{ and } x_t^{[m]} \notin S_{l\,\mathrm{var}} \\ (1-\frac{1}{\alpha})w_t^{[m]} + \frac{\beta}{\alpha} w_t^{[m\_near]} & \text{,where } x_t^{[m]} \in S_{l\,\mathrm{var}} \text{ or } x_t^{[m]} \in S_{h\,\mathrm{var}} \\ \frac{\gamma-1}{\gamma} w_t^{[m]} & \text{, where } x_t^{[m\_near]} \text{ not exist} \end{cases} \quad (3.11)$$

where α and β are factors which serve to compute the weighted average between $w_t^{[m]}$ and $w_t^{[m\_near]}$. $w_t^{[m\_near]}$ denotes the weight of the closest particle from the *m*th particle which is not to be compensated. If there is no $x_t^{[m\_near]}$ around $x_t^{[m]}$, $w_t^{[m]}$ is directly compensated by γ. In $S_{h\,\mathrm{var}}$, γ is empirically assigned as $\frac{10}{9}$. Otherwise, γ is $\frac{10}{11}$.

The above mentioned adaptive process, called piecewise average based

weight compensation scheme, is summarized in Algorithm 2. After the weights of all particles are calculated, μ_1^0 and μ_2^0 are selected according to (3.8). Subsequently, a GMM can be obtained and four sets S_{hvar}, S_{lvar}, S_h, and S_l are determined. Lastly, the weights of all particles are compensated using (3.11).

TABLE 3.1

ALGORITHM FOR WEIGHT COMPENSATION

Algorithm 2. Piecewise Average based Weight Compensation
Given $w_t^{[m]}$ for all particles
1: Determine the initial states μ_1^0 and μ_2^0 according to (3.8)
2: Obtain GMM $N(\mu_1, \sum_1)$ and $N(\mu_2, \sum_2)$ according to (3.5) and (3.6)
3: Set up four sets S_{hvar}, S_{lvar}, S_h, and S_l
4: Compensate the weights of all particles according to (3.11)

3.2 Applicability of Particle Formation Maintenance

The weights of particles can be more properly assigned to particles through the weight compensation technique. Although the wrong rejection and wrong replication of particles are reduced and the particle filter may converge more accurately, the particle depletion problem inevitably occurs over time in a large environment. This phenomenon does not disappear as long as particles with relatively low weights and particles with relatively high weights are rejected or

replicated, respectively.

They can be eliminated by considering the particle swarm characteristics, i.e. a particle formation maintenance technique (PFM) that is proposed in this paper. The basic structure is generated by the adaptive triangular mesh approach. It was originally utilized with a swarm of robots [71, 72]. It has some advantages of the fact that, among all the possible types of n-polygons, the triangular mesh is highly scalable, and less influenced by the number of neighboring robots. Likewise, it can also be adaptively applied to the formation of particles and to prevent filter divergence for the following reasons. In terms of the particle properties, each particle includes the robot pose, which is denoted by

$$X_t^{[m]} = \{x_{x,t}^{[m]}, x_{y,t}^{[m]}, x_{\theta,t}^{[m]}, \mu_{1,t}^{[m]}, \sum_{1,t}^{[m]}, \cdots, \mu_{N_f,t}^{[m]}, \sum_{N_f,t}^{[m]}\} \supset R_t = \{R_{x,t}, R_{y,t}, R_{\theta,t}\} \quad (3.12)$$

where $\mu_t^{[m]}$ and $\sum_t^{[m]}$ denote the mean and the covariance of the observed features for the *m*th particle at a time *t*, respectively. In addition, $x_{x,t}$, $x_{y,t}$ and $x_{\theta,t}$ represent the pose of the *m*th particle and R_t means robot pose at a time *t*. This indicates that the adaptive triangular mesh structure can be geometrically carried out to maintain the formation of particles. Also, the objectives are very similar to each other. In case of robot swarms, they enclose a source of contamination according to their contamination density to prevent the contamination from spreading out. The particles with different weights should estimate a real robot pose well by maintaining multiple data association hypotheses. Another fact which is immutable is the mean pose of the particles

before and after performing the PFM. Essentially, the mean of particles is the most accurate representation for an estimated robot pose as well as the mean and the variance of the observed features [94]. While carrying out the PFM, keeping the consistency regarding the mean of the particles implies that particles to be diverged are simply adjusted toward it. For these reasons, PFM based on the triangular mesh structure can be adaptively applied to UFastSLAM.

3.3 Particle Formation Maintenance

In this section, the process of PFM is described in detail. Figure 3.4 shows a comparison between the conventional resampling scheme and the proposed resampling scheme. Some particles with high-weight are replicated and some particles with low-weight are rejected in the conventional resampling scheme. Unlike the conventional resampling scheme, all particles are aligned with respect to the proposed resampling scheme.

The detailed process can be described as follows. Triangular configuration T_m is initially built and composed of the pose of the mth particle $x_t^{[m]}$ and its two neighbors, $x_t^{[n_1]}$ and $x_t^{[n_2]}$. $x_t^{[n_1]}$ has the shortest Euclidean distance from particle $x_t^{[m]}$. In addition, $x_t^{[n_2]}$ is determined to minimize the sum of two distances, i.e., the distance between $x_t^{[m]}$ and $x_t^{[n_2]}$ and the distance between $x_t^{[n_1]}$ and $x_t^{[n_2]}$ as follows:

$$x_t^{[n_2]} = \arg\min_{[n_2]} \left(d\left(x_t^{[m]}, x_t^{[n_2]}\right) + d\left(x_t^{[n_1]}, x_t^{[n_2]}\right) \right), \qquad (3.13)$$

where $d(A,B)$ is the distance function between pose A and pose B. The weight of T_m, written by W_{T_m}, is determined as follows:

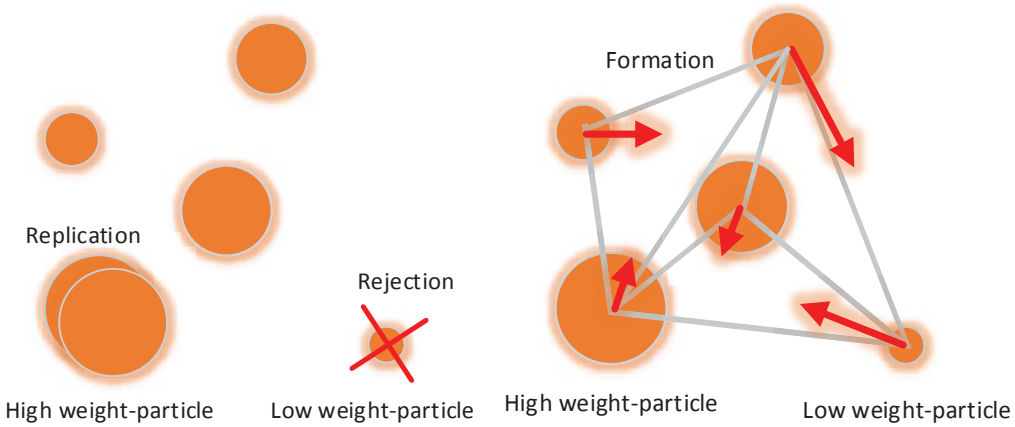
$$W_{T_m} = \frac{1}{3} \sum_{x_t^{[m]} \in T_m} Pw_t^{[m]}, \quad m = 1, \cdots, N, \tag{3.14}$$

$$Pw_t^{[k]} = \frac{1}{N-1} \frac{\sum_k w_t^{[k]} - w_t^{[k]}}{\sum_k w_t^{[k]}}, \quad k = 1, \cdots, N, \tag{3.15}$$

where $Pw_t^{[k]}$ is the transformed weight for the kth particle to operate the PFM. Based on the centroid R_{ct} and W_{T_m}, the target of the mth particle $x_t^{[m]}$ can be calculated by

$$x\_\text{target}_{x,t}^{[m]} = x_{ct.x} + k_{T_m} d_m \cos(\phi + \pi/2) \Big/ \sqrt{3}, \tag{3.16}$$

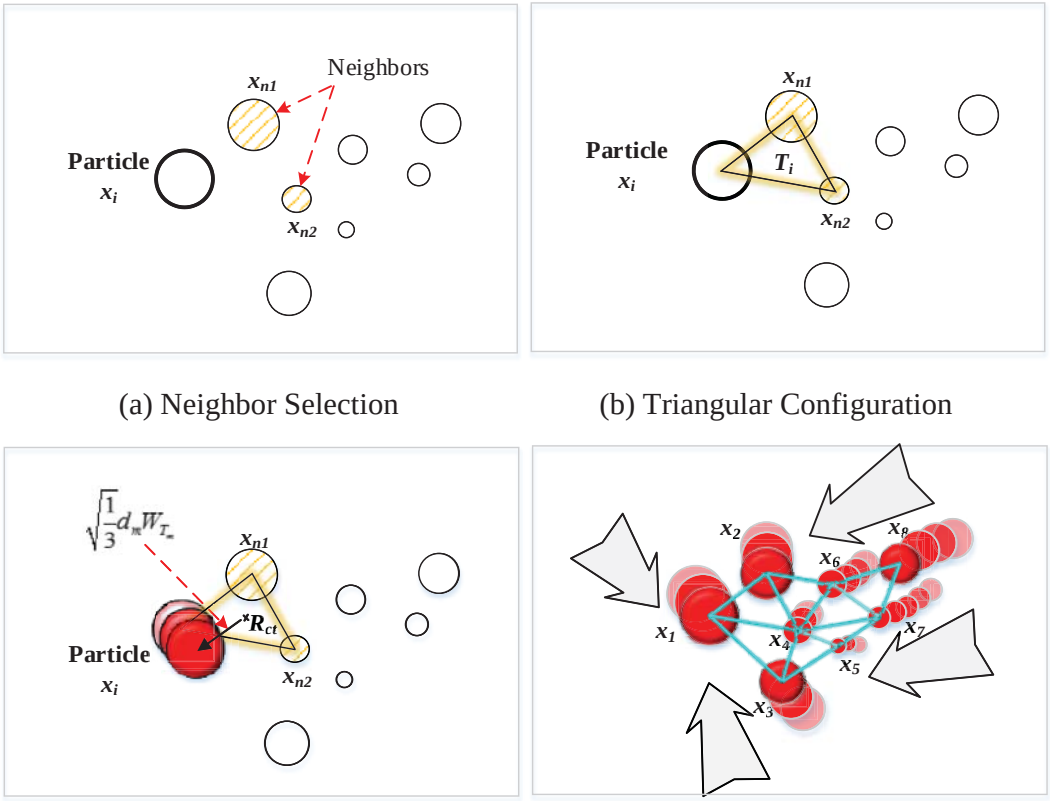
$$x\_\text{target}_{y,t}^{[m]} = x_{ct.y} + k_{T_m} d_m \sin(\phi + \pi/2) \Big/ \sqrt{3}, \tag{3.17}$$



(a) Conventional Resampling (b) Proposed Resampling

Figure 3.4 Comparison between the conventional resampling scheme and the proposed resampling scheme. Unlike the conventional resampling scheme, all particles are aligned with respect to the proposed resampling scheme without the particle rejection and replication.

where $x_{ct.x}$ and $x_{ct.y}$ are the x-y pose for the center of gravity of triangular configuration T_m which consists of $x_t^{[n_1]}, x_t^{[n_2]}$ and $x_t^{[m]}$. $x_t^{[n_1]}$ and $x_t^{[n_2]}$ are the nearest neighbors of $x_t^{[m]}$. ϕ represents the angle which is perpendicular to the line passing through $x_t^{[n_1]}$ and $x_t^{[n_2]}$. k_{T_m} is the average weight factor of T_m. If it is given a constant large value, the filter can be diverged. In this dissertation, we define this factor as follows:



(a) Neighbor Selection (b) Triangular Configuration

(c) Target Calculation and Move (d) Triangular Formation

Figure 3.5 Illustration of PFM process. In (a), the neighbors of each particle are selected and triangular configuration is performed during (b). From the centroid of the triangle, the target where the *i*-th particle will be moved is calculated in (c). (d) shows the PFM after repeated performances from (a) to (c).

$$d_m = \begin{pmatrix} D = \min_m \ dist(x^{t,[m]}, mean(x^t)), & \text{if } D < D_{\max} \\ D_{\max} \quad , & \text{otherwise,} \end{pmatrix}, \qquad (3.18)$$

where the reasoning behind the selection of d_m is that the PFM is sequentially constructed from a particle close to the mean to a particle far from the mean. The orientation of the target is determined by the average of T_m.

The angle of the target, $x\_\text{target}_{\theta,t}^{[m]}$, is determined by the average of three angles in T_m. From the target pose, the final pose of the mth particle, $x_t^{[m]} = \left\{ x_{x,t}^{[m]}, x_{y,t}^{[m]}, x_{\theta,t}^{[m]} \right\}$, is given by

$$x_{x,t}^{[m]} = x_{x,t}^{[m]} + k^{[m]} \left(x\_\text{target}_{x,t}^{[m]} - x_{x,t}^{[m]} \right), \qquad (3.19)$$

$$x_{y,t}^{[m]} = x_{y,t}^{[m]} + k^{[m]} \left(x\_\text{target}_{y,t}^{[m]} - x_{y,t}^{[m]} \right), \qquad (3.20)$$

$$x_{\theta,t}^{[m]} = x_{\theta,t}^{[m]} + k^{[m]} \left(x\_\text{target}_{\theta,t}^{[m]} - x_{\theta,t}^{[m]} \right), \qquad (3.21)$$

where $k^{[m]}$ is the weight factor of the mth particle that is computed by

$$k^{[m]} = \left(1 - \frac{wc_t^{[m]}}{\sum_i wc_t^{[i]}} \right) \qquad (3.22)$$

The weight factor ranges from 0 to 1. If $k^{[m]}$ is equal to 0, the particle has been stopped at its current pose. Otherwise, the particle approaches to the target pose. Since $k^{[m]}$ is proportional to the uncertainty of the mth particle, large uncertainty in the pose of the particle can be compensated after the particle formation maintenance. The process of the PFM is graphically illustrated in Fig. 3.5. Each particle has its two neighbor particles and the triangular configuration is established using three particles. For the triangular structure, particles are selected one by one according to the order of the N-dimension distance array.

TABLE 3.2

ALGORITHM FOR PARTICLE FORMATION MAINTENANCE

Algorithm 3 Particle Formation Maintenance
1: for $i = 1 : N$
2: Selection of neighbors (3.13)
3: Triangular configuration
4: Target calculation (3.16) and (3.17)
5: Final pose calculation (3.19) and (3.21)
6: end for

It is obtained by the distance between the mth particle and the center of the particles that is given by

$$dist^{[m]} = \sqrt{\left(\frac{1}{N} \sum_i x_{x,t}^{[i]} - x_{x,t}^{[m]} \right)^2 + \left(\frac{1}{N} \sum_i x_{y,t}^{[i]} - x_{y,t}^{[m]} \right)^2}, \qquad (3.23)$$

If a particle is selected from the above process, the average weight of the triangular configuration T_m is calculated by taking the average weight of those particles. The target position of the particle is also calculated based on the centroid of the triangular configuration and the weight. Lastly, all particles concurrently move toward their target based on their weights and form a triangular mesh structure.

3.4 Overview of Relational RBPF-SLAM

We examined the cooperation between particles to improve some parts of the UFastSLAM. Algorithm 4 describes the procedure of relational RBPF-SLAM. Sampling, the update of observed feature(s) and the calculation of the importance weight for each particle are performed according to the procedure of UFastSLAM, as described in [60]. To assign the weight more accurately, we divide two probabilistic regions using the particle relationship. The weight of each particle is compensated for by the adaptive compensation scheme in each region. Triangular configurations for all particles are constructed through the connection between neighbors after the weight compensation process. In addition, the weight of each triangular configuration is calculated by averaging the compensated weights of the particles which constitute a triangular formation. At the end of this process, the formation based on the triangular mesh structure is completed and the particles readily retain their formation over

TABLE 3.3

ALGORITHM FOR RELATIONAL RBPF-SLAM

Algorithm 4 Relational RBPF-SLAM
1: Sample particles (2.3) – (2.13)
2: Update features (2.14) – (2.21)
3: Compute the weights of particles according to (2.22)
4: Compensate the weights of particles (Algorithm 2)
5: Maintain the particle formation (Algorithm 3)

time.

3.5 Complexity of Relational RBPF-SLAM

Let us suppose that N and N_f are the number of particles and landmarks, respectively. In FastSLAM or Unscented FastSLAM (UFastSLAM), the complexities of the parts for sampling, measurement update and importance weight are O(N). The resampling part has a complexity level of O($N \cdot N_f$) [or O(NlogN_f), if the landmarks are represented by a binary tree]. In relational RBPF-SLAM, the weight compensation part requires O($N \cdot L$) where L is the number of iterations, as determined by the user. The complexity for particle formation maintenance is O($N \cdot N$) [or O(Nlog$N$), if the poses of particles are represented by a binary tree]. Thus, the complexity of relational RBPF-SLAM depends on N, N_f and L. If N_f is larger than N and L, the computational complexity of the relational RBPF-SLAM is identical to that of FastSLAM and UFastSLAM, which occurs in most large datasets, including the Victoria Park dataset.

Chapter 4

Robot to Robot Cooperation in RBPF-SLAM

Two robots are exploring an environment from distant (and unknown) initial locations. After a while, the robots encounter one another and measure their relative poses. At this time, a compensation filter is initialized using the current robot poses and their measurements. Subsequently, measurements obtained from the two robots are fed to the filter, and thereby fused into a common map. A series of the above mentioned process is considered in Section 4.1. When a rendezvous between two robots occurs or two robots are sharing a common feature, they can exchange their information, i.e. inter-robot measurements. Subsequently, their pose errors and feature errors are largely reduced using the Kalman consensus information filter. These are described in

Section 4.2 and Section 4.3. Section 4.4 summarizes the proposed RBPF-SLAM framework.

4.1 Multi-Robot Initialization in the Unknown Initial Condition

Under the unknown initial condition, the coordination of multiple robots should be unified in one frame. To tackle this problem simply, we put a reference robot R_f as described in [72]. R_f incrementally estimates its pose x_t^f (The particle indices are omitted) and map m_t. Other robots just accumulate their control input u_t and sensor measurement z_t over time. Although the state of the *i*th particle should be written by $x_t^{[i]}$, i is omitted to simplify following expressions. Suppose that R_f meets an arbitrary robot R_n at time $t = a$. R_f measures the relative transformation vector Δ_a^n between R_f and R_n. In addition, the pose of R_n is initialized on the frame of R_f as follows:

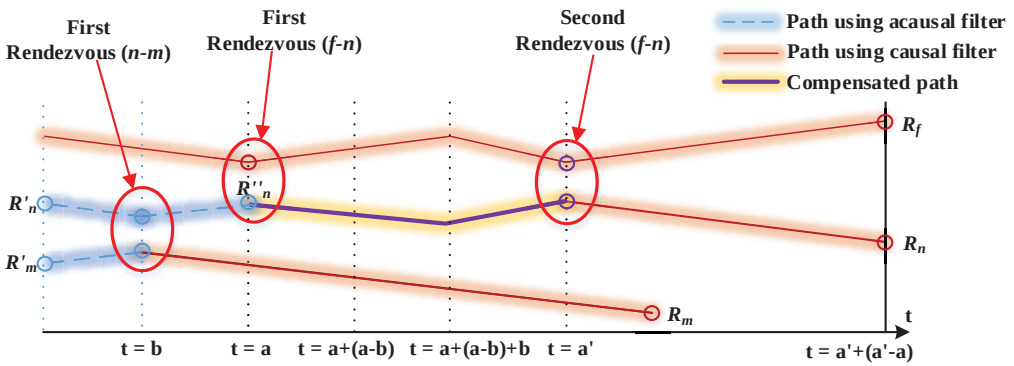


Figure 4.1 An example to explain the proposed RBPF-SLAM framework in MRSs

$$x_a^n = \Delta_a^n \oplus x_a^f \quad , \tag{4.1}$$

where the operator $\oplus$ indicates an appropriate 2D coordinate transform, and x_a^n is the initialized pose of R_n at $t = a$. We assume that the uncertainty of Δ_a^n is negligible. After the initialization, the past poses $x_{1:a-1}^n$ and maps $m_{1:a-1}^n$ of R_n are estimated using the accumulated control input $u_{1:a}$ and sensor measurement $z_{1:a}$. The current pose x_t^n and map m_t^n of R_n are also estimated using the current control input u_t and sensor measurement z_t, simultaneously. In those estimations, two estimators are exploited, which are an acausal filter and a causal filter. The posterior for R_f and R_n is now represented as follows:

$$\begin{aligned} &p(x_{1:t}^n, x_{1:t}^f, m \mid z_{1:t}^f, u_{0:t-1}^f, x_0^f, z_{1:t}^n, u_{0:t-1}^n, \Delta_a^n) = \\ &p(m \mid x_{1:t}^f, z_{1:t}^f, x_{1:a-1}^n, z_{1:a-1}^f, x_{a+1:t}^n, z_{a+1:t}^f) \cdot p(x_{1:t}^f \mid z_{1:t}^f, u_{0:t-1,}^f x_0^f) \cdot \quad , \\ &p(x_{1:a-1}^n \mid z_{1:a-1}^n, u_{0:a-1}^n, x_a^f, \Delta_a^n) \cdot p(x_{a+1:t}^n \mid z_{a+1:t}^n, u_{a:t-1}^n, x_a^f, \Delta_a^n) \end{aligned} \tag{4.2}$$

where $z_{1:t}^f$, $u_{0:t-1}^f$, $z_{1:t}^n$ and $u_{0:t-1}^n$ are the measurements and the control inputs of R_f, and the measurements and the control inputs of R_n until *t*, respectively. If R_n met another robot R_m at $t = b$, R_m is also initialized using the estimated pose of R_n at $t = a + (a - b)$ as computed in (4.1).

4.2 Pose Consensus at Rendezvous Events

If a reference robot R_f meets an arbitrary robot R_n for the first time, R_n is initialized according to (4.1). As shown in Fig.4.2, if the second rendezvous

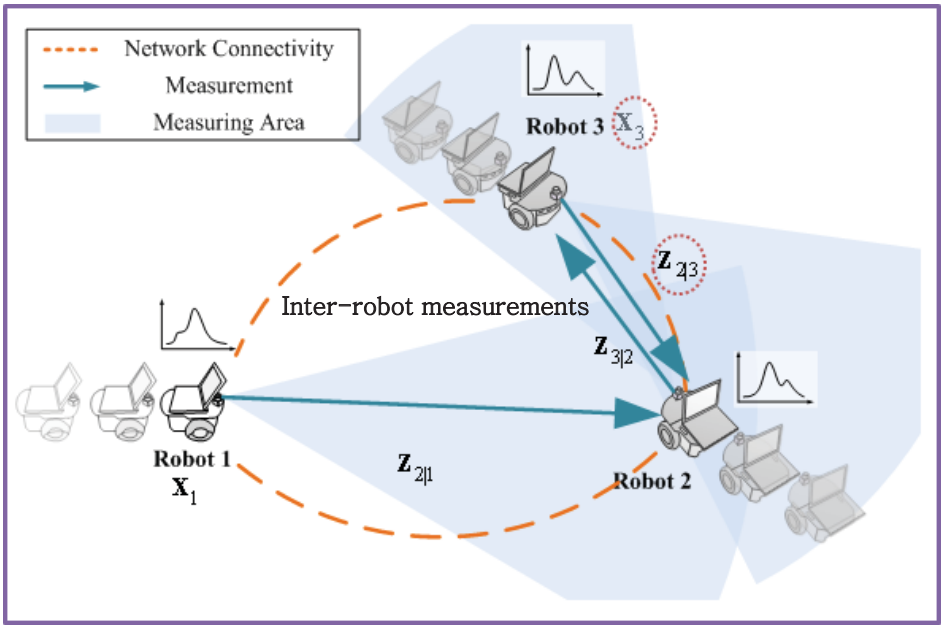


Figure 4.2 Inter-robot measurements at rendezvous point

occurs between them at $t = a'$, the current states of two robots can be compensated more correctly. The Kalman consensus information filter (KCIF) is involved in the proposed SLAM framework for the pose compensation. At first, two robots exchange their prior/predicted state estimate $\hat{x}^i_{a'|a'-1}$ each other. The prior/predicted state $\hat{x}^i_{a'|a'-1}$ is given as the state of the maximum-weight particles, which is computed by

$$\hat{x}^i_{a'|a'-1} = \hat{x}^{i,[k_{\max}]}_{a'|a'-1}, \text{ where } k_{\max} = \arg\max_j w^{[j]}_{a'} \tag{4.3}$$

Subsequently, their information vector $v^i_{a'}$ and information matrix $V^i_{a'}$ are computed by

$$v^{a'}_i = H^T_i R^{-1}_i z_i, \; V^{a'}_i = H^T_i R^{-1}_i H_i \tag{4.4}$$

where z_i is the inter-robot measurement obtained from the *i*th robot and H_i is the Jacobian of the observation model. They can be described as follows:

$$z_i = \begin{bmatrix} \rho_i^2 \\ \theta_i \end{bmatrix}, \tag{4.5}$$

$$h_i = \begin{bmatrix} (x_k - x_i)^2 + (y_k - y_i)^2 & \tan^{-1}\dfrac{(y_k - y_i)}{(x_k - x_i)} - \theta_i \end{bmatrix} \tag{4.6}$$

$$H_i = \begin{bmatrix} -2(x_k - x_i) & \dfrac{(y_k - y_i)}{(x_k - x_i)^2 + (y_k - y_i)^2} \\ -2(y_k - y_i) & \dfrac{-(x_k - x_i)}{(x_k - x_i)^2 + (y_k - y_i)^2} \\ 0 & -1 \\ 2(x_k - x_i) & \dfrac{-(y_k - y_i)}{(x_k - x_i)^2 + (y_k - y_i)^2} \\ 2(y_k - y_i) & \dfrac{(x_k - x_i)}{(x_k - x_i)^2 + (y_k - y_i)^2} \\ 0 & 0 \end{bmatrix}^T \tag{4.7}$$

where h_i is the measurement model which is the 1x2 matrix. z_i and H_i are the 2x1 matrix and the 2x6 matrix, respectively. ρ_i and θ_i are the observed relative distance and the observed relative orientation with Gaussian noise, respectively. Also, (x_i, y_i) is the estimated x-y pose of the *i*th robot and (x_k, y_k) is the prior/predicted x-y state of the *k*th robot. After the calculation, $v_i^{a'}$ and $V_i^{a'}$ are transmitted to the other robot.

In case of R_f, the information vectors and matrices are fused using the transmitted information of R_n as follows:

$$b_{a'}^f = v_{a'}^f + v_{a'}^n, \quad B_{a'}^f = V_{a'}^f + V_{a'}^n, \tag{4.7}$$

where $b^f_{a'}$ and $B^f_{a'}$ are the fused information vector and the fused information matrix, respectively. Like (4.7), the fused information vector and matrix for R_n are calculated via the same procedure. To deal with the fused information vector and matrix, a new state $\hat{x}^{fn,(i)}_{a'|a'-1}$ and a new covariance matrix are set to represent the ith particle's state and covariance for R_f by involving the transmitted state $\hat{x}^{n}_{a'la'-1}$ from the other robot R_n as follows:

$$\hat{x}^{fn,(i)}_{a'|a'-1} = \begin{bmatrix} \hat{x}^{f,(i)}_{a'|a'-1} \\ \hat{x}^{n}_{a'la'-1} \end{bmatrix}, \quad \hat{P}^{fn,(i)}_{a'|a'-1} = \begin{bmatrix} \hat{P}^{f,(i)}_{a'la'-1} & 0 \\ 0 & \hat{P}^{n}_{a'la'-1} \end{bmatrix} \tag{4.8}$$

where, $\hat{x}^{fn,(i)}_{a'|a'-1}$ is the 6x1 vector and $\hat{P}^{fn,(i)}_{a'|a'-1}$ is the 6x6 matrix. Now, the state $\hat{x}^{fn,(i)}_{a'|a'-1}$ is updated as follows:

$$\hat{x}^{fn,(i)}_{a'|a'} = \overline{x}^{fn,(i)}_{a'|a'-1} + M^{f}_{a'}(b^f_{a'} - B^f_{a'}\overline{x}^{fn,(i)}_{a'|a'-1}) + \gamma\left(\hat{P}^{fn,(i)}_{a'|a'-1}\right)\cdot\left(\overline{x}^{n}_{a'|a'-1} - \overline{x}^{fn,(i)}_{a'|a'-1}\right), \tag{4.9}$$

where Kalman gain $M^f_{a'}$ is determined by $\left(B^f_{a'} + \left(\hat{P}^{fn,(i)}_{a'|a'-1}\right)^{-1}\right)^{-1}$, consensus gain is represented by $\gamma\left(\hat{P}^{fn,(i)}_{a'|a'-1}\right)$, and γ is defined by $\varepsilon\Big/\left(1+\left\|\hat{P}^{fn,(i)}_{a'|a'-1}\right\|\right)$. ε is the parameter of the KCIF, which is defined as 0.3. In the consensus term, $\overline{x}^{n}_{a'|a'-1}$ indicates the average state of particles, which is obtained from R_n. The state and the covariance of the ith particle are computed by choosing the upper part of the $\hat{x}^{fn,(i)}_{a'|a'}$ and $M^f_{a'}$. Likewise, R_n is also updated as follows:

$$\hat{x}^{nf,(i)}_{a'|a'} = \overline{x}^{nf,(i)}_{a'|a'-1} + M^{n}_{a'}(b^n_{a'} - B^n_{a'}\overline{x}^{nf,(i)}_{a'|a'-1}) + \gamma\left(\hat{P}^{nf,(i)}_{a'|a'-1}\right)\left(\overline{x}^{f}_{a'|a'-1} - \overline{x}^{nf,(i)}_{a'|a'-1}\right) \tag{4.10}$$

where Kalman gain $M^n_{a'}$ is determined by $\left(B^n_{a'} + \left(\hat{P}^{nf,(i)}_{a'|a'-1}\right)^{-1}\right)^{-1}$ and consensus gain is represented by $\gamma\left(\hat{P}^{nf,(i)}_{a'|a'-1}\right)$. In the consensus term, $\overline{x}^{f}_{a'|a'-1}$ denotes the average state of all particles transmitted from R_f. In addition, the state and the

covariance of the *i*th particle are computed by choosing the upper part of the $\hat{x}_{a'|a'}^{nf,(i)}$ and $M_{a'}^{n}$. Based on $\hat{x}_{a'|a'}^{nf,(i)}$ and $\hat{x}_{a'|a'}^{fn,(i)}$, two acausal filters are generated, which are carried out from the second rendezvous point to the first rendezvous point. It also has map $M(z_{1:b+\eta}^{f}, x_{1:b+\eta}^{f})$ and $M(z_{1:b+\eta}^{n}, x_{1:b+\eta}^{n})$, which implies that the quality of the map around the first rendezvous point is relatively reliable. A reliability parameter η is defined to set a reliable range. After the update, these states are predicted according to the sampling process of the proposed RBPF-SLAM framework as shown in Section 2.1.1. Figure 4.3 shows the pose compensation process for MRSs. Each robot basically estimates its pose and map by operating relational RBPF-SLAM. This filter has better performance than the extended Kalman filter due to multi-hypothesis data association and time complexity. At the first rendezvous with the reference robot, each robot is initialized in the frame of the reference robot. A causal filter and an acausal filter are generated to estimate its past poses and maps as well as its current pose and map. If the second or more rendezvous occurs, the current poses of two robots are promptly updated via the process of the KCF. Subsequently, two acausal filters with early maps around the first rendezvous point are generated for the pose compensation. Finally, these acausal filters are terminated when the pose at the previous rendezvous point is updated. In addition, the acausal filter, which is generated at the first rendezvous, is terminated when the pose and the map estimations at the start point of the robot are finished.

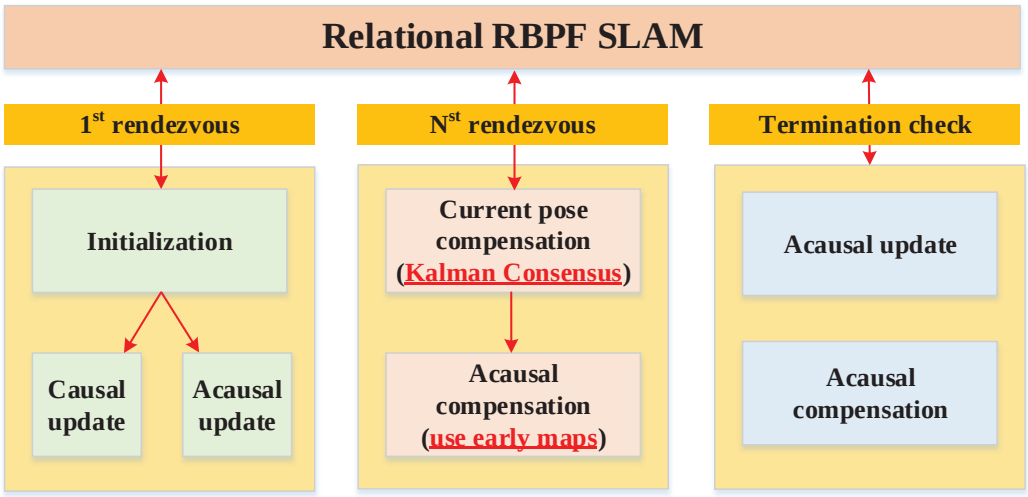


Figure 4.3 Compensation process in several rendezvous events

4.3 Feature Consensus at Feature Sharing Events

A reference robot R_f is looking at a feature m_t^i while an arbitrary robot R_n is also looking at the same feature m_t^i, which is called a feature sharing event. Unlike the case of rendezvous, the feature state is just updated without the prediction because we assume that all features registered in the SLAM framework are stationary. Unlike the conventional approach (2.28) – (2.29), the feature mean and covariance are computed by

$$\mu_{j,k}^{t,i} = \mu_{j,k-1}^{t,i} + K_{j,k}^{t,j}(z_{j,k}^{t,i} - \hat{z}_{j,k}^{t,i}) + \varepsilon \sum_{n \in N_{j,k-1}} \left(\mu_{j,k-1}^{n,\text{mean}} - \mu_{j,k-1}^{t,i}\right) \tag{4.11}$$

$$\Sigma_{j,k}^{t,i} = \Sigma_{j,k-1}^{t,i} - \Sigma_{j,k-1}^{t,i}\left(H_j^{t,i}\right)^T \left(H_j^{t,i}\,\Sigma_{j,k-1}^{t,i}\left(H_j^{t,i}\right)^T + R^i\right)^{-1} H_j^{t,i}\,\Sigma_{j,k-1}^{t,i} \tag{4.12}$$

$$K_{j,k}^{t,j} = \Sigma_{j,k-1}^{t,j}\left(H_j^{t,i}\right)^T \left(H_j^{t,i}\,\Sigma_{j,k-1}^{t,j}\left(H_j^{t,i}\right)^T + R_k^t\right)^{-1} \tag{4.13}$$

where $\mu_{j,k-1}^{n,\text{mean}}$ is the *j*th landmark of the robot *n* with the mean of all particles at *k*-1. Because the mean of particles for the *j*th landmark represents a true feature more accurately than other expressions, i.e. the feature mean of a particle with maximum weight and the feature mean with the weighted average of all particles for the *j*th landmark.

4.4 Overview of the Proposed RBPF-SLAM Framework

To perform accurate RBPF-SLAM, *particle to particle cooperation* and *robot to robot cooperation* are considered, which is described in Chapter 3 and Chapter 4, respectively. By adopting the concept of particle to particle cooperation, the particle depletion problem, the data association problem and filter divergence problem which are described in Chapter 2 and Chapter 3 are overcome. In addition, it leads to improvements of the importance weight step and the resampling step of the RBPF-SLAM framework. The weights of particles are compensated using clustering tools and the particle formation maintains the capability of multi-hypothesis data association, which is the strong property of RBPF-SLAM. The sampling step and the feature estimation step of the RBPF-SLAM framework can also be enhanced using inter-robot measurements in multi-robot systems. In these steps, the Kalman consensus information filter, which is robust than the covariance intersection, is employed for data fusion. The proposed RBPF-SLAM framework is shown in Fig. 4.3. All steps of RBPF-SLAM is affected by the proposed schemes for robot pose

consensus, feature consensus, weight compensation, and particle formation maintenance. The above mentioned process are described in Table 4.1.

TABLE 4.1

ALGORITHM FOR THE PROPOSED RBPF-SLAM FRAMEWORK

Algorithm 5. Proposed RBPF-SLAM Framework

Unscented FastSLAM $(z_t, u_t, X_{t-1}, Q_t, R_t)$
n = feature dimension; L = vehicle dimension
Set SUT parameters $(\alpha, \beta, \kappa, \lambda)$
Calculate SUT weights $(w_g^{[i]}, w_c^{[i]}, i = 0 \sim 2n(or 2L))$
$X_t = X_{aux} = \varnothing$
For all particles
  Retrieve X_{t-1}
  Predict mean and covariance of the vehicle (2.11) and (2.12)
  For all observations
    $\hat{k}$ =compatibility test $\left(z_t, \left\langle \mu_{k,t-1}^{[m]}, \Sigma_{k,t-1}^{[m]} \right\rangle\right)$
  end for
  For $\hat{k}$ = known feature
    Update mean and covariance of the vehicle (2.20) and (2.21)
    Refresh sigma points
    $w_t^{[m]}$ = calculate importance weight (2.22)
  end for
  Sample from updated posterior (2.13)
  $w_t^{'[m]}$ =weight compensation (Algorithm 2)
  Data acquisition from neighbors ($v_{a'}^n, V_{a'}^n \ \hat{x}_{a'|a'-1}^n$)
  If Rendezvous event = on
    Update the robot pose (4.10)
  end if
  If $\hat{k}$ = new feature
    Calculate new feature mean and covariance
  else
    Update mean and covariance of feature (2.20) and (2.21)
  end if
  For all shared features
    Update mean and covariance of feature (4.11) and (4.12)
  End for
  For unobserved features

$\mu_{k.t}^{[m]} = \mu_{k.t-1}^{[m]}, \sum_{k.t}^{[m]} = \sum_{k.t-1}^{[m]}$

end for

add $\left\langle x_t^{[m]}, N_t^{[m]}, \left\langle \mu_{k.t-1}^{[m]}, \sum_{k.t-1}^{[m]} \right\rangle \right\rangle$ to X_{aux}

end for

Resample from X_{aux} with probability $\propto w_t^{[m]}$

Formation Maintenance from X_{aux} with probability $w_t^{'[m]}$ (Algorithm 3)

Add new particles to X_t

Return X_t

Chapter 5

Simulations

This section shows the simulation results of the relational RBPF-SLAM which is explained in the above chapters. Since the purposes of the simulations are different one another, the simulation results are shown separately. To check the need for weight compensation, a simple simulation is performed by varying a parameter in Section 5.1. Section 5.2 shows the simulation results of FastSLAM [36], PSO based FastSLAM [46] and the proposed RBPF-SLAM in known data associations. The proposed RBPF-SLAM with unknown known data associations is performed and compared to FastSLAM and PSO based FastSLAM in the Section 5.3. To verify the SLAM performance by robot to robot cooperation, the proposed RBPF-SLAM is conducted in three simulations which are described in Section 5.4.

5.1 Verification for Needs of Weight Compensation and Particle Formation Maintenance

5.1.1 Simple Weight Compensation

To find out the need for weight compensation, weights of particles are simply compensated according to their regions as follows:

$$wc_t^{[m]} = \begin{cases} 0 & , \ x_t^{[m]} \notin S_{h\,\mathrm{var}} \ and \ x_t^{[m]} \notin S_{l\,\mathrm{var}} \\ \dfrac{(1-\eta)w_t^{[m]}}{\eta} & , \ x_t^{[m]} \in S_{l\,\mathrm{var}} \\ (\eta-1)w_t^{[m]} & , \ x_t^{[m]} \in S_{h\,\mathrm{var}} \end{cases} , \qquad (5.1)$$

where the term $wc_t^{[m]}$ is simply defined according to η, which is a parameter

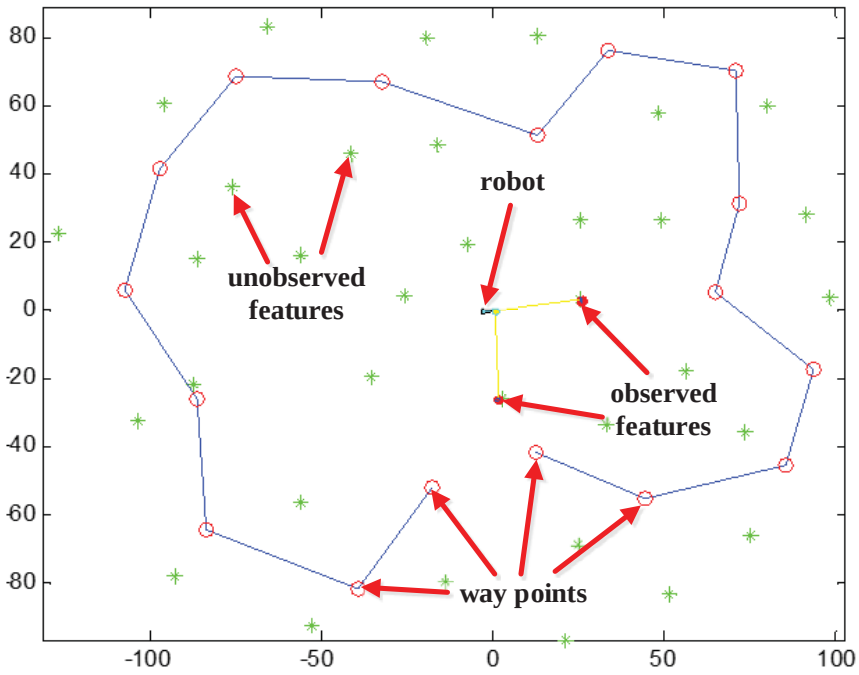


Figure 5.1 A simulator made by Tim Bailey [80]. Thirty-two features (*, green) and 17 way points (o, red) are illustrated in the simulator.

of the simple compensation scheme. It is changed from 1 to 100 in the simulation. If η is defined as one, the compensation does not occur, which is the conventional approach. As shown in Fig.5.1, the simulator that was made by Tim Bailey [80] is used. It has 32 point features and 17 way points, and the robot follows these way points by observing the point features. In addition, the robot is operated at a speed of 3m/s and the control noise regarding translation and rotation is described as follows:

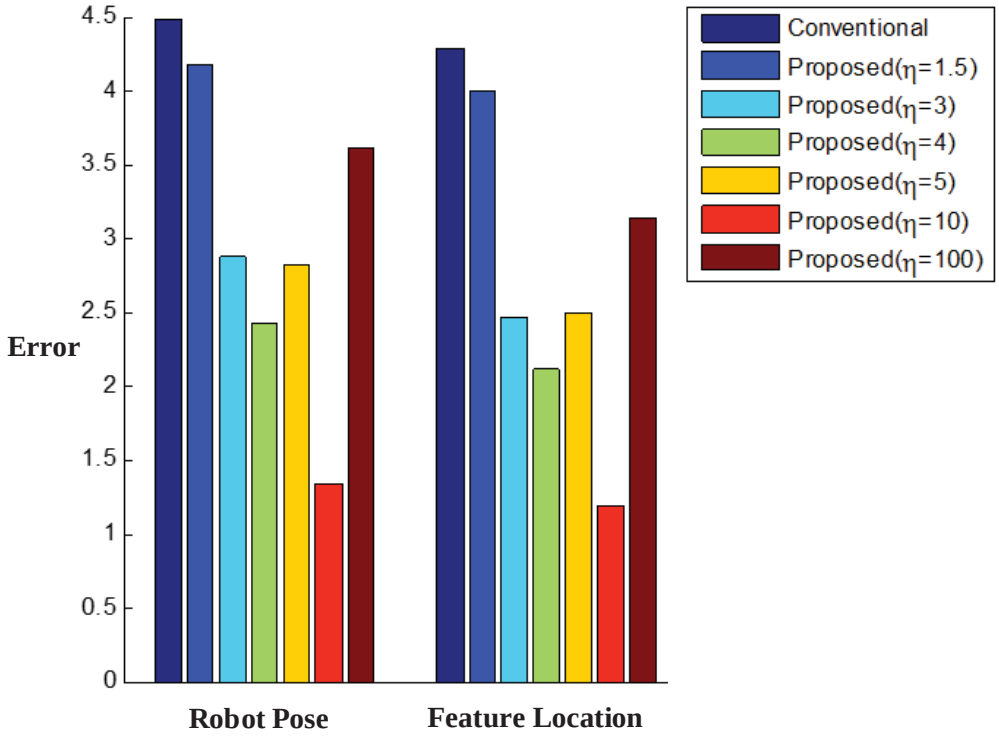


Figure 5.2 Robot pose errors and feature errors. The conventional approach implies that weights of particles are not compensated. The others are compensated according to the parameter.

$$Q = \begin{pmatrix} \sigma_V{}^2 & 0 \\ 0 & \sigma_G{}^2 \end{pmatrix} \tag{5.2}$$

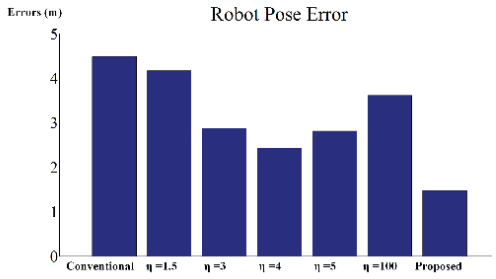
where $\sigma_V{}^2$ and $\sigma_G{}^2$ are the translation noise and the rotation noise, respectively.

For the consistency, the simulations are performed during the two laps in every trial while changing the parameter η. The results of robot pose errors and feature errors are shown in Fig.5.2. The conventional approach implies that weights of particles are not compensated and defined as $\eta = 1$. In case of $\eta = 10$, the errors of the robot pose and the feature location are highly reduced. But the errors in $\eta = 100$ are more increased than ones in $\eta = 10$, which means that the weights should be assigned by a proper value. Although the errors show different results in the several approaches, both robot pose errors and feature errors are more reduced than the conventional approach. It implies that weights of particles are incorrectly assigned over time from the data association problem described in Section 3.1 and should be compensated from the compensation scheme.

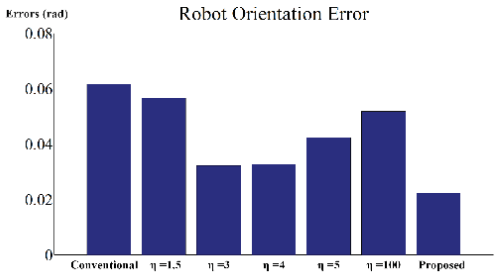
5.1.2 Piecewise Average based Weight Compensation

As simply described in the previous section, the data association problem cannot be evitable and the weight of each particle should be compensated. However, the approach using parameter η is very simple and quite heuristic. In addition, we cannot find out proper η because it changes according to the

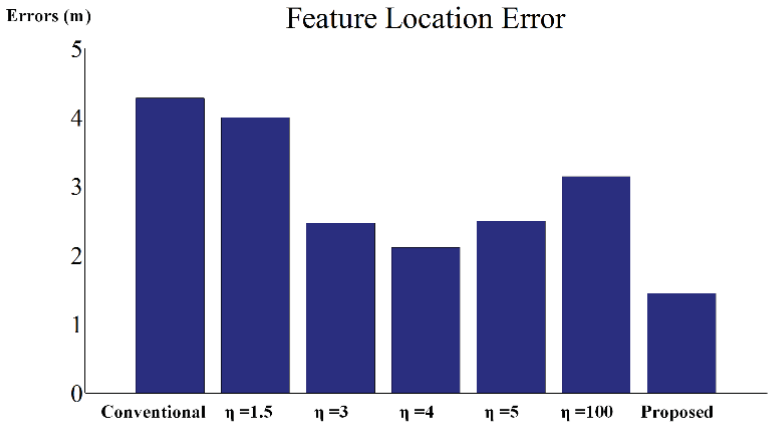
situations and the distribution of particles. To update the weight not heuristically but adaptively, piecewise average based weight compensation scheme that was described in Section 3 is performed in the simulation. When a weight of a particle is compensated, it takes an average weight between the particle and its neighbor of which weights are correctly assigned in advance.



(a)



(b)



(c)

Figure 5.3 The effect of the weight compensation scheme. The robot pose error, the robot orientation error, and the feature location error are represented in (a), (b), and (c), respectively. From the results, piecewise average based weight compensation scheme is more robust than other approaches.

As shown in Fig. 5.3., the result shows that the piecewise weight compensation scheme more correctly estimates the true proposal distribution than the conventional approach.

5.1.3 Particle Formation Maintenance Test

In this simulation, the same simulator conducted in the previous section is used. Several parameters related to control noise and measurement noise are the same as the previous ones. The simulation with known data association is done to check the effect of the weight compensation and to make a comparison

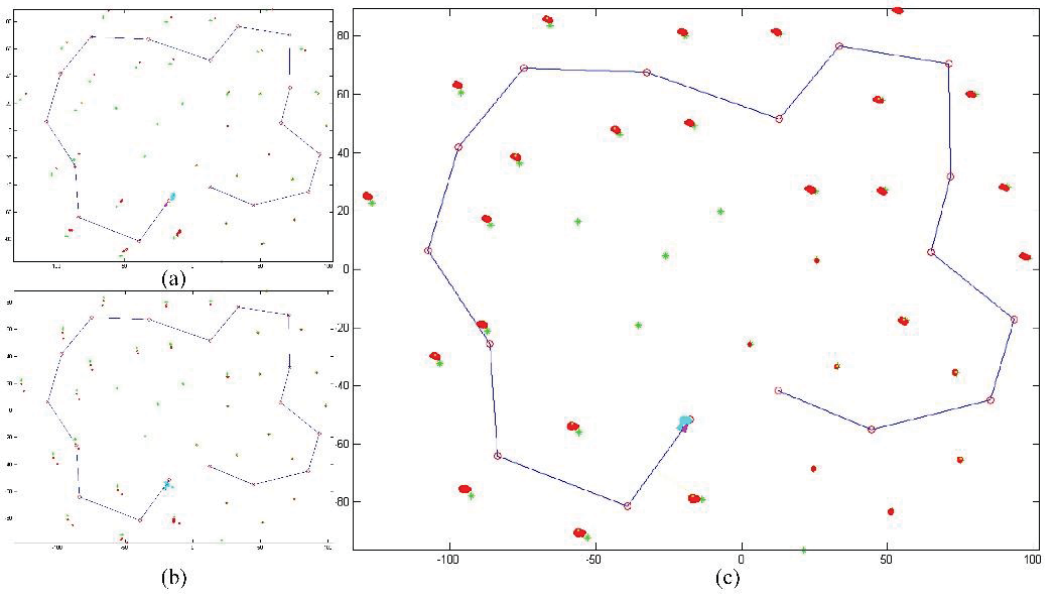


Figure 5.4 Graphical results of simulation with known data association. (a) and (b) show the estimated maps using RBPF-SLAM and RBPF-SLAM with weight compensation, respectively. The proposed RBPF-SLAM makes a scattered map due to PFM as shown in (c).

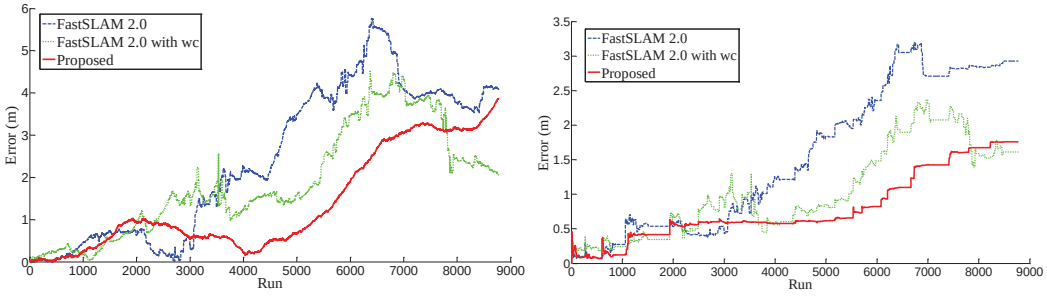
between the PFM and the resampling phase. To check the weight compensation and PFM effects, the proposed RBPF-SLAM is compared with RBPF-SLAM and RBPF-SLAM with the weight compensation, concurrently. One hundred particles are used for this simulation, which is the proper number of particles to evaluate the property of the weight compensation and PFM. Figure 5.4 denotes the final map involving features estimated by each particle. Unlike RBPF-SLAM and RBPF-SLAM with weight compensation, the proposed framework estimates features from more scattered particles due to PFM. PFM makes the filter block the filter divergence and consistently maintains multiple data association hypotheses at each time step. Figure 5.5(a) graphically shows the variation of robot pose errors over time. Here, the errors is computed by

$$\varepsilon_{Pose}(t) = \sqrt{\left(\left(\sum_i x^{t,[i]}\right)/N - x_{true}\right)^T \left(\left(\sum_i x^{t,[i]}\right)/N - x_{true}\right)} \qquad (5.3)$$

where the pose error $\varepsilon_{Pose}(t)$ is calculated according to the difference between the average pose of all particles and the true pose of robot. The average pose of all particles is the best way to describe the true pose of the robot [94]. The error of the feature location $\varepsilon_{f\_Pose}(t)$ is also calculated by

$$\varepsilon_{Feature}(t) = \frac{\sum_i \sqrt{\left(\left(\sum_m \mu_{i,t}^{[m]}\right)/N - \mu_{i,true}\right)^T \left(\left(\sum_m \mu_{i,t}^{[m]}\right)/N - \mu_{i,true}\right)}}{N_{f,t}} \qquad (5.4)$$

where $N_{f,t}$ is the number of observed features at time *t*. The term $\left(\sum_m \mu_{i,t}^{[m]}\right) / N$ is the average information of the *i*th feature and can be simply represented due to known data association. The variation of feature location errors over time is shown in Fig. 5.5(b). The graphical results show that the error of RBPF-SLAM with weight compensation is mostly lower than that of RBPF-SLAM, except for a small portion at the beginning. This implies that the originally calculated weights of the particles are not correctly assigned and they are compensated correctly. In the case of the proposed RBPF-SLAM, the robot pose errors and the feature location errors are similar to ones of other methods at the beginning and at the end of journey, but other parts are more correctly estimated than others. In short, the proposed approach is comprehensively superior to the other methods. Table 5.1 quantitatively represents the root mean square (RMS) pose errors of the robot and the features obtained from



(a) (b)

Figure 5.5 SLAM error comparison. It shows the errors of the robot poses and features over time. The proposed framework (red solid line) has fewer errors than the others.

TABLE 5.1

TOTAL COMPARISON OF RMS POSITION

	RBPF-SLAM	RBPF-SLAM with weight compensation	Proposed
RMS robot position error (m)	2.4091	1.8657	1.3762
RMS feature position error (m)	1.5350	1.0352	0.8017

$\sum_{t=1}^{T} \varepsilon_{Pose}(t)/T$ and $\sum_{t=1}^{T} \varepsilon_{f\_Pose}(t)/T$ for the three methods. Among them, the proposed RBPF-SLAM greatly reduces the robot pose errors as well as the errors of the feature location using the weight compensation and the PFM technique.

5.2 Simulation with Unknown Data Association

In the previous section, we confirm the performance of the proposed RBPF-SLAM. Here, the proposed RBPF-SLAM is compared to two methods, i.e. FastSLAM [36] and PSO-based FastSLAM [46] on the simulator with an unknown data association as shown in Fig.5.6. This simulator is based on the simulator used in Section 5.1 and modified for different map test. Fifty features and particles are employed for the simulation. The robot takes a round to the simulator by following the wall. For data association, we adopt the maximum likelihood assignment procedure on a per-particle basis. Each particle selects the features maximizing the likelihood, given by

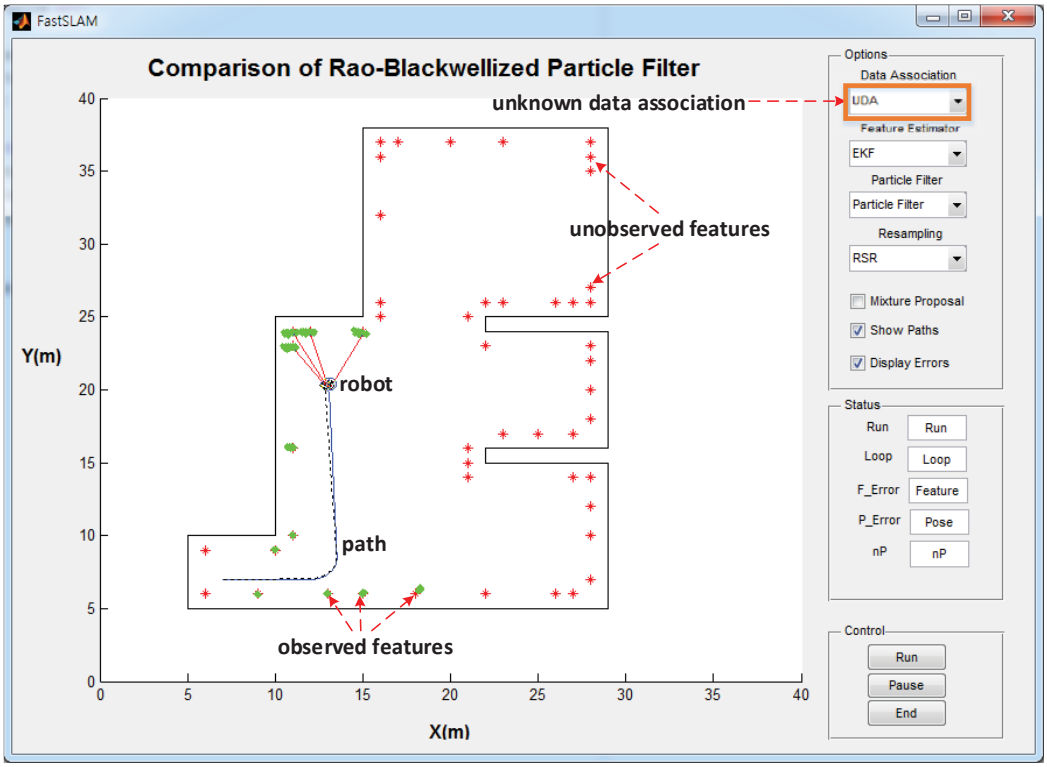


Figure 5.6 Simulator with an unknown data association. Fifty features and particles are used for the simulation.

$$c_t^{[y]} = \arg\max_{c^t} p(z_t \mid x_t^{[y]}, m_{c^t}, c_t) \tag{5.5}$$

where $c_t^{[y]}$ is the correspondence regarding the *y*th particle.

Figure 5.7 shows the errors of the estimated robot pose and feature location. They are computed by the difference between true value and the estimated value as represented in (5.3) and (5.4). In case of FastSLAM and PSO based FastsLAM, the errors are highly oscillated due to the resampling scheme which causes the brutal rejection and replication of particles. This problem is a lack of multiple hypotheses, called the particle depletion problem. However, the proposed RBPF-SLAM keeps the errors down in the presence of an unknown

TABLE 5.2

COMPUTATIONAL TIME PER ONE STEP (SEC)

	UFastSLAM	PSO-UFastSLAM	Proposed
Simulation	0.1482	0.1554	0.2319

data association while maintaining multiple hypotheses. Here, PFM gives a positive effect to the proposed RBPF-SLAM using the geometric information between particles. In particular, the orientation is greatly reduced in the proposed RBPF-SLAM. The reduction of orientation error implies that several particles involved in their triangular structure estimates the true robot orientation concurrently and adaptively, which results in more correct estimation for the environment. The computational time per one step is described in Table 5.2. The gap in the execution time between FastSLAM and the proposed RBPF-SLAM, caused by the weight compensation and PFM, is less than 0.1 sec. It means that the effect of computational complexity in real operation is negligible.

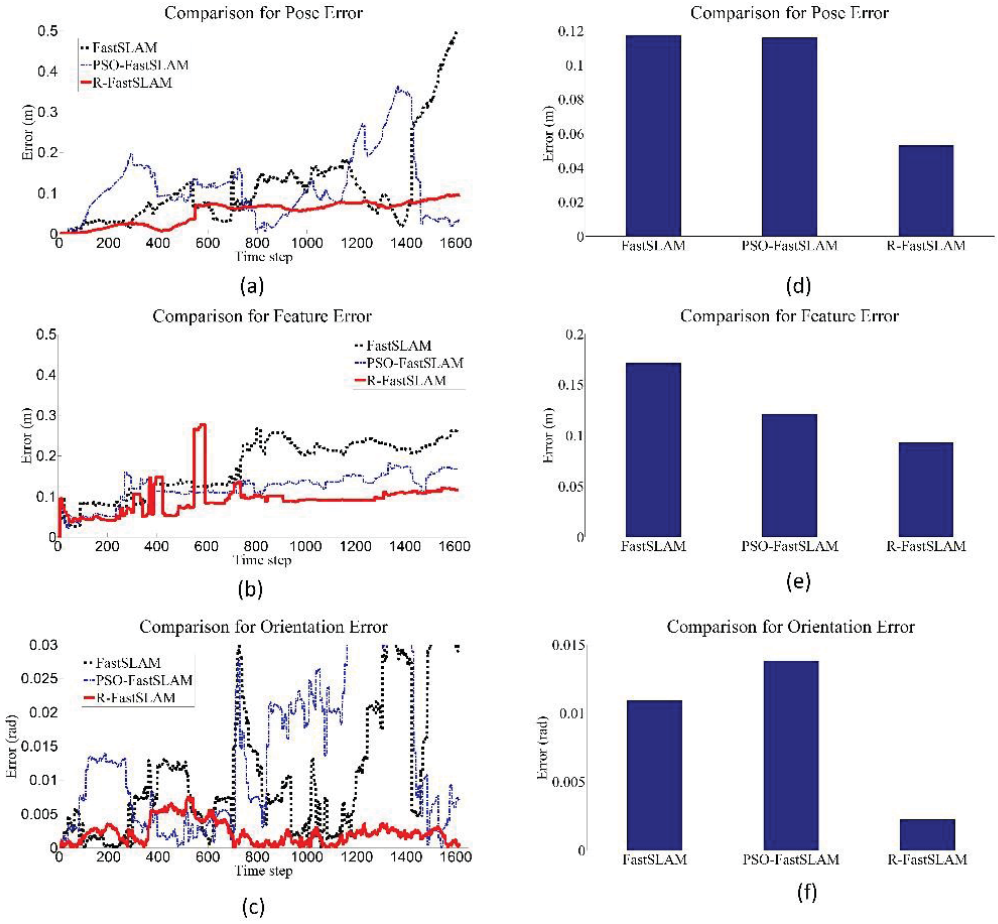


Figure 5.7 Comparison of SLAM errors. The errors of the robot pose, orientation and feature location are shown over time in (a), (b) and (c), respectively. FastSLAM, PSO-FastSLAM, the proposed RBPF-SLAM are denoted by the dashed line, the dashed-dotted line and the solid line, respectively. (d), (e) and (f) represent errors as bar graph by accumulating the errors over time.

5.3 Simulations for Robot Pose Consensus and Feature Consensus

To evaluate the performance of robot to robot cooperation, specifically for robot pose consensus, we also extend and refine the simulator made by Time

Bailey [80]. In two different simulations, robots localize their poses and build maps by assuming the unknown initial condition. These robots move at a maximum speed of 3m/s. The period of the update for the control input is 0.5s. The period of the update for observation is 1.6s. In addition, they have non-holonomic constraints (e.g. maximum steering angle: 30 degree and maximum rate of change in steer angle: 20 degree). Their control noise Q and observation noise R are defined as follows:

$$Q = \begin{bmatrix} \sigma_V & 0 \\ 0 & \sigma_G \end{bmatrix}, \quad R = \begin{bmatrix} \sigma_R & 0 \\ 0 & \sigma_B \end{bmatrix}, \tag{5.6}$$

where σ_V, σ_G, σ_R and σ_B are 0.33, 3*rad*, 0.1, 1*rad*, respectively. Ten particles are used to operate RBPF-SLAM.

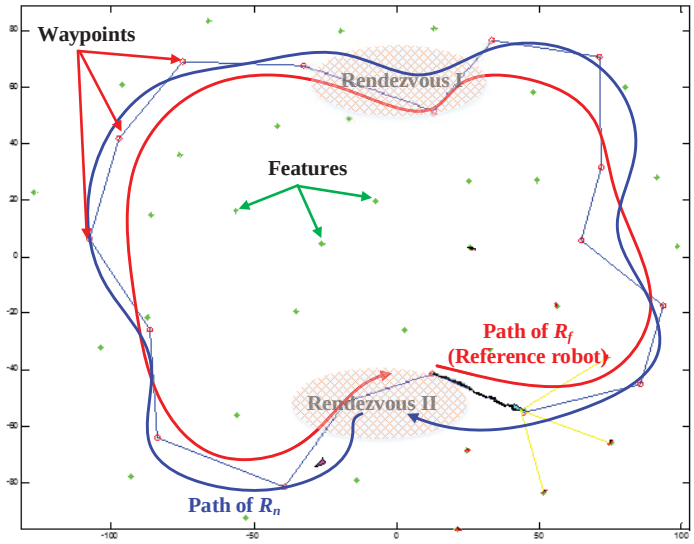


Figure 5.8 The environment of the simulation I. The path of R_n is represented as the blue line. The path of R_f is represented as the red line. They move in opposition directions. There are 16 way points and 35 features.

5.3.1 Robot Pose Consensus Test I

In this simulation, there are 16 way points and 35 features as shown in Fig. 5.8. A reference robot R_f and an arbitrary robot R_n move in opposition directions. During their journey, they meet two times. Until the first rendezvous point, the poses and maps of R_f are only estimated. The control input and the observations of R_n are just accumulated. After the first rendezvous, the past poses and maps of R_n are estimated through an acausal filter and its current pose and map is updated by a causal filter. The acausal filter is operated from the first rendezvous point to the start point of R_n. In this simulation, the time step t at the first rendezvous is about 200. A boundary constant of reliability η is

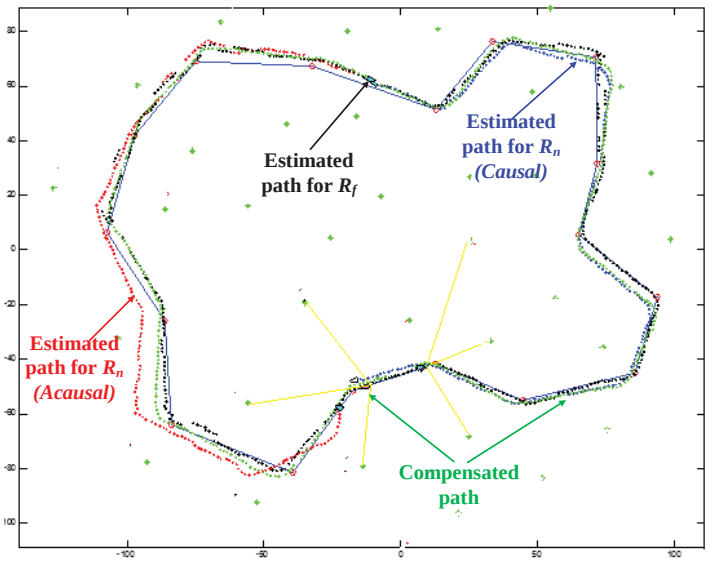


Figure 5.9 The result of the simulation I. The estimated acausal and causal paths for R_n are represented as red and blue. The estimated path for R_f is represented as black. The compensated paths of R_n and R_f are represented as the green lines.

defined as 50, which implies that the map updated from $t = 200$ to $t = 250$ is used for the compensation after the second rendezvous. When the second rendezvous occurs, the current poses of two robots are updated using the KCF. The accuracy for the poses is described in Table 5.3. The errors of both poses are reduced using the KCF. In addition, the covariance of both robots is decreased, which implies that the robot poses can be more correctly estimated. Subsequently, the poses and the map of each robot are also updated based on the compensated current pose. It is conducted between the first rendezvous point and the second rendezvous point. Figure 5.9 shows the final result of the simulation. The estimated path for R_f is represented as black. The estimated path for R_n is divided by a path obtained from the causal filter and a path obtained from the acausal filter. The compensated paths are described as green lines. As shown in this figure, the path of each robot estimated between the first rendezvous and the second rendezvous is sophisticatedly compensated. Total errors for the robot poses and features are computed by

$$\varepsilon_{Pose} = \frac{\sum_t \sqrt{\left(\left(\sum_i x^{t,[i]}\right)/N - x_{true}(t)\right)^T \left(\left(\sum_i x^{t,[i]}\right)/N - x_{true}(t)\right)}}{total\ runtime}, \tag{5.7}$$

$$\varepsilon_{Features} = \frac{\sum_t \sum_i \left(\sum_j^{N_f} \sqrt{\left(m_{j,t}^{[i]} - m_{j,true}(t)\right)^T \left(m_{j,t}^{[i]} - m_{j,true}(t)\right)}\right)}{N_f \cdot N \cdot total\ runtime}, \tag{5.8}$$

where N is the number of particles, $x_t^{[i]}$ is the ith particle at t, $x_{true}(t)$ is the true vehicle pose at t, The jth feature of the ith particle is defined as $m_{j,t}^{[i]}$, and N_f is the number of features in the map. The robot pose and feature errors of the conventional approach and the proposed approach are described in Table 5.4, respectively. The errors are correctly compensated in the proposed approach. If the constant parameter of the consensus gain CI, ε is defined more properly, the errors can be more reduced.

TABLE 5.3

COMPARISON OF POSE ERRORS AT RENDEZVOUS

	No fusion	KCF fusion
For R_n	1.7634	0.7936
For R_f	0.4461	0.3871

TABLE 5.4

COMPARISON OF TOTAL ERRORS

	Conventional		Proposed	
	Pose	Feature	Pose	Feature
For R_n	3.8912	2.5185	1.9126	1.8414
For R_f	2.8543	2.7269	1.7323	2.5085

5.3.2 Robot Pose Consensus Test II

As shown in Fig.5.10, two robots have different trajectories along their waypoints shaped like *M*. They meet two times that are described as Rendezvous I and Rendezvous II in the figure. In addition, they only communicate at two rendezvous points. Likewise with the previous simulation, the initialization of R_n is conducted at the first rendezvous with R_f. It occurs when the time is about 150. A boundary constant of reliability η is defined as 20, which implies that the map updated from $t=150$ to $t=170$ is used during the compensation after the second rendezvous. The compensation and the fusion of information occur at the second rendezvous. The errors of the

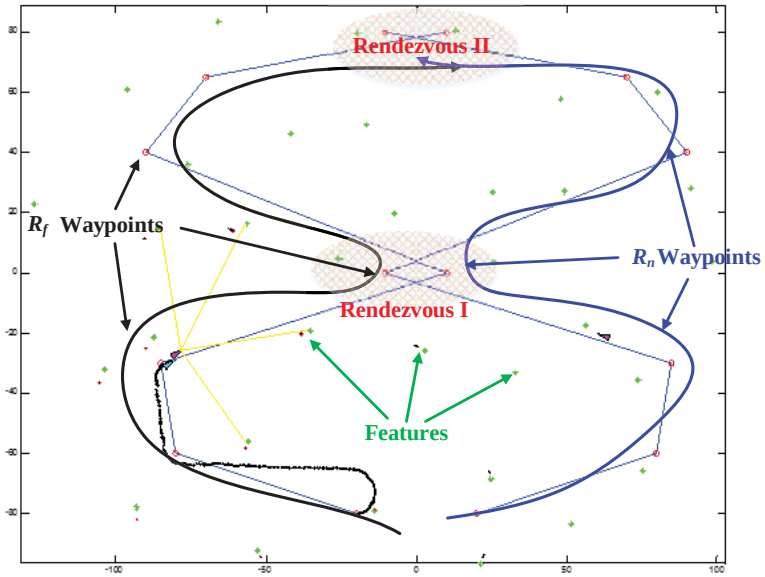


Figure 5.10 The environment of the simulation II. The path of R_n is represented as the blue line. The path of R_f is represented as the red line. There are seven way points for each robot.

fused pose for both R_f and R_n are described in Table 5.5. In the case of the proposed approach, the errors are remarkably reduced due to the KCIF, which affects subsequent compensation. The robot pose and feature errors of the conventional approach and the proposed approach are described in Table 5.6. Based on the compensation of the current pose for both R_n and R_f, the errors for the robot poses and features are more correctly compensated in the proposed approach. The graphical result of the simulation II is represented in Fig. 5.11. The path of R_f (black) is incrementally estimated since it starts. When the first rendezvous between R_f and R_n occurs, the paths of R_n are estimated by the causal filter (blue) and the acausal filter (red). In addition, its path and map are represented in the frame of R_f. After the second rendezvous, the paths and maps of R_f and Rn are compensated by the KCIF and backtracking. As shown in the figure, their paths and maps are compensated more accurately in the proposed framework.

TABLE 5.5

COMPARISON OF POSE ERRORS AT THE RENDEZVOUS EVENT

	No filter	KCIF fusion
For R_n	5.6575	1.7399
For R_f	2.1133	1.7678

TABLE 5.6

COMPARISON OF TOTAL ERRORS

	Conventional		Proposed	
	Pose	Feature	Pose	Feature
For R_n	1.4752	1.1585	1.1926	1.0441
For R_f	1.3544	1.2296	1.3233	1.0885

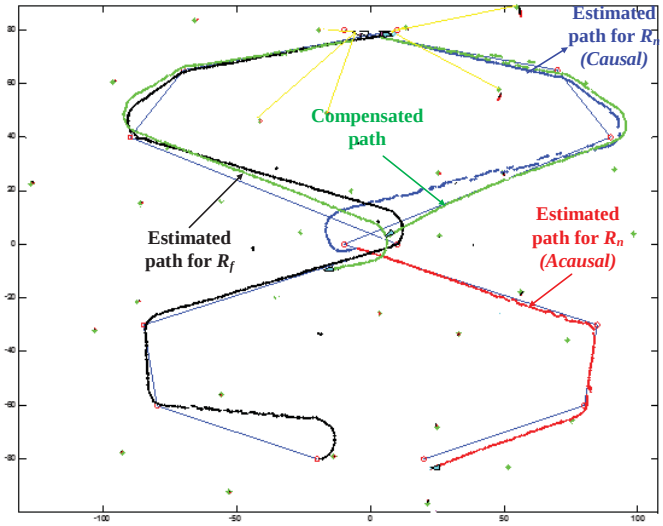
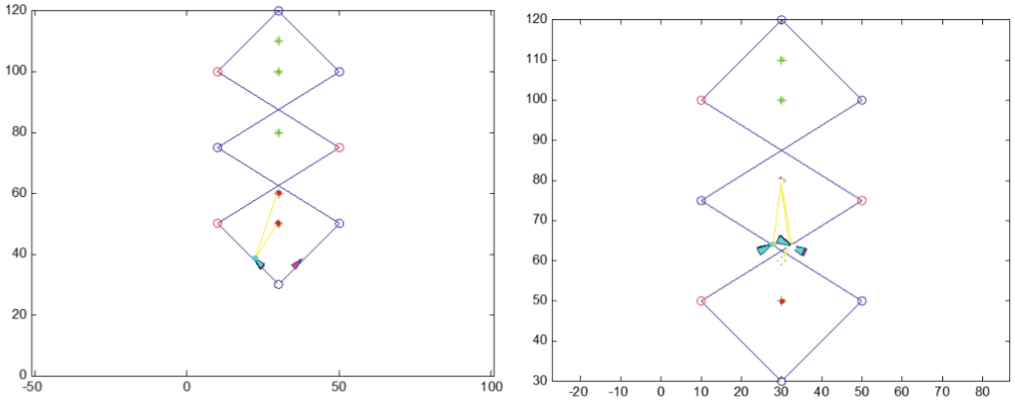


Figure 5.11 The result of the simulation II. The estimated acausal and causal paths for R_n are represented as red and blue. The estimated path for R_f is represented as black. The compensated paths of R_n and R_f are represented as the green line. The compensation is operated between the first rendezvous and the second rendezvous.

5.3.3 Feature Consensus Test

A simulation for the feature sharing event is conducted in this section. As shown in Fig. 5.12, two robots and five features are used in this simulation. The robots move forward in a zig-zag pattern. At first, a reference robot R_f estimates

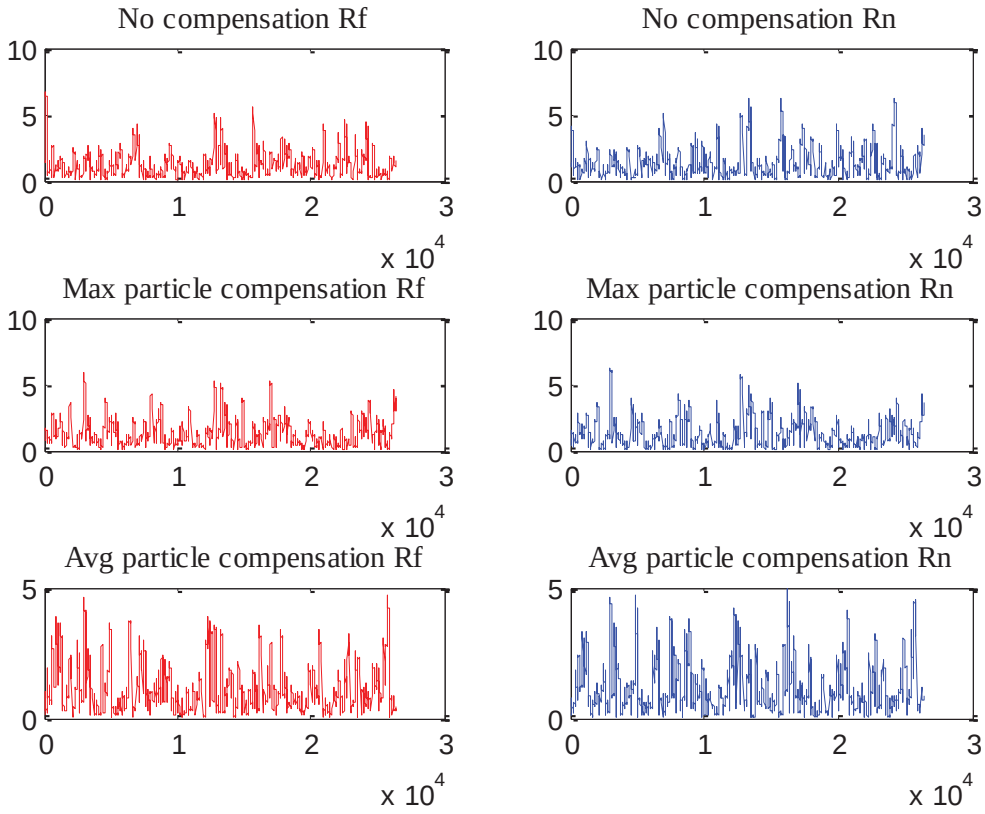


(a) In the Beginning (b) After the Initialization

Figure 5.12 Simulation for the feature sharing event. Two robots and five features are used in the simulation. After the initialization of R_n, they are sharing common features and reduce the errors of the estimated features by adopting the Kalman consensus scheme in the proposed RBPF-SLAM.

its poses and a map of surroundings. Another robot R_n just collects its control and sensor signals. After a rendezvous between R_n and R_f, R_n also estimates its poses and map using the collected control and sensor signals. At the same time, they are sharing a common feature which is located in (30, 80). The Kalman consensus scheme as mentioned in the previous Chapter is operated to fuse the state of this feature. Here, the proposed consensus scheme is compared with no compensation approach and [92]. To measure the errors of each approaches, the simulation is performed one hundred times. Fig. 5.13 shows the average feature errors over time for R_n and R_f, which are computed by

$$\varepsilon_{Avg}(t) = \frac{\sum_{K}\sum_{i}\left(\sum_{j}^{N_f}\sqrt{\left(m_{j,t}^{[i]} - m_{j,t}^{true}\right)^{T}\left(m_{j,t}^{[i]} - m_{j,t}^{true}\right)}\right)}{N_f \cdot N \cdot K} \tag{5.9}$$



(a) Feature errors of R_f (b) Feature errors of R_n

Figure 5.13 Simulation result for the feature sharing event. The feature errors of both R_f and R_n are reduced using the average feature mean of particles in the proposed approach.

where K is the total iteration number, which is defined as 100. The total average errors are computed by dividing $\varepsilon_{Avg}(t)$ by total time, which is described in Table 5.7. In case of KCIF fusion using a particle with maximum weight, the feature errors are reduced for R_f but increase for R_n. However, the feature errors of both R_n and R_f are reduced by transmitting the average feature mean of particles for KCIF fusion.

TABLE 5.7

COMPARISON OF TOTAL FEATURE ERRORS

	No filter	KCIF fusion with Maximum Particle	KCIF fusion with Average of Particles
For R_n	1.3333	1.3459	1.2089
For R_f	1.3439	1.3032	1.2556

5.4 Discussions

In the Section 5, several simulations are performed using the formal simulator made by Tim Bailey. Before the performance test, the verification for needs of weight compensation (WC) and particle formation maintenance (PFM) was carried out. In the weight compensation, two kinds of approach were tried, i.e. a heuristic weight compensation and piecewise average based weight compensation. The error results showed that RBPF-SLAM using the weight compensation schemes more correctly estimates the robot poses and feature locations than the conventional RBPF-SLAM. In terms of WC, the piecewise average based weight compensation scheme showed better performance, which was described in Section 5.1.1 and Section 5.1.2. Because the WC techniques have been proposed to assign more correct weights to particle even in failure of data association as mentioned in Section 3, the simulation results denote the weight of each particle is more correctly assigned using WC. Table 5.8 summarizes the results of all simulations. In Section 5.1.3, PFM was also

verified by comparing RBPF-SLAM and RBPF-SLAM with WC. The result showed that the robot pose and feature location error rates are decreased to 52 percent and 57 percent, respectively, in the best case. Because PFM has been proposed to overcome the particle depletion problem, the results imply the problem is highly curbed throughout the simulation. When data association is unknown, each particle determines the feature using the maximum likelihood method as shown in Section 5.2. FastSLAM[36], PSO based FastSLAM[46] and the proposed were compared using 50 particles. In terms of robot poses and feature locations, the both errors are highly reduced in the proposed approach. For multi-robot SLAM, the performance verification using KCIF were conducted in case of rendezvous and feature sharing events. In Section 5.3.1 and 5.3.2, robot poses are fused using the consensus scheme when two robots meet each other. The robot poses at the rendezvous point where more accurately estimated, which can be seen from the error results. It implies SLAM performance is highly enhanced using rendezvous events in the multi robot system. Especially, the sampling step for robot pose estimation is improved. In addition, features were more correctly updated when two robots share common features. The errors are reduced using the feature consensus scheme ranging from 4 percent to 10 percent. It leads to considerable enhancement of the measurement update part in the proposed SLAM framework.

TABLE 5.8

RESULTS OF SIMULATIONS

Simulations	Pose Error Reduction Rate		Feature Error Reduction Rate		Location
Simulation I	[36]	[36] + WC	[36]	[36] + WC	Section 5.1.3
	51%	25%	57%	43%	
Simulation II	[36]	[46]	[36]	[46]	Section 5.2
	52%	51%	43%	20%	
Simulation III	[89]		[89]		Section 5.3.1
	51%		27%		
Simulation VI	[89]		[89]		Section 5.3.2
	20%		10%		
Simulation V	[92] (For R_n)		[92] (For R_f)		Section 5.3.3
	10%		4%		

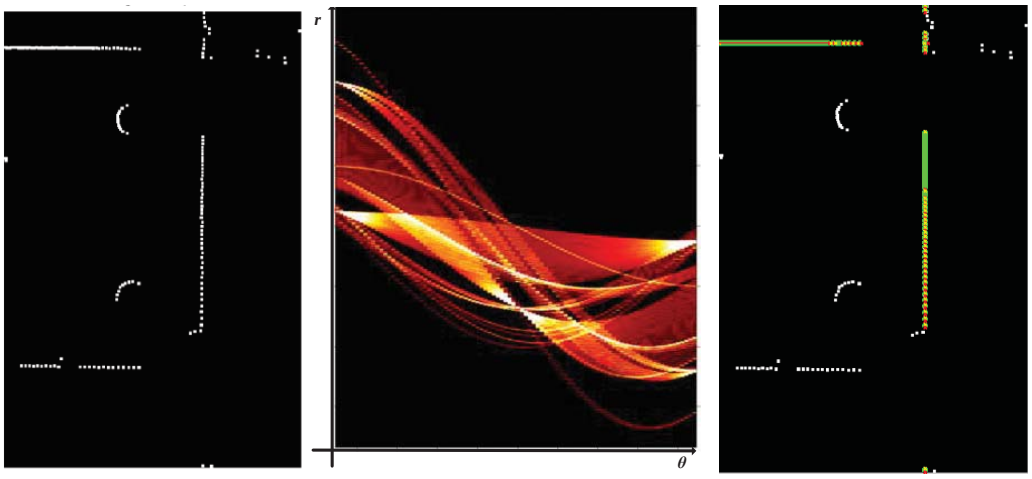
Chapter 6

Experiment

In this chapter, the SLAM performance of proposed framework is verified from experimental results. Line features that represent a map of surroundings of robots are extracted using the Hough transform in Section 6.1. The proposed approach was also tested in formal datasets, i.e. Car park dataset and Victoria park dataset, in Section 6.2 and Section 6.3. The estimated robot pose results from FastSLAM 2.0, PSO based FastSLAM, UFastSLAM, particle fission based UFastSLAM, PSO based UFastSLAM and the proposed RBPF-SLAM framework are compared with the GPS data while varying the number of particles. The robot poses obtained from the proposed RBPF-SLAM is remarkably consistent with GPS. In Section 6.4 and Section 6.5, the proposed SLAM approach was verified in the large indoor and outdoor environments.

6.1 Line Feature Extraction

In this dissertation, line features are extracted from the Hough transform, which is one of feature extraction techniques used in image analysis, computer vision, and digital image processing. Duda and Hart [82] proposed the use of a different pair of parameters, denoted r and θ, for the lines in the Hough transform. These two values, taken in conjunction, define a polar coordinate. The parameter r represents the algebraic distance between the line and the origin, while θ is the angle of the vector orthogonal to the line and pointing toward the half upper plane. If the line is located above the origin, θ is simply the angle of the vector from the origin to this closest point. Using this Hough transform, line features can be extracted when measurements are obtained from



(a) Laser scan data (b) Hough transform (c) Extracted features

Figure 6.1 Line feature extraction. Laser scan data is represented in (a). The data is transformed in a polar coordinate as shown in (b). After the Hough transform, several features are extracted and (c) shows the results.

the laser range finder, which is shown in Fig.6.1. In this dissertation, a line feature vector consists of $\mathbf{I} = \{\rho, \theta, l, (x_s, y_s), (x_t, y_t)\}$, which is depicted in Fig. 6.2 [83]. To deal with this line feature, the measurement model is the projection function that maps the line features in the state vector from the world to the robot centered coordinate, described by:

$$\begin{bmatrix} z_\rho \\ z_\theta \end{bmatrix} = \begin{bmatrix} \rho_r^k \\ \theta_r^k \end{bmatrix} = \begin{bmatrix} \sqrt{dx^2 + dy^2} \\ \tan^{-1}(dy/dx) - r_\theta \end{bmatrix}, \tag{6.1}$$

$$dx = (-r_x \cos\theta^k - r_y \sin\theta^k + \rho^k)\cos\theta^k, \tag{6.2}$$

$$dy = (-r_x \cos\theta^k - r_y \sin\theta^k + \rho^k)\sin\theta^k, \tag{6.3}$$

where $\begin{bmatrix} z_\rho & z_\theta \end{bmatrix}^T$ denotes a projected location of $\mathbf{I}^k$ in the robot centered coordinate and $\tan^{-1}(dy/dx) - r_\theta$ returns the angle from the origin in the local

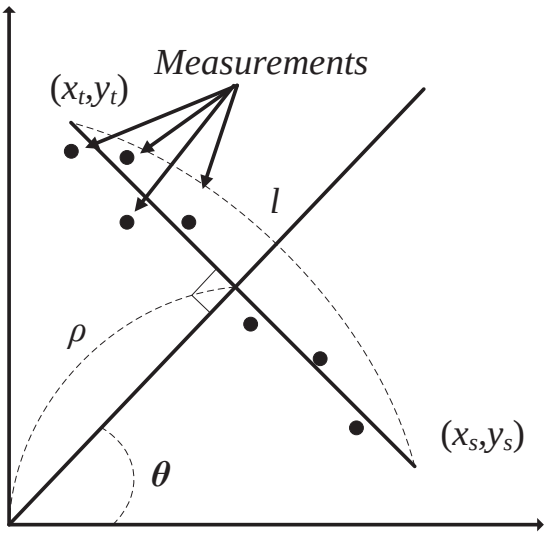


Figure 6.2 Parameters of a line feature vector

coordinate to a point (dx, dy).

6.2 Tests using Car Park Dataset

We also verify the proposed approach using the Car Park dataset, which is one of well-known datasets that has been used in [87]. The car was equipped with a horizontal scanning laser sensor with 80 meters observing radius and 180 degrees field of view. Figure 6.3 shows the vehicle kinematic information. All vehicle kinematic variables, i.e. a, b, H, and L are defined by a = 3.78 m, b = 0.5 m, H=0.76 m, L=2.83 m. The dead-reckoning noise, laser range and bearing noise and GPS noises are defined as follows:

$$\sigma_U = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\varphi^2 \end{bmatrix}, \ \sigma_L = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}, \ \sigma_G = \begin{bmatrix} \sigma_{gx}^2 & 0 \\ 0 & \sigma_{gy}^2 \end{bmatrix} \qquad (6.4)$$

where σ_v, σ_φ, σ_r, σ_θ, σ_{gx} and σ_{gy} are noise parameters which are defined as 0.7m, 7 rad, 0.10m, 1 rad, 0.05m, and 0.05m, respectively. The

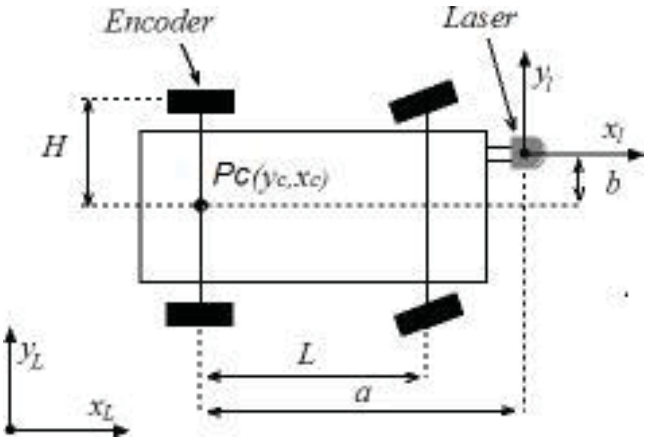


Figure 6.3 Vehicle Kinematic Information

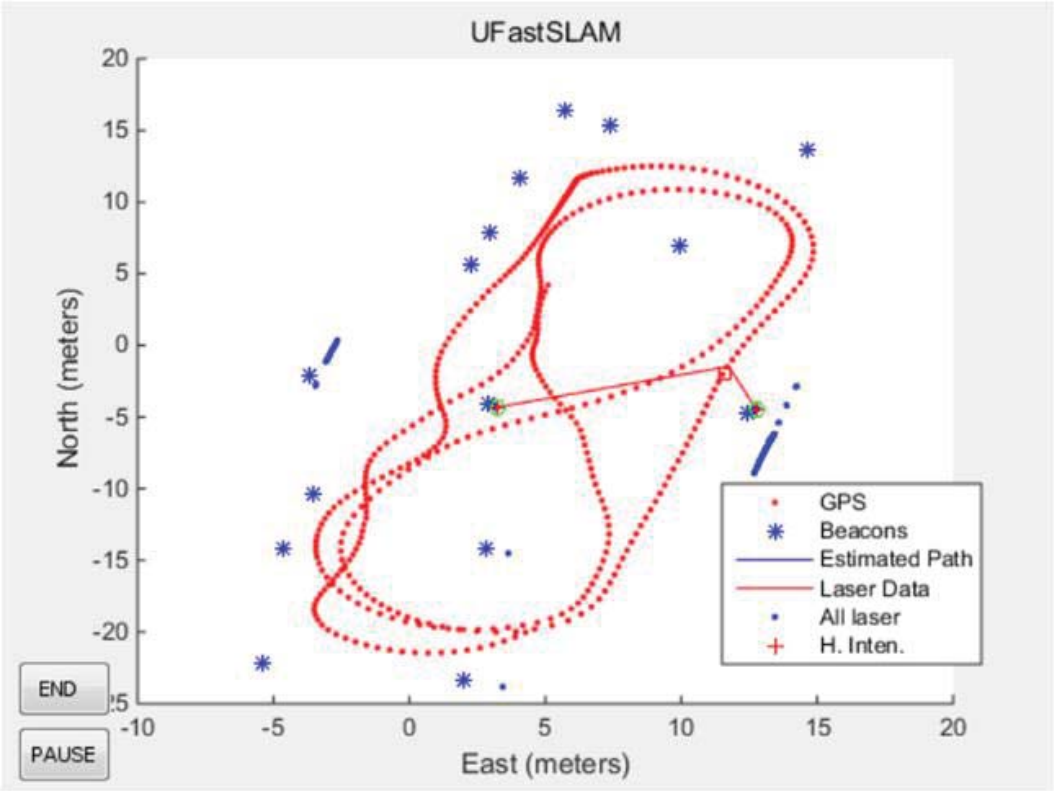


Figure 6.4 GUI for algorithm tests in car park dataset.

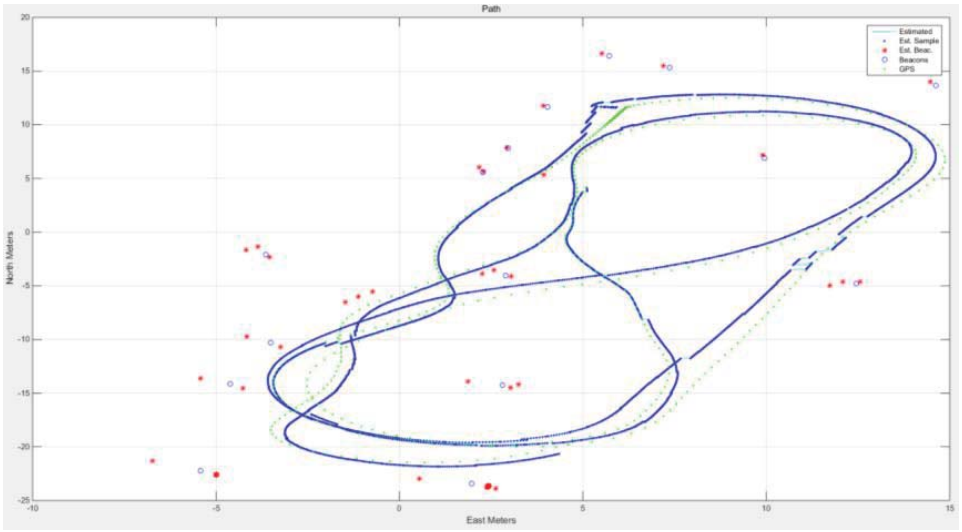
dataset is operated using the defined variables in a simulator, which is represented in the Fig. 6.4. GPS data, the true location of beacons, the estimated path, and laser data are illustrated in this figure. Especially 15 beacons are used for features which are obtained by laser data. The size of the environment is 20m x 20m. In the test, a variety in the number of the particles was chosen, with minimum 3 particles and maximum 50 particles. The test was performed iteratively 50 times. Three methods, i.e. UFastSLAM, PSO based UFastSLAM and the proposed approach were compared in terms of errors of the robot pose and the map. Fig. 6.5 and Fig. 6.6 show the results of the estimated map and the estimated robot poses obtained from the methods. The robot pose x_t is

TABLE 6.1

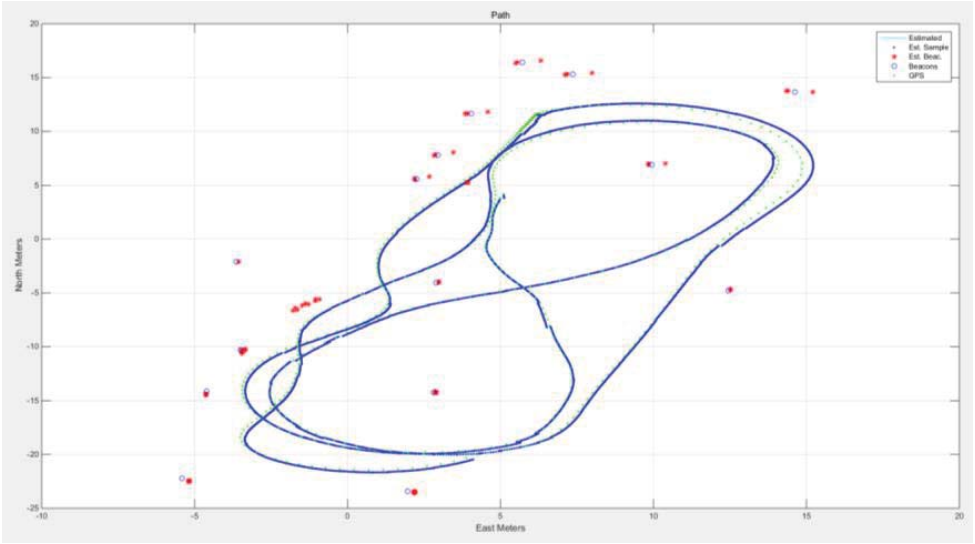
COMPARISON OF AVERAGE POSE ERROR AND STANDARD DEVIATION OF POSE ERRORS

Errors	[60]	[40]	[62]	Proposed
Avg of Pose errors for 3particles	0.7186	0.6779	0.6015	0.5402
Std of Pose errors for 3particles	0.2993	0.3803	0.4808	0.1711
Avg of Pose errors for 10particles	0.4246	0.6661	0.3323	0.3521
Avg of Pose errors for 10particles	0.4246	0.6661	0.3323	0.3521
Std of Pose errors for 10particles	0.3522	0.3622	0.2222	0.0808
Avg of Pose errors for 50particles	0.2301	0.3287	0.2664	0.2288
Std of Pose errors for 50particles	0.0458	0.1232	0.0772	0.0587

represented by the average pose of all particles and the map M_t is represented using the estimated poses from all particles. UFastSLAM and PSO based UFastSLAM properly estimates the map and the robot poses but some points cannot be estimated well. These phenomenon can be affected by the particle depletion problem which is removed in the proposed approach. Unlike the conventional methods, the locations of the estimated beacons are scattered in the proposed approach. It implies that all particles constitute the triangular formation by maintaining the number of distinctive particles.



(a) UFastSLAM



(b) PSO based UFastSLAM

Figure 6.5 Estimated beacons (red colored points) and robot poses (blue colored lines)

The robot pose errors of the three methods are also compared using the GPS data x_t^g. Although GPS data does not mean the true robot poses, it is the best to use it in this experiment. For the comparison, the average pose error e_{avg} and its standard deviation S_{error} are computed as follows:

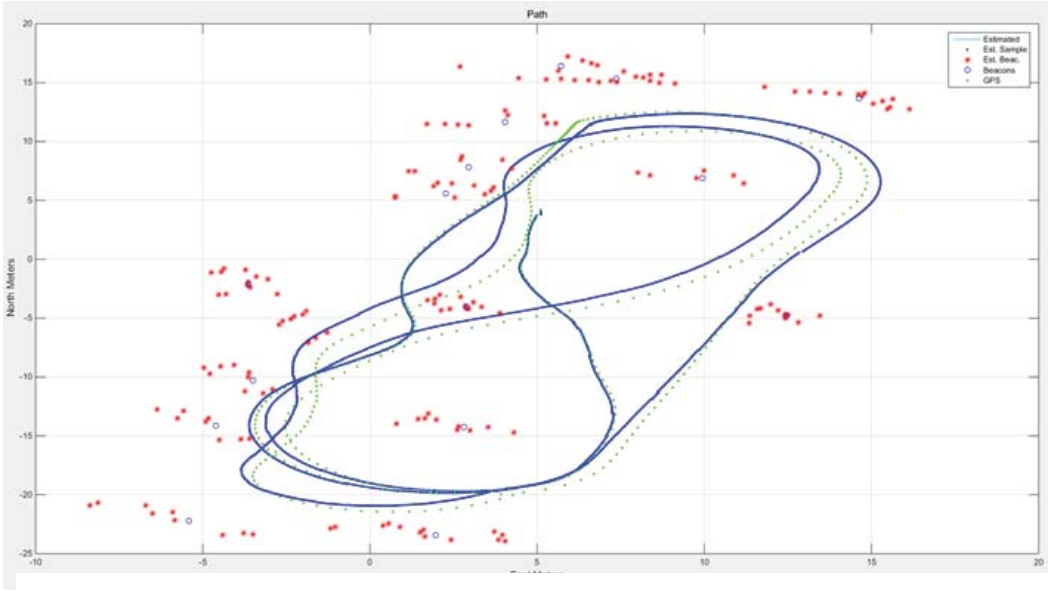


Figure 6.6 Estimated beacons (red colored points) and robot poses (blue colored lines) using the proposed approach.

$$e_{avg} = \frac{\sum_t \left(x_t^g - \frac{\sum_m x_t^{[m]}}{N} \right)^T \left(x_t^g - \frac{\sum_m x_t^{[m]}}{N} \right)}{T} \tag{6.5}$$

$$S_{error} = \frac{1}{T} \sum_t \left(e_{avg} - \left(x_t^g - \frac{\sum_m x_t^{[m]}}{N} \right)^T \left(x_t^g - \frac{\sum_m x_t^{[m]}}{N} \right) \right) \tag{6.6}$$

where T is the total time step and N is the number of particles. These are described in Table 6.1. The proposed approach leads to lower errors than the other methods. It implies that the robot poses and the map are accurately estimated by the ability of the multi-hypothesis data association in the proposed approach.

6.3 Tests using Victoria Park Dataset

We also verify the Relational RBPF-SLAM using the Victoria Park dataset, which is well-known dataset that has been used in [81]. In addition, this dataset has 6898 odometry time steps, and it offers logged range/bearing measurements from a laser range sensor. Trees in a park are used as natural features, and they are detected by a tree detection algorithm [81]. The test was conducted on Intel® Core™ i5-2500 CPU 3.30-GHz. Figure 6.7(a) shows Victoria Park in Australia provided by Google Maps. Also, Fig. 6.7(b) and Fig. 6.7(c) represent the raw odometry robot poses and robot poses obtained from GPS, respectively. From the two trajectories, we can easily note that the odometry data are highly

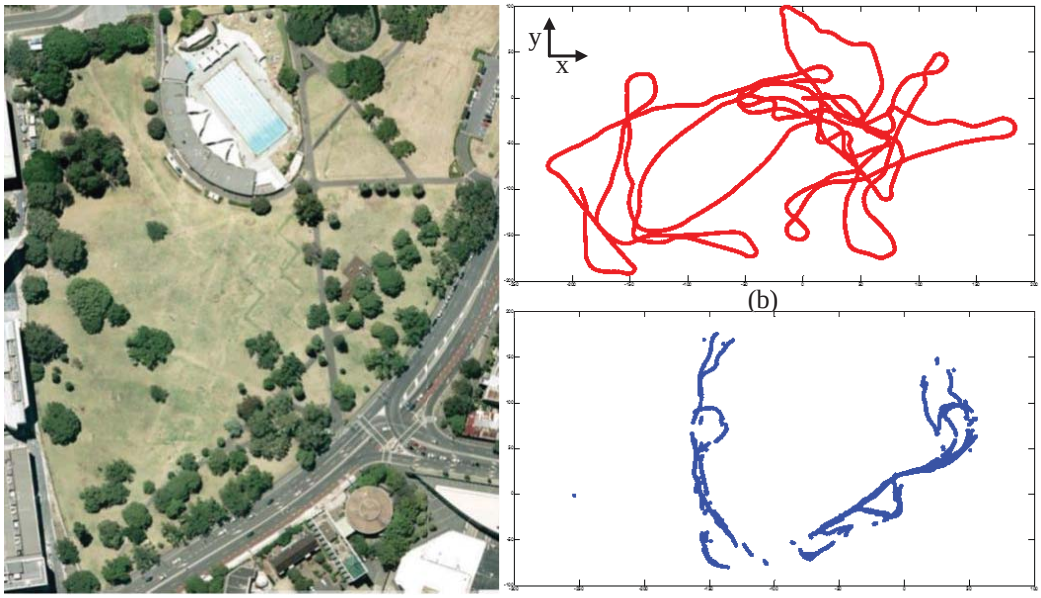


Figure 6.7 Victoria Park dataset. (a) shows Victoria Park on Google Maps. (b) and (c) represent robot trajectories obtained from odometry and GPS, respectively.

corrupted by noise. Figure 6.8 shows the estimated robot pose results from the proposed RBPF-SLAM, FastSLAM 2.0 and PSO-based FastSLAM by overlapping the true robot trajectory obtained from GPS. In this test, ten particles are employed. In case of FastSLAM 2.0 and PSO-FastSLAM, the robot trajectories are similar to the GPS result until the middle of the journey. However, the latter part of the robot journey shows numerous differences between two trajectories. The estimated robot poses obtained from the proposed RBPF-SLAM is remarkably consistent with GPS, even in the latter part of the robot journey, meaning that the proposed RBPF-SLAM maintains multiple data association hypotheses using the weight compensation and particle formation maintenance while avoiding the particle depletion problem. We also operate UFastSLAM in the test, which improves FastSLAM using the unscented transform in sampling and measurement update parts. Although its SLAM performance is better than that of FastSLAM, it exhibits the same drawback over time. As shown in Table 6.2, the proposed RBPF-SLAM has a better average execution time than the other methods. Because proposed RBPF-SLA depends on the number of particles, the circumstances are more remarkable in the first simulation. However, the gap in the execution time between FastSLAM and proposed RBPF-SLA is less than 0.03 sec despite the increase in the number of features.

TABLE 6.2

AVERAGE COMPUTATIONAL TIME (SEC)

	UFastSLAM	PSO-UFastSLAM	Proposed
Victoria Park Dataset	0.0811	0.0837	0.104

Another test is conducted using three particles, which is the minimum number of particles to form a triangular mesh structure. In this test, the control noises are $\sigma_V = 1m/s$ and $\sigma_G = 3^\circ$. The measurement noises are $\sigma_r = 0.5m$ and $\sigma_\varphi = 2^\circ$. Unlike the previous test, UFastSLAM [60], particle fission based UFastSLAM [40], PSO-based UFastSLAM [62] and the proposed RBPF-SLAM are compared one another and the results are represented in Fig. 6.9 and Fig. 6.10. In case of UFastSLAM, the SLAM performance deteriorates over time because of the tiny number of particles. Particle fission based UFastSLAM

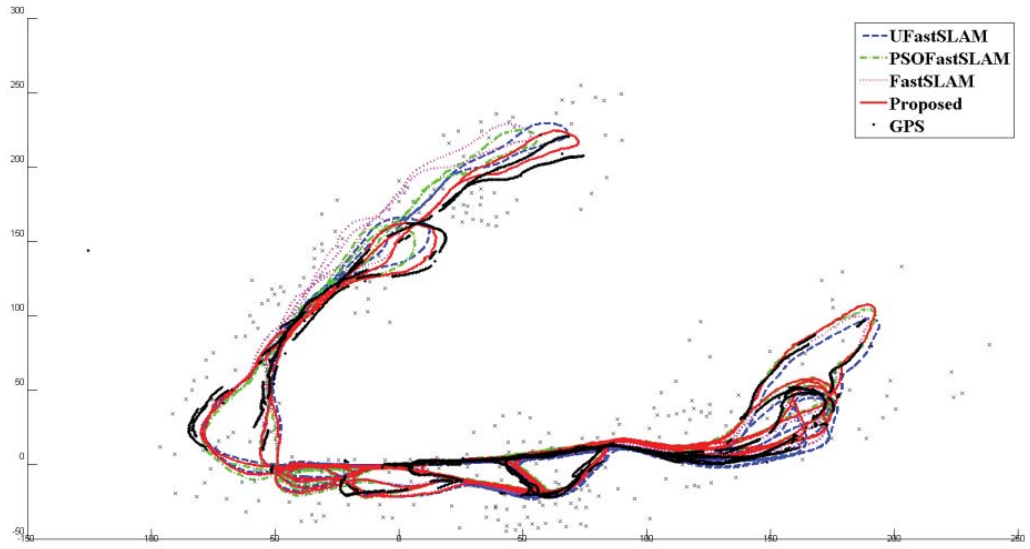
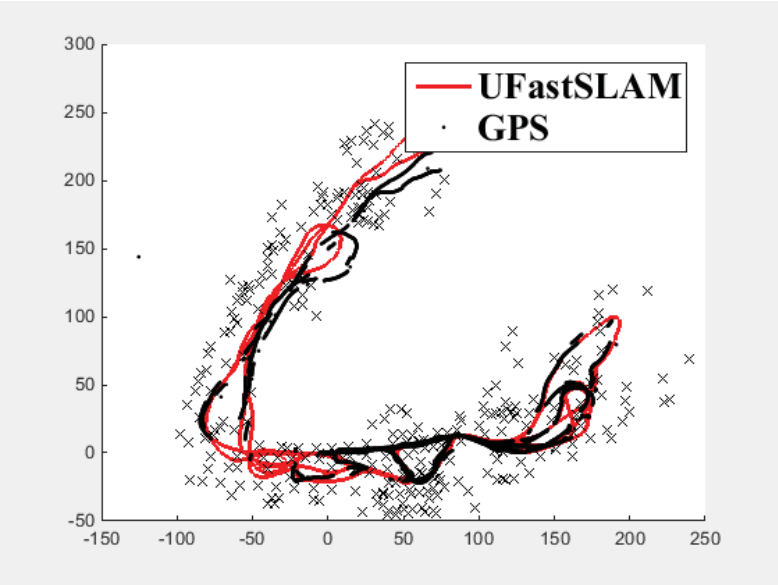
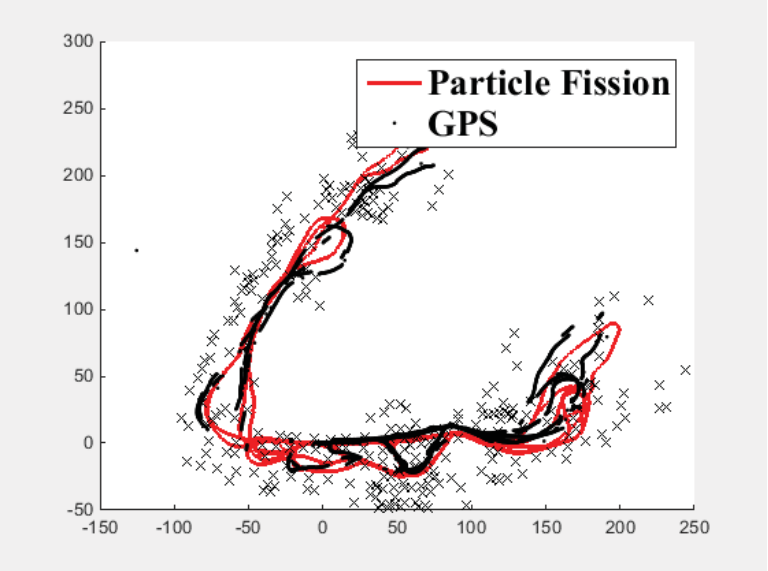


Figure 6.8 SLAM test using the Victoria Park dataset. The results from FastSLAM 2.0 (dotted line), PSO-based FastSLAM (dashed-doted line), the proposed RBPF-SLAM (solid line), UFastSLAM (dashed line) and GPS (point) are represented in a single figure. The control noises are $\sigma_V = 0.5m/s$ and $\sigma_G = 3^\circ$. The measurement noises are $\sigma_r = 0.05m$ and $\sigma_\varphi = 0.3^\circ$. Ten particles are used in the test.

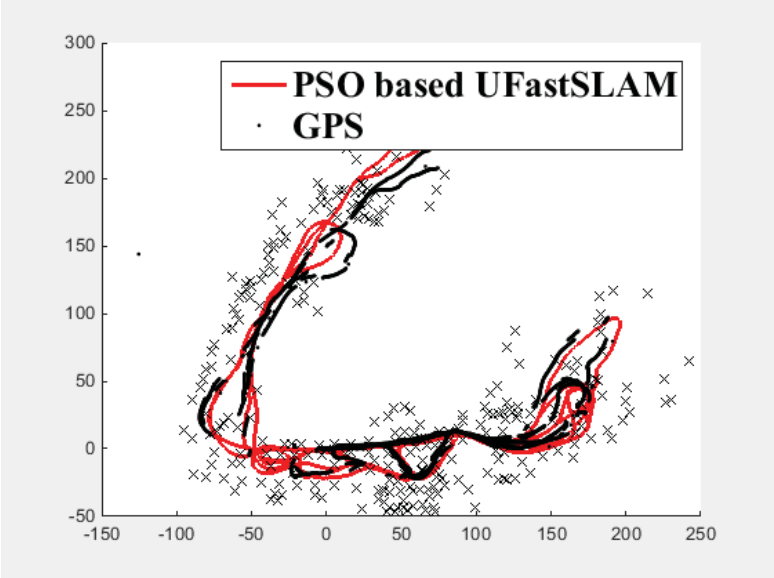
shows improvements for the robot poses and features. However, the SLAM performance also deteriorates at turning points. It is weak for rotation. The robot poses are more correctly estimated using PSO-based UFastSLAM and the proposed RBPF-SLAM. In case of PSO-based UFastSLAM, the estimation performance deteriorates around the last point. However, the estimated robot poses are remarkably consistent with GPS data in the proposed RBPF-SLAM. It means that when the odometry and measurements are highly corrupted, the proposed RBPF-SLAM is more robust than others. Especially, if a finite number of particles is used, the proposed RBPF-SLAM better results than others.



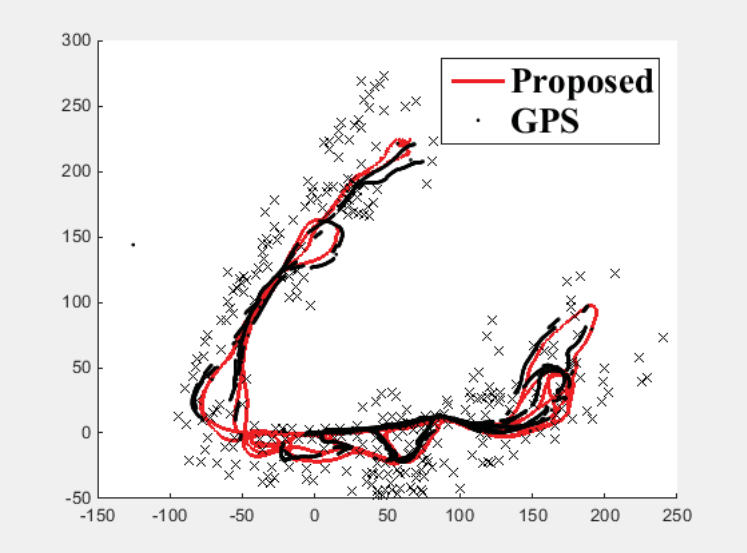


(b) Particle Fission based UFastSLAM

Figure 6.9 SLAM test using the Victoria Park dataset. The results from UFastSLAM and particle fission based UFastSLAM are represented, respectively. GPS results (black points) are also represented in two figures. The control noises are $\sigma_V = 1m/s$ and $\sigma_G = 3^\circ$. The measurement noises are $\sigma_r = 0.5m$ and $\sigma_\varphi = 2^\circ$. Three particles are used in the test.



(a) PSO based UFastSLAM



(b) Proposed RBPF-SLAM

Figure 6.10 SLAM test using the Victoria Park dataset. The results from PSO based UFastSLAM and the proposed RBPF-SLAM are represented, respectively. GPS results (black points) are also represented in two figures. The control noises are $\sigma_V = 1m/s$ and $\sigma_G = 3^\circ$. The measurement noises are $\sigma_r = 0.5m$ and $\sigma_\varphi = 2^\circ$. Three particles are used in the test.

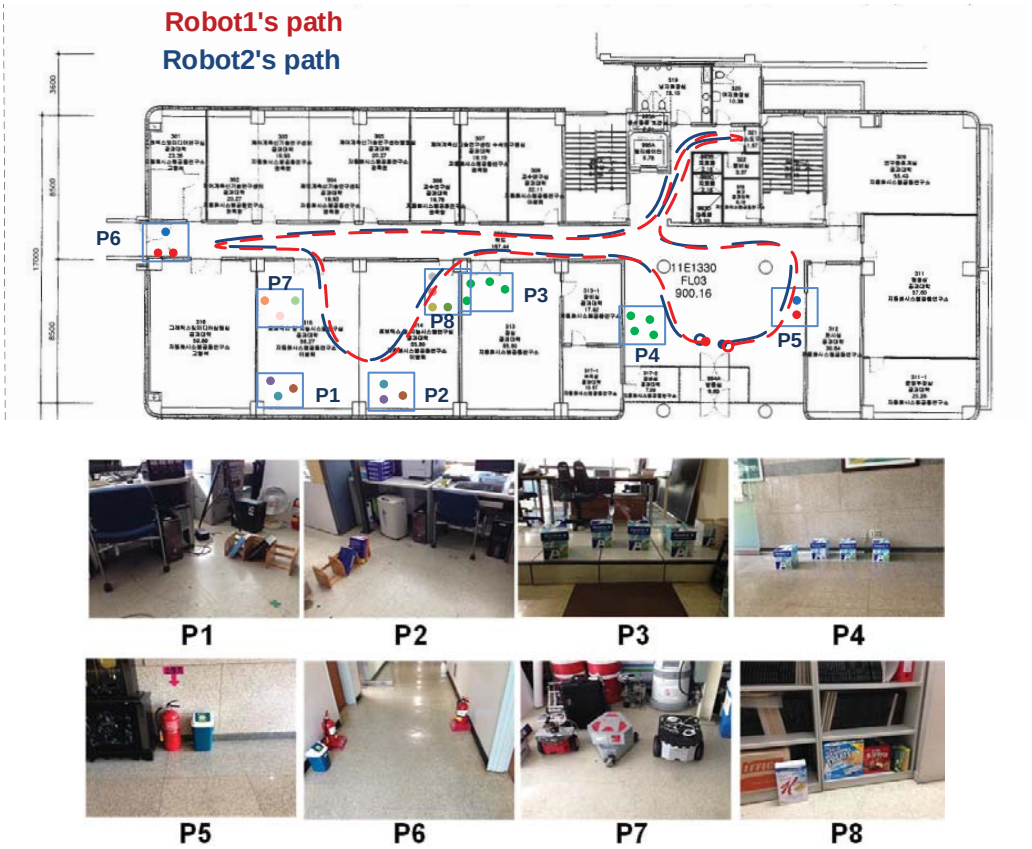


Figure 6.11 Environmental maps and the predicted robot paths. There are many objects in the office and walls between doors.

6.4 Indoor Experiments

The experiment is performed at 3 floors in Automation and Systems Research Institute in Seoul National University as an indoor environment as shown in Fig. 6.11. The size of the environment is 38m x 16m. There are many walls between doors and many objects in the office, which make the environment more complex. In many cases, data association can be failed. Each robot follows the wall according to the simple wall-following scheme with

(a) Robot hardware platform (b) Initial conditions for two robots

Figure 6.12 Robot hardware platform and initial conditions at the start point.

0.2m/s. It The Hokuyo URG-04LX-UG01 Laser Rangefinder and the Xtion Pro Camera are carried by each robot that is Pioneer 3DX as shown in Fig. 6.12. The detectable range of the laser scan sensor was 5m within $\pm 135^{\circ}$. The structure of the surrounding environment and robots are observed by these sensors.

UFastSLAM, PSO based UFastSLAM, and the proposed approach are applied to the above indoor dataset. In this test, three particles are used. Features are extracted using the Hough transform introduced in Section 6.1. For unknown data association, each line feature $\mathbf{I} = \{\rho, \theta, l, (x_s, y_s), (x_t, y_t)\}$ is matched as follows:

$$D_H = \sqrt{w_p \left(\rho_r^k - xf(1)\right)^2 + \left(1 - w_p\right)\left(\theta_r^k - xf(2)\right)^2} \tag{6.7}$$

$$D'_H = \sqrt{w_p \left(\rho_r^k - xf(1)\right)^2 + \left(1 - w_p\right)\left(\theta_r^k - xf(2)\right)^2}, \quad w_p = 0.5 \tag{6.8}$$

$$L_1 = l_r^k + xf(3) - \sqrt{\left(x_t - xf(6)\right)^2 + \left(y_t - xf(7)\right)^2} \tag{6.9}$$

$$L_2 = l_r^k + xf(3) - \sqrt{\left(x_t - xf(4)\right)^2 + \left(y_t - xf(5)\right)^2} \tag{6.10}$$

$$Angle\_diff_1 = \sqrt{\left(\theta_r^k - xf(2)\right)^2} \cdot 180 / pi \tag{6.11}$$

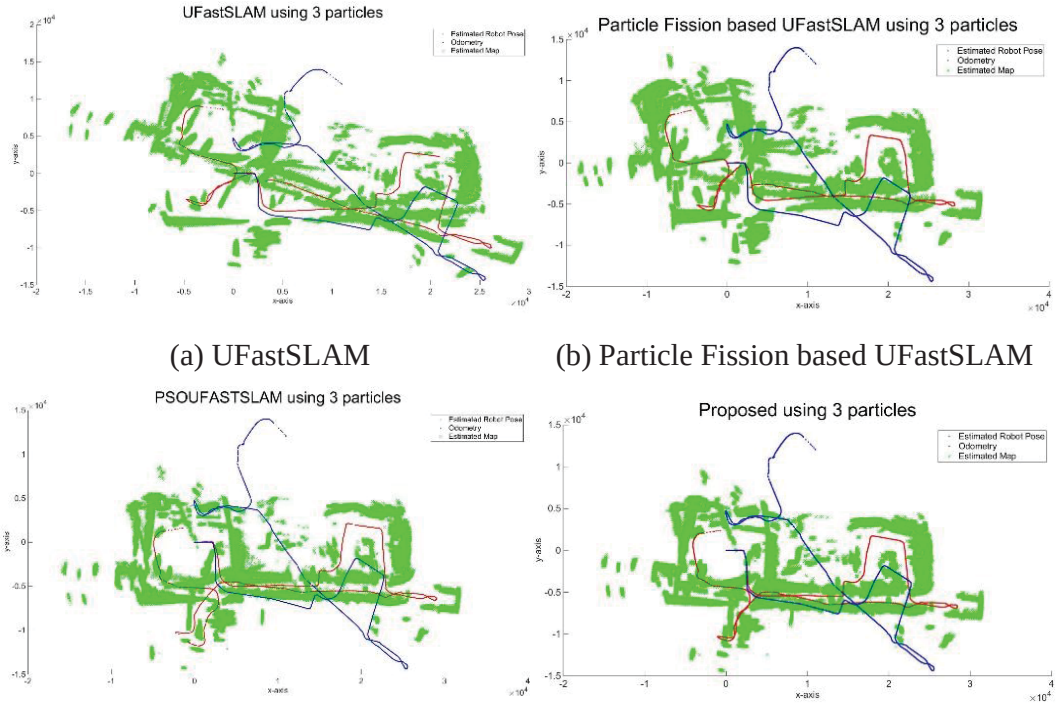
$$Angle\_diff_2 = abs(360 - Angle\_diff_1) \tag{6.12}$$

where xf is obtained from the measurement model, which is a 7x1 vector. D'_H and D_H are square root of the error terms for the distance and angle. w_p in D_H is defined as 0.8. L_1 and L_2 are the line length constraint. $Angle\_diff_1$ and $Angle\_diff_2$ are the angle difference.

From the above equations, we can find out the correspondence if the following condition is satisfied

$$\begin{aligned}&(D_H < thresh_{\min} \textbf{ or } D'_H < thresh'_{\min}) \textbf{ and } (L_1 > 0 \textbf{ or } L_2 > 0) \textbf{ and } \\ &(Angle\_diff_1 < T_1 \textbf{ or } Angle\_diff_2 < T_1)\end{aligned} \tag{6.13}$$

where $thresh_{\min}$ and $thresh'_{\min}$ are defined as 60 and 30. T_1 is 60 in degree.



(a) UFastSLAM (b) Particle Fission based UFastSLAM

(c) PSO based UFastSLAM (d) Proposed

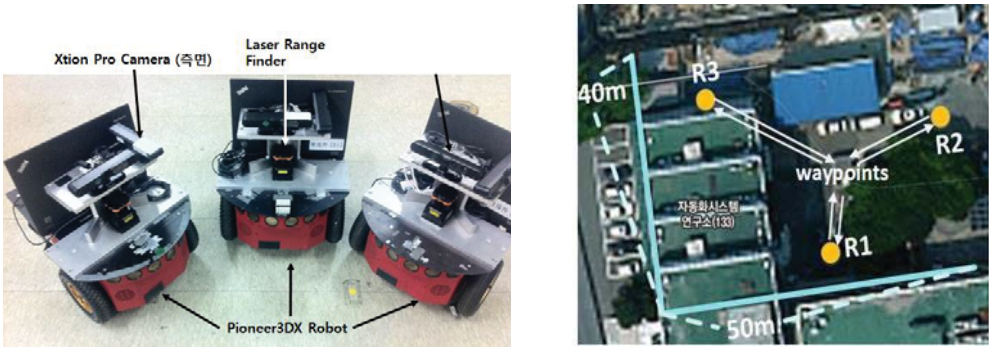
Figure 6.13 The graphical results of indoor dataset.

Figure 6.13 shows the SLAM results obtained by all methods. Specifically, the maps of R_2 is drawn for performance comparison. In case of UFastSLAM, PSO-UFastSLAM, the estimations of the robot poses are failed and the associated maps are constructed inaccurately. However, the proposed approach tracks the robot trajectory quite well and its associated map is constructed more accurately than others. Table 6.3 shows that the proposed approach more correctly estimates the robot pose than others. Here, the errors are computed by the difference between the estimated last robot pose and the true robot pose at end of its journey.

TABLE 6.3

ROBOT POSE ERROR COMPARISON

	UFastSLAM	Particle Fission based UFastSLAM	PSO based UFastSLAM	Proposed
Pose Error (cm)	1039.30	891.5572	274.58	149.29



(a) Robot hardware platform (b) Experimental environments

Figure 6.14 Robot hardware platform and experimental environments.

6.5 Outdoor Experiments

In an outdoor environment, i.e. a parking lot, an experiment is performed using the proposed RBPF-SLAM, UFastSLAM and PSO based UFastSLAM. The Hokuyo UTM-30LX Laser Rangefinder and Xtion Pro Cameras are carried by robots whose models are Pioneer 3DX as shown in Fig. 6.14(a). The detectable range of the laser scan sensor was 30m within $\pm 135°$. The structure of the surrounding environment and robots are observed by these sensors. The size of the environment is 40m x 50m which is located in the parking lot of Automation and Systems Research Institute in Seoul National University. There are three robots that have their waypoints and different start points as shown in Fig. 6.14(b). The robots move at 0.2m/s and turn to their start points. Here, the

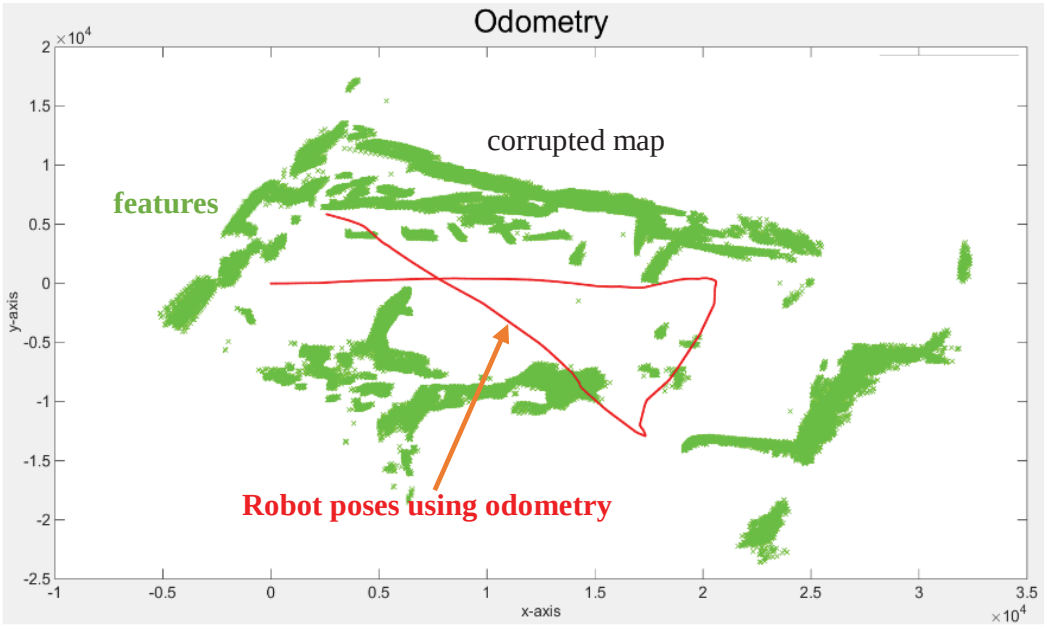


Figure 6.15 The map (green-colored points) and the robot poses (red-colored lines) using only odometry data.

frontier based exploration technique [86] was used, which is based on the grid map consisting of ‘open’, ‘occupied’ and ‘unknown’. All grids are initially filled with 0.5 that means the ‘unknown’ state. If some grids are detected by laser range data, their conditions can be changed into open area using the prior probability and the occupancy probability. The frontier is defined as the boundary between open space and uncharted territory. The robots can follow the frontiers close to the predefined waypoints.

6.5.1 Performance Comparison for Single Robot SLAM

In case of R_1, its estimated poses using its odometry and corresponding map are represented in Fig. 6.15. The odometry data of the robot is corrupted and the map is not represented accurately. To acquire the accurate robot poses and map, UFastSLAM, PSO based UFastSLAM and the proposed approach are applied to the R_1 dataset using the robot odometry and laser measurements which are used to extract line features. Three approaches estimate the robot pose and the map by varying the number of particles. In the figures, the robot p and the map are represented by red-colored line and green-colored points. When incorrect data association frequently occurs, which means the particle depletion problems can be accelerated, UFastSLAM cannot be tracking the robot poses and the map accurately as shown in Fig.6.16 and Fig.6.17 even though one-hundred particles are used. It is due to the decreased number of distinctive particles. On the contrary, the proposed approach more accurately

estimates the map and the robot pose than those of UFastSLAM. In case of ten or one-hundred particles, the map and the robot pose are estimated quite precisely. Because, each particle tracks the target consistently. In addition, the proposed approach more accurately estimates the map and the robot pose using only three particles compared to others. These are described in Fig. 6.18 and Fig. 6.19. As shown in these figures, the rotation of the map is gradually accurate according to the number of particles. It implies that each particle does not disappear or is not replicated after the resampling step but rather affects the target estimation. In short, from the result, the particle depletion problem is overcome in the proposed approach.

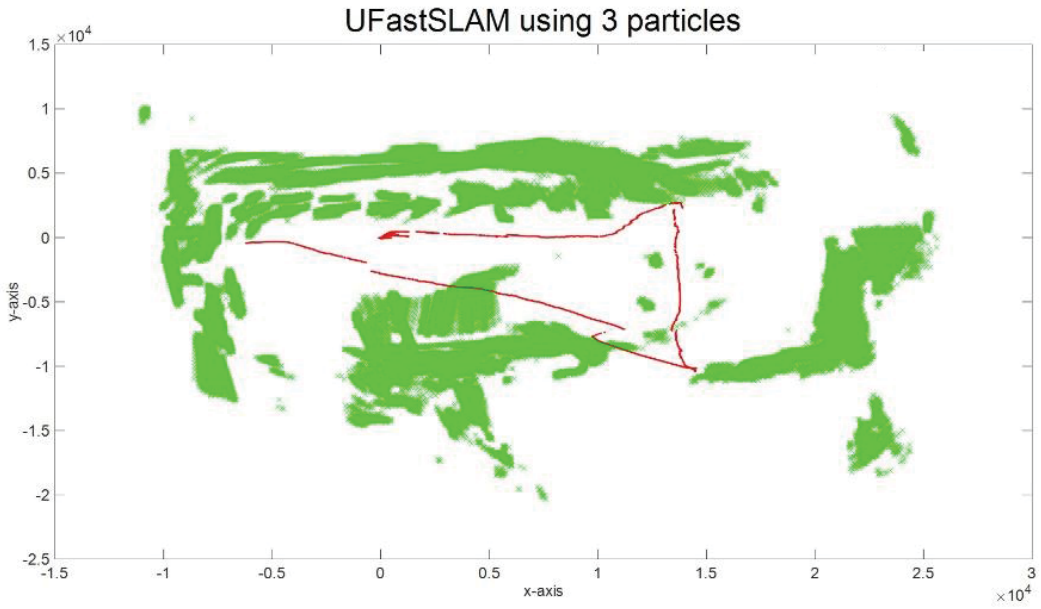
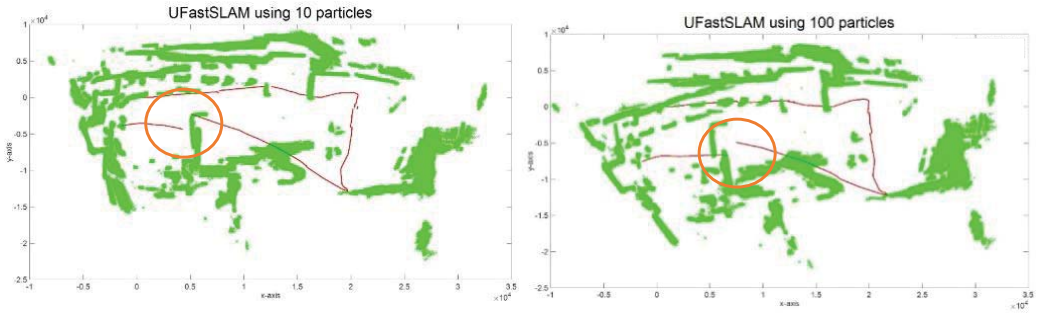


Figure 6.16 Estimated map and robot poses using UFastSLAM with three particles. These are represented by green colored points and red colored lines.



(a) Ten particles (b) One-hundred particles

Figure 6.17 Estimated maps and robot poses using UFastSLAM with ten particles and one-hundred particles. Red lines and green points are the estimated robot poses and estimated map, respectively.

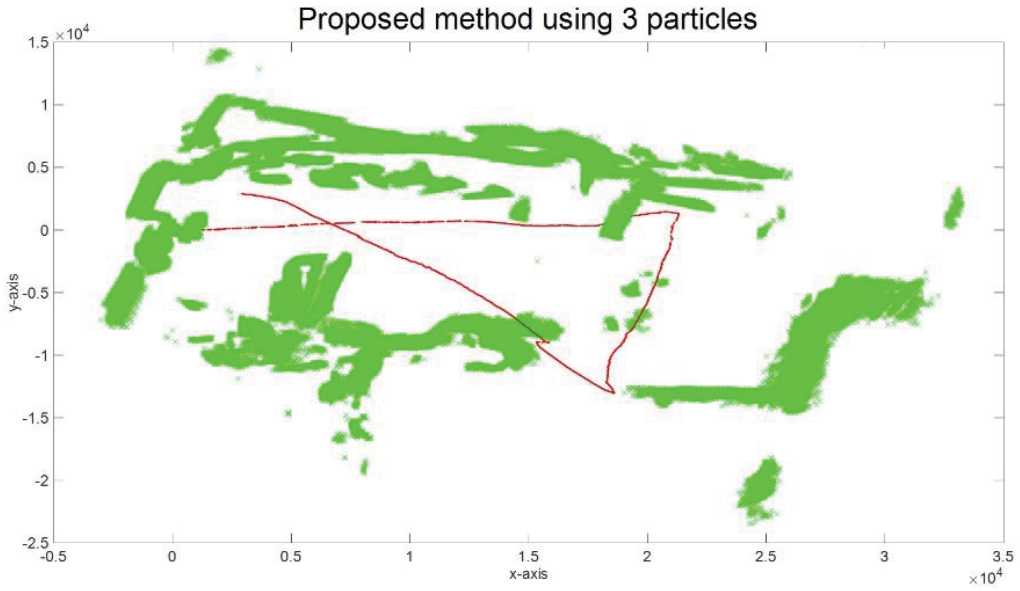
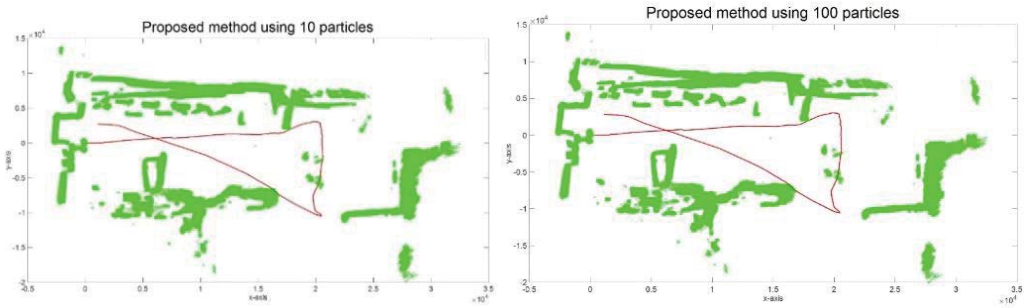
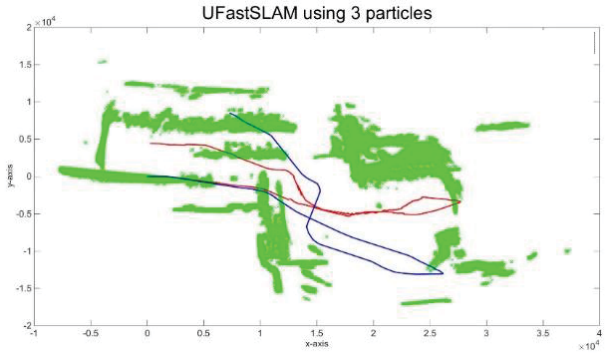


Figure 6.18 Estimated maps and robot poses using the proposed approach with three particles. Red lines and green points are the estimated robot poses and estimated map, respectively.

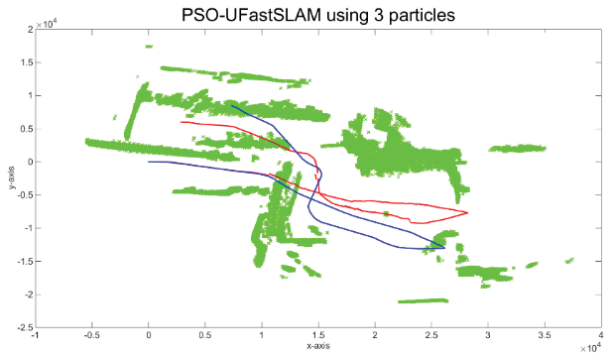


(a) Ten particles (b) One-hundred particles

Figure 6.19 Estimated maps and robot poses using the proposed approach. Red lines and green points are the estimated robot poses and estimated map, respectively.



(a) UFastSLAM



(b) PSO based UFastSLAM

Figure 6.20 Estimated maps and robot poses for R_3. The odometry and the estimated poses are represented by the blue line and the red line, respectively. (a) and (b) are UFastSLAM and PSO based UFastSLAM, respectively.

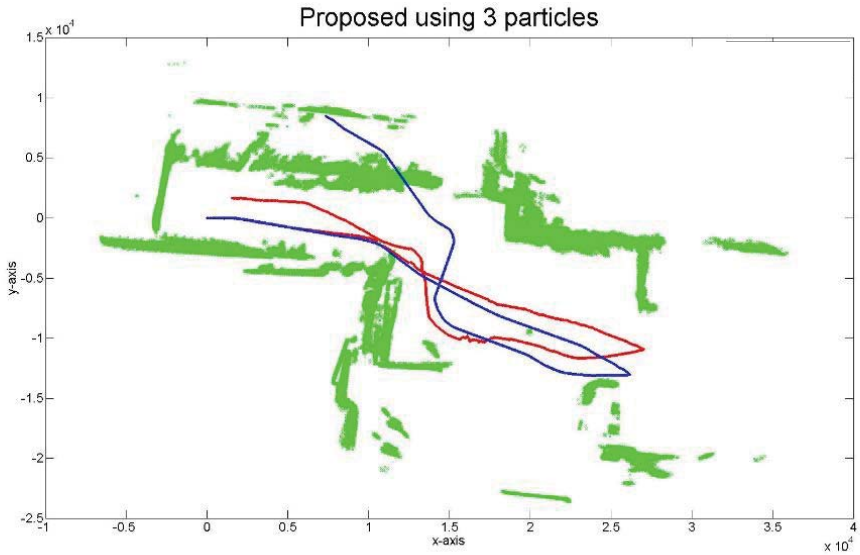


Figure 6.21 Estimated maps and robot poses for R_3. The odometry and the estimated poses are represented by the blue line and the red line, respectively.

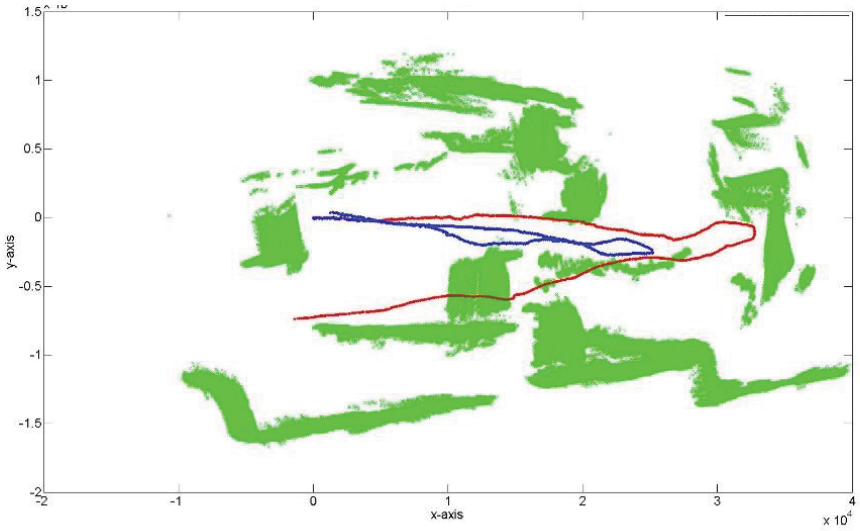


Figure 6.22 Estimated maps and robot poses for R_2 using UFastSLAM. The odometry and the estimated poses are represented by the blue line and the red line, respectively.

TABLE 6.4

POSE ERROR COMPARISON

	UFastSLAM	PSO based UFastSLAM	Proposed
R_1 Error (cm)	431.96	58.29	55.09
R_2 Error (cm)	512.26	389.31	131.70
R_3 Error (cm)	391.60	50.56	43.81
Avg Error	445.27	339.38	76.86

The estimated maps and poses of other robots are also obtained using the above mentioned methods as shown in Fig.6.19, Fig.6.20 and Fig.6.22. The proposed approach gives better performance than others in terms of accuracy for the robot poses and the map as shown in Fig.6.23. In case of R_2, the map using the proposed RBPF-SLAM is not better than one of PSO-UFastSLAM in terms of the rotation error because the raw sensor data is quite accurate using the sensor matching method. This case is the exceptional case because we assume that measurement noise is always smaller than one of odometry. The total merged maps are represented using the three methods as shown in Fig. 6.24. The maps of R_2 and R_3 are involved in one of the reference robot R_1. In some points, the maps obtained from the proposed RBPF-SLAM framework are more overlapped than other methods, which means that the map is more correctly estimated using weight compensation and particle formation maintenance schemes. The robot pose errors which are measured through comparison between the last estimated robot pose and its true pose are shown

in Table 6.4. Since the problem of tires of R_2 occurs, we can use not the odometry data but the result of scan matching. Here, the iterative closest point (ICP) algorithm is used for scan matching.

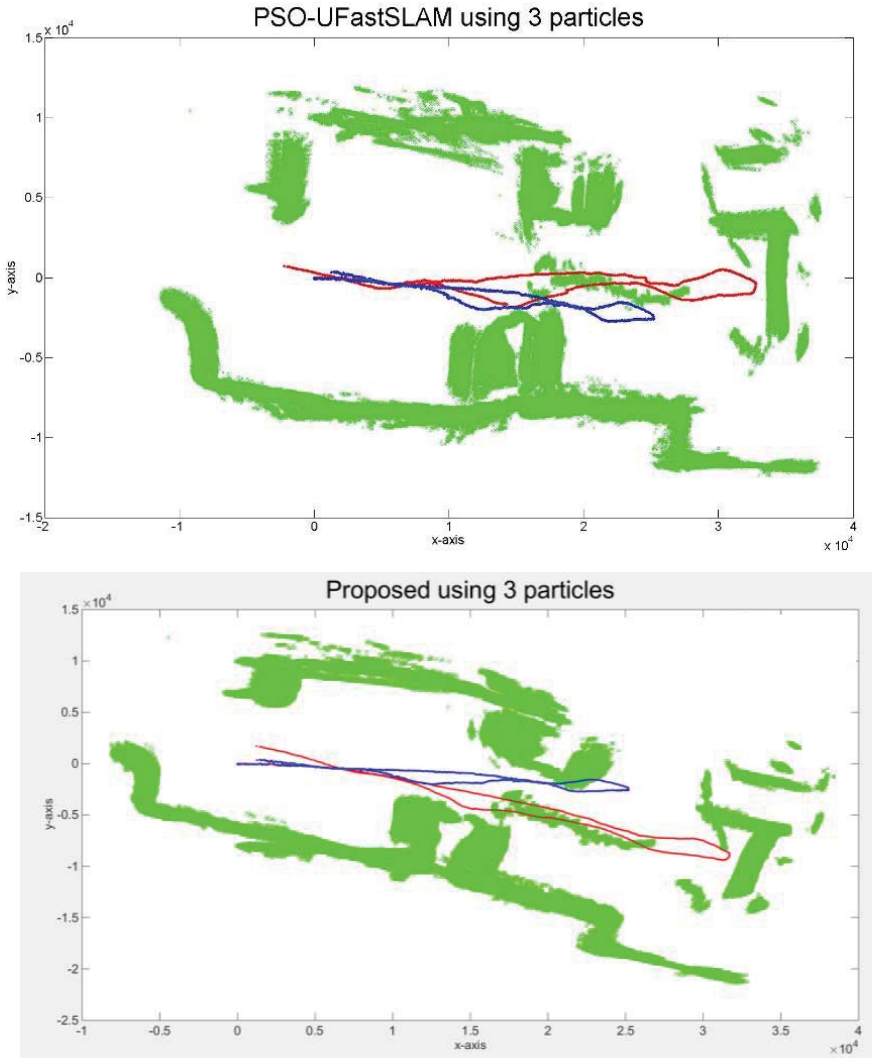
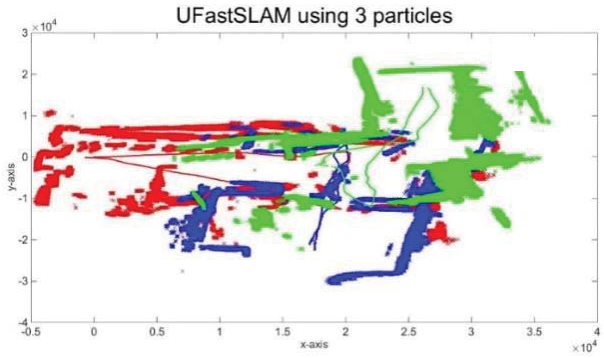
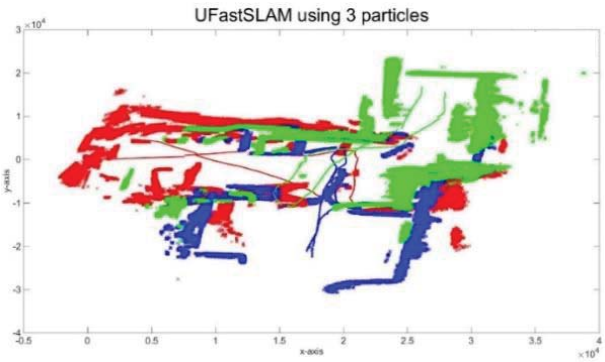


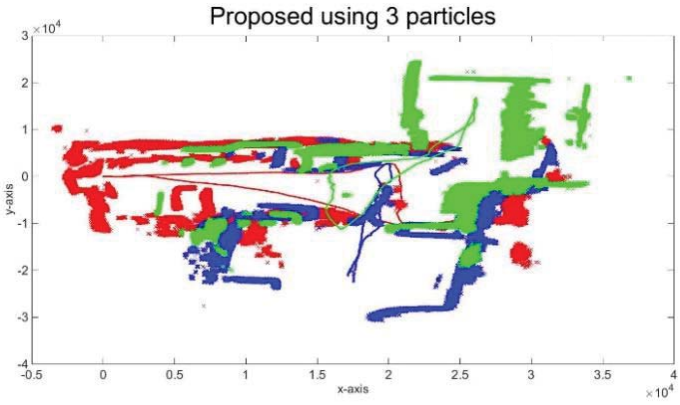
Figure 6.23 Estimated maps and robot poses for R_2 using the PSO-UFastSLAM and the proposed approach. The odometry and the estimated poses are represented by the blue line and the red line, respectively.



(a) UFastSLAM



(b) PSO-UFastSLAM



(c) Proposed

Figure 6.24 Total merged map using the UFastSLAM (a), PSO-UFastSLAM (b) and proposed approach (c).

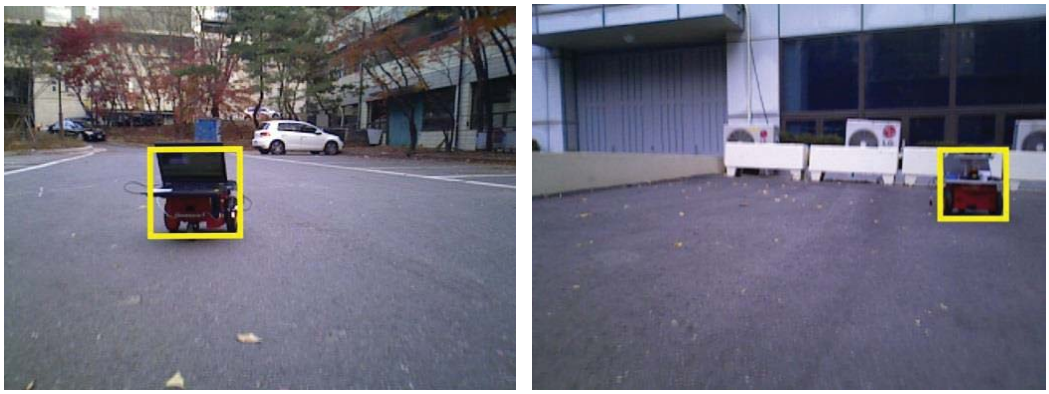
The results of the robot pose errors show that the proposed approach outperforms other methods. PSO based UFastSLAM has better performance than UFastSLAM but PSO may degenerate the filter at some points in the large environment. From the results, the proposed schemes, i.e. weight compensation and particle formation maintenance, are well operated during the journey of each robot.

6.5.2 Performance Comparison for Data Consensus

Three robots can meet (*Rendezvous*) or share several features (*Feature-Sharing*) one another during their journeys. To meet other robots, each robot should detect other robots using its front vision sensor. Here, the multi-view of the robot was learned using binary descriptor, i.e. *brisk features*, in advance. The brisk features have some advantages which are fast computational speed using Hamming distance [85] and robust for blur and rotation [84]. The detected robots are represented in Fig.6.25 which are represented as the yellow box. After the detection, they can obtain the inter-robot measurements that are the distance and the angle between them. The acquired inter-robot measurements are used in the KCIF process in the proposed approach.

In case of feature-sharing events, their line features or common places are shared and can be updated according to the KCIF update scheme. Figure 6.26 shows the shared place between two robots. The detected times are different but the vast amount of information is shared at that place. In this experiment,

the similarity score is set to zero point six. Using the above mentioned events, the initial pose of all robots are given and approximatively described in Table 6.5.



(a) Detected by R_1 (b) Detected by R_2

Figure 6.25 Rendezvous. The inter-robot measurements occur after robot detection.

(a) Detected by R_1 (b) Detected by R_2

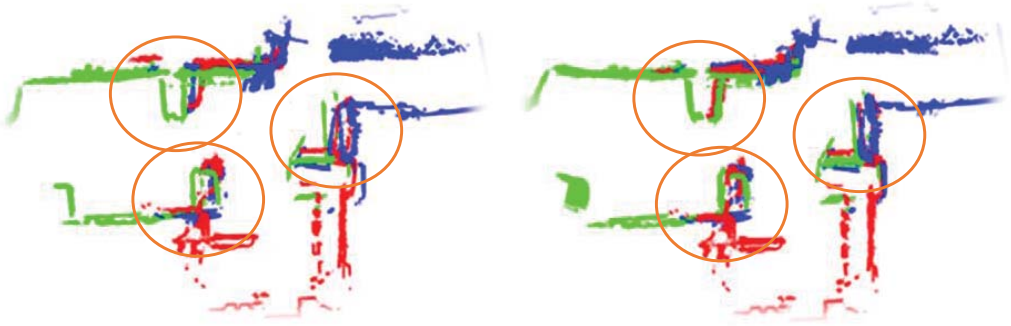
Figure 6.26 Feature-sharing event. The inter-robot measurements also occur.

TABLE 6.5

INITIAL POSE OF ALL ROBOTS

	R_1	R_2	R_3
Robot Poses	(0m, 0m, 0°)	(22423.8m, 17494.9m, 90°)	(-16834m, 26127m, -90°)

Since robots cannot be suddenly emerged and manually controlled by human at the beginning, the given initial conditions are not critical issues anymore. From the initial conditions, the methods, i.e. the conventional approach and the proposed approach, can be applied to the outdoor dataset. Figure 6.27 represents the total maps obtained from the methods. In the total map obtained from the proposed RBPF-SLAM framework, the overlapped areas are more correctly estimated using inter-robot measurements than ones of the conventional approach. From Table 6.6, the proposed RBPF-SLAM framework estimates the final pose of the robot more accurately than one



(a) Conventional Approach (b) Proposed Approach

Figure 6.27 Final merged map obtained from all robots. (a) and (b) are the total maps obtained from the conventional approach and the proposed approach, respectively.

obtained from the conventional approach. It implies that the consensus scheme is operated pretty well.

TABLE 6.6
COMPARISON OF ROBOT POSE ERRORS

	Conventional	Proposed
Average Robot x-y Pose Errors	99.06cm	68.05cm
Average Robot orientation Errors	8.4°	5.9°

6.6 Discussions

In the Chapter 6, several tests are performed using the formal dataset, i.e. Car park dataset and Victoria park dataset, and indoor and outdoor real experiments. In the tests, the proposed SLAM framework was compared with UFastSLAM [60], PSO based UFastSLAM [62], particle fission based UFastSLAM. In case of particle fission based UFastSLAM, the concept of particle fission is applied to UFastSLAM, which was originally used in FastSLAM [40]. In most cases, three particles were used in RBPF-SLAM. In case of Car park dataset, each method was employed using different number of particles as well as three particles. In the graphical results, the distribution of particles in the proposed approach was more scattered than others because it steadily makes particles constitute a formation. The errors are measured using the difference between GPS data and estimated robot poses when GPS data is

available. Overall, the proposed SLAM framework had better performance than others. Especially, when the small number of particles was used, the proposed approach showed better results than others. It reduced the errors up to 19 percent. These are described in Section 6.2. The SLAM methods were also applied to the Victoria park dataset for graphical comparison. The tests were performed in different motion and measurement noise. The robot poses obtained from the proposed approach are more overlapped with GPS data than ones estimated by other methods even when the motion and measurements are highly corrupted. These are described in Section 6.3. In case of the indoor experiment, the experiment was performed at 3 floors in Automation and Systems Research Institute in Seoul National University whose size is 38m x 16m. UFastSLAM and particle fission based UFastSLAM failed the estimation of the robot poses and maps from the wrong data association. However, PSO based UFastSLAM and the proposed SLAM approach estimated the robot poses and maps until the end. Among them, we can know that the proposed SLAM approach more accurately estimated the robot poses and maps than the PSO based UFastSLAM from the error result at the last robot pose. It is proper that errors at the last robot pose are compared because errors are accumulated in SLAM. By comparing UFastSLAM, the proposed SLAM approach reduced the error from 1039.30cm to 149.29cm, which are shown in Section 6.4. In an outdoor environment, i.e. a parking lot, an experiment was performed using the proposed RBPF-SLAM, UFastSLAM and PSO based UFastSLAM. Like the

indoor experiment, the errors are measured at the last robot pose. In addition, the average errors from the results of three robots were computed and compared. In case of all cases, PSO based UFastSLAM is better performance than UFastSLAM. The proposed RBPF-SLAM framework showed better results in case of all robots than others. It means the proposed approach maintains the number of distinctive particles until the end. The PFM is operated more robustly than PSO. It implies that particles converge a wrong point at some points using PSO, which can cause the particle depletion problem. However, in case of R_3, PSO based UFastSLAM showed more accurate rotation in the map than the proposed RBPF-SLAM framework. It is due to the exceptional case. Originally, RBPF-SLAM assume that measurement noise is always smaller than one of odometry. Since noise by the scan matching method is lower than the noise by the measurement in some cases, the assumption is broken. Therefore, PSO based UFastSLAM may better performance in this condition. These are described in Section 6.5.1. In Section 6.5.2, the consensus scheme are performed when two robots meet each other, i.e. the rendezvous event. As a conventional approach, map-merging algorithm was applied, which does not compensate the map itself. When several rendezvous occur, the consensus is consistently performed, which results in the robot poses are more correctly estimated and errors are reduced by 32 percent at the last pose. From the results, the enhancement of the sampling step in the RBPF framework is verified. These are described in Table 6.7.

TABLE 6.7

RESULTS OF SEVERAL TESTS

Experiments	Target	Error Reduction Rate			Location
Car Park Dataset	Single Robot	[60]	[40]	[62]	Section 6.2
		19%	33%	7%	
Victoria Park Dataset	Single Robot	Generate more overlapped map with GPS data			Section 6.3
Indoor Dataset	Single Robot	[60]	[40]	[62]	Section 6.4
		86%	83%	46%	
Outdoor Dataset	Single Robot	[60]		[62]	Section 6.5
		83%		54%	
Outdoor Dataset	Multiple Robots	[89]			Section 6.5
		31%			

Chapter 7

Conclusions

This dissertation presents a novel Rao-Blackwellized particle filter based SLAM (RBPF-SLAM) framework using geometric information among particles and inter-robot measurements. The conventional RBPF-SLAM framework is often degenerated in large environments due to the inherent problems, i.e. the particle depletion problem, the data association problem and the filter. These problems originally come from the improper weight assignment and brutal rejection and replication of particles. Thus, all parts of RBPF-SLAM, i.e. the sampling step, the feature estimation step, the importance weight step, and the resampling step are enhanced using *particle to particle cooperation* and *robot to robot cooperation*. The concepts of particle to particle cooperation are applied to the importance weight step and the resampling step. To find out some particles with the improperly assigned weight, two clusters

are generated using the k-means algorithm and the EM algorithm, which denotes an area where the target is highly involved and the other area where the target is not located. In the two clusters, improperly assigned weights are corrected. To eliminate the rejection and the replication of the resampling step, particles form a triangular mesh structure and maintain this formation until the end of SLAM. These are verified from the formal simulations, tests using Victoria park dataset and Car park dataset and outdoor experiments, which are introduced in Section 5.2, Section 5.3, Section 6.2, Section 6.3 and Section 6.4. The estimated robot poses and estimated features obtained from the proposed RBPF-SLAM are compared with competitive methods, which are FastSLAM 2.0, PSO based FastSLAM, PSO based UFastSLAM, particle Fission based UFastSLAM and UFastSLAM.

For multi-robot systems, the consensus scheme is involved in the proposed RBPF-SLAM framework, which results in the enhancement of the sampling step and the feature estimation step. The estimated maps obtained from each robot are unified in the coordinates of reference robot R_f. When a rendezvous between two robots occurs, their poses are more correctly estimated using the consensus scheme, which gives an improvement in the sampling step of the RBPF-SLAM framework. Here, the proposed RBPF-SLAM framework adopts a Kalman filter based consensus scheme, which is robust than the covariance intersection method. When two robots share a common feature, this feature is more correctly estimated via the Kalman filter based consensus scheme, which

results in an improvement of the feature estimation step. Here, the average particles information, not information of a particle with maximum weight, of one robot is transmitted to the other robot. Simulations to verify SLAM performance by robot to robot cooperation, is carried out in Section 5.4. The proposed approach is also verified in a large outdoor environment as described in Section 6.4. In this experiment, the results show the proposed approach outperforms other methods using the geometric information of particles and inter-robot measurements between robots.

Bibliography

[1] P. J. Besl and N. D. McKay, “A Method for Registration of 3-D Shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992.

[2] Y. Chen and G. Medioni, “Object Modeling by Registration of Multiple Range Images,” *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2724-2729, 1991.

[3] Z. ZHANG, “Iterative Point Matching for Registration of Free-Form Curves and Surfaces,” *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119-152, 1994.

[4] F. Lu and E. Milios, “Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans,” J*ournal of Intelligent and Robotic Systems: Theory and Applications*, vol. 18, no. 3, pp. 249-275, 1997.

[5] A. Diosi and L. Kleeman, “Fast Laser Scan Matching using Polar Coordinates,” *The International Journal of Robotics Research*, vol. 26, no. 10, pp. 1125-1153, Oct. 2007.

[6] Seung-Hee Lee, Heon-Cheol Lee, and Beom-Hee Lee, "A Scan Restoration Method for Robust Polar Scan Matching in Dynamic Environments," *Advanced Robotics*, vol. 27, no. 11, pp. 877-891, 2013.

[7] A. Censi, L. Iocchi, and G. Grisetti, “Scan Matching in the Hough Domain,” *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 2005.

[8] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Proceedings of Robotics: Science and Systems*, Seattle, USA, 2009.

[9] P. Biber and W. Straber, “The Normal Distributions Transform: A New Approach to Laser Scan Matching,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, USA, 2003.

[10] F. Lu and E. Milios, “Globally consistent range scan alignment for environment mapping,” *Autonomous Robots*, pp. 333–349, 1997.

[11] U. Frese, P. Larsson, and T. Duckett, “A Multilevel Relaxation Algorithm for Simultaneous Localization and Mapping,” *IEEE TRANSACTIONS ON ROBOTICS*, vol. 21, no. 2, 2005.

[12] F. Dellaert and M. Kaess, “Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing,” *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181-1203, 2006.

[13] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard “A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent,” *Proceedings of Robotics: Science and Systems (RSS)*, 2007

[14] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “g2o: a general framework for graph optimization,” *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3607-3613, 2011.

[15] E. Olson, J. Leonard, and S. Teller, “Fast Iterative Alignment of Pose Graphs with Poor Initial Estimates,” *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2262-2269, 2006

[16] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM: Incremental Smoothing and Mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365-1378, 2008.

[17] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “SAM2: Incremental Smoothing and Mapping Using the Bayes Tree,” *International Journal of Robotics Research*, vol. 31, no. 2, pp. 216-235, 2012.

[18] J. Sprickerhof, K.Lingemann, J. Hertzberg, and A. Nuchter, “A heuristic loop closing technique for large-scale 6D SLAM,” *Automatika*, vol. 52, no.

3, pp. 199-222, 2011.

[19] K. W. Lee, S. H. Lee and B. H. Lee, “Pose Compensation Framework with Explicit Loop Closing for Multiple Robots,” *Electronics Letters*, vol. 50, no. 22, pp. 1581-1583, 2014.

[20] M. Cummins, P. Newman, “FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance,” *International Journal of Robotics Research*, vol. 27, no.6, pp.647-665, 2008.

[21] E. Olson, P. Agarwal, “Inference on networks of mixtures for robust robot mapping,” *International Journal of Robotics Research*, vol. 32, no. 7, pp. 826-840, 2013.

[22] G. H. Lee, F. Fraundorfer, and M. Pollefeys, “Robust Pose-Graph Loop-Closures with Expectation-Maximization,” *Source of the DocumentIEEE International Conference on Intelligent Robots and Systems*, pp. 556-563, 2013.

[23] N. Ayache and O.D. Faugeras, “Maintaining representations of the environment of a mobile robot,” *IEEE Trans on Robotics and Automation*, vol. 5, no. 6, pp. 804-819, 1989.

[24] R. Smith, M. Self, and P. Cheeseman, “A stochastic map for uncertain spatial relationships,” *International Symposium on Robotics Research*, 1987.

[25] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, “Simultaneous localization and mapping with sparse extended information filters,” *International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693-716, 2004.

[26] M. R. Walter, R. M. Eustice, and J. J. Leonard, “Exactly Sparse Extended Information Filters for Feature-Based SLAM,” *International Journal of Robotics Research*, vol. 26, no. 4, pp. 335-359, 2007.

[27] J. Nieto, T. Bailey, and E. Nebot, “Recursive scan-matching SLAM,” *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 39-49, 2007.

[28] H. Durrant-Whyte and T.Bailey, “Simultaneous localisation and mapping (SLAM): Part I - the essential algorithms,” *Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99-110, 2006.

[29] S. Huang, Z. Wang and G. Dissanayake, “Sparse Local Submap Joining Filter for Building Large-Scale Maps,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1121-1130, 2008.

[30] G. P. Huang “On the complexity and consistency of UKF-based SLAM,” *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4401-4408, 2009.

[31] X. Wang and H. Zhang, “A UPF-UKF framework for SLAM,” *Proceedings of IEEE International Conference on Robotics and*

Automation, pp. 1664-1669, 2007.

[32] L. Qu, S. He, Y. Qu, “An SLAM algorithm based on improved UKF,” *Proceedings of 24th Chinese Control and Decision Conference (CCDC)*, pp. 4154–4157, 2012.

[33] K. Murphy, “Bayesian map learning in dynamic environments,” *Proceedings of Conference on Neural Information Processing Systems*, pp. 1015-1021, 1999.

[34] A. Doucet, J.F.G. de Freitas, K. Murphy, and S. Russel, "Rao-Blackwellized partcile filtering for dynamic bayesian networks", *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, Stanford, CA, USA, pp. 176-183, 2000.

[35] M. Montemerlo, S. Thrun, D. Koller, D, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” *Proceedings of the National Conference on Artificial Intelligence, Alberta, Canada*, pp. 593-598, 2002.

[36] M. Montemerlo, S. Thrun, D. Koller, D, and B. Wegbreit, “Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges,” *Proceedings of International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, pp. 1151-1156, 2003.

[37] G. Grisetti, C. Stachniss, and W. Burgard, “Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling,” *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 2432-2437, 2005.

[38] C. Stachniss, D. Hahnel, W. Burgard, “Exploration with Active Loop-Closing for FastSLAM,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[39] C. Stachniss, D. Hahnel, W. Burgard, and G. Grisetti, “On Actively Closing Loops in Grid-based FastSLAM,” *Advanced Robotics*, vol. 19, no. 10, pp. 1059-1079, 2005.

[40] Z. Q. Wei, J. Cao, B. Yin, and B. Liu, “Improved FastSLAM based on the Particle Fission for Mobile Robots,” *2010 8th IEEE International Conference on Control and Automation, ICCA 2010*, pp. 1379-1384, 2010.

[41] I. Kim, N. Kwak, H. Lee, and B. Lee, “Improved Particle Filter using Geometric Relation between Particles in FastSLAM,” *Robotica*, vol. 27, no. 6, pp. 853-859, 2009.

[42] N. Kwak, G. W. Kim, and B. H. Lee, “A new compensation technique based on analysis of resampling process in FastSLAM,” *Robotica*, vol. 26, no. 2, pp. 205-217, 2008.

[43] D. Liu, G. Liu, and M. Yu, “An Improved FastSLAM Framework Based

on Particle Swarm Optimization and Unscented Particle Filter," *Journal of Computational Information Systems*, vol. 8, no. 7, pp. 2859-2866, 2012.

[44] R. Havangi, M. A. Nekoui, and M. Teshnehlab, "An improved FastSLAM framework using soft computing," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 20, no.1, pp. 25-46, 2012.

[45] J. C. Ma, Q. Zhang, and L. Y. Ma, "A novel robust approach for SLAM of mobile robot," *Journal of Central South University*, vol. 21, no. 6, pp. 2208-2215, 2014.

[46] H. C. Lee, S. Park, J. S. Choi, and B. H. Lee, "PSO-FastSLAM: An improved FastSLAM framework using particle swarm optimization," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, TX, USA, pp. 2842-2847, 2009.

[47] G. Zhu, D. Liang, Y. Liu, Q. Huang, and W. Gao, "Improving particle filter with support vector regression for efficient visual tracking," *Proceedings of International Conference on Image Processing*, Genoa, Italy, pp. 422-425, 2005.

[48] W. Jiang, G. Yi, and Q. Zeng, "Application of proximal support vector regression to particle filter," *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems*, Shanghai, China, pp. 239-243, 2009.

[49] S. H. Lee, Y. J. Cho, and B. H. Lee, “Adaptive Weight Compensation Technique for Improved FastSLAM,” *Proceedings of IEEK Summer Conference*, Jeju, Republic of Korea, pp.1738-1741, 2013. (In Korean).

[50] S. H. Lee and B. H. Lee, “Selective Weight Controlling Technique in Rao-Blackwellized Particle Filter for SLAM,” *Proceedings of KRoC 2012*, Gangneung, Republic of Korea, pp.551-554, 2012. (In Korean)

[51] S. H. Lee, H. C. Lee, G. Eoh, and B. H. Lee, “A New Formation Maintenance Technique for Particle Diversity in RBPF-SLAM,” *Applied Mechanics and Materials*, 330, pp. 629-634, 2013.

[52] S. H. Lee, G. Eoh, and B. H. Lee, “Relational FastSLAM: An Improved Rao-Blackwellized Particle Filtering Framework using Particle Swarm Characteristics,” *Robotica*, 2014. (online published)

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

[54] R. Sim, P. Elinas, M. Griffin and J. J. Little, “Vision-based SLAM using the Rao-Blackwellised Particle Filter,” *Proceedings of IJCAI Workshop on Reasoning with Uncertainty in Robotics*, vol. 14, no. 1, pp. 9-16, 2005.

[55] R. Sim, P. Elinas, and J. J. Little, “A Study of the Rao-Blackwellised Particle Filter for Efficient and Accurate Vision-Based SLAM,”

International Journal of Computer Vision, vol. 74, no. 3, pp. 303-318, 2007.

[56] J. Kim, K. Yoon, and I. S. Kweon, "Bayesian filtering for keyframe-based visual SLAM," *International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 517-531, 2015.

[57] M. Calonder, "EKF SLAM vs. FastSLAM: A Comparison," *Technical Report CVLAB-REPORT*, 2010.

[58] A. Brooks and T. Bailey "HybridSLAM: Combining FastSLAM and EKF-SLAM for Reliable Mapping," *Proceedings of Algorithmic Foundation of Robotics VIII; Springer-Verlag*, pp. 647-661, 2008.

[59] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An Efficient FastSLAM Algorithm for Generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurements," *Proceedings of the IEEE/RSJ Intelligent Robots and Systems*, 2003.

[60] C. Kim, R. Sakthivel, and W. K. Chung, "Unscented FastSLAM: A robust and efficient solution to the SLAM problem," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 808-820, 2008.

[61] C. Kim, H. Kim, and W. K. Chung, "Exactly Rao-blackwellized Unscented Particle Filters for Slam," *Proceedings of IEEE International Conference on Robotics and Automation*, Shanghai, China, pp. 3589-3594, 2011.

[62] B. He, L. Ying, S. Zhang, X. Feng, T. Yan, R. Nian, and Y. Shen,

"Autonomous navigation based on unscented-FastSLAM using particle swarm optimization for autonomous underwater vehicles," *Measurement: Journal of the International Measurement Confederation*, vol. 71, pp. 89-101, 2015.

[63] C. Qiu, X. Zhu, and X. Zhao, "Vision-based unscented FastSLAM for mobile robot," *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, pp. 3758-3763, 2012.

[64] Z. Kurt-Yavuz and S. Yavuz, "A comparison of EKF, UKF, FastSLAM2.0, and UKF-based FastSLAM algorithms," *Proceedings of the IEEE International Conference on Intelligent Engineering Systems*, pp. 37-43, 2012.

[65] R. Havangi and H. D. Taghirad, "A Square Root Unscented FastSLAM With Improved Proposal Distribution and Resampling," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2334-2345, 2014.

[66] X. Yan, C. Zhao, and J. Xiao, "A novel FastSLAM algorithm based on Iterated Unscented Kalman Filter," *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 1906–1911, 2011.

[67] Y. Song, Q. L. Li, and Y. F. Kan, "Conjugate Unscented FastSLAM for Autonomous Mobile Robots in Large-Scale Environments," *Cognitive Computation*, vol. 6, pp. 496-509, 2014.

[68] C. Qiu, X. Zhu, and X. Zhao "Vision-based unscented FastSLAM for mobile robot," *Proceedings of the 10th World Congress on Intelligent Control and Automation*, China, 2012.

[69] L. Liu and X. Zhu, "Vision-based Semantic Unscented FastSLAM for Mobile Robot," *Proceeding of the 11th World Congress on Intelligent Control and Automation*, China, 2014.

[70] G. Lee, N. Y. Chong, and H. Christensen, "Adaptive triangular mesh generation of self-configuring robot swarms," *Proceedings of IEEE International Conference on Robotics and Automation*, Kobe, Japan, pp. 2737-2742.34, 2009.

[71] G. Lee and N. Y. Chong, "Self-configurable Mobile Robot Swarms with Hole Repair Capability," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp. 1403-1408, 2008.

[72] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 201-208, 2006.

[73] L. Carlone, M. K. Ng, J. Du, B. Bona, and M. Indri, "Rao-Blackwellized Particle Filters Multi Robot SLAM with Unknown Initial Correspondences and Limited Communication," *Proceedings of IEEE International*

Conference on Robotics and Automation, Alaska, pp. 243-249, 2010.

[74] T. A. Vidal-Calleja, C. Berger and S. Lacroix, "Event-Driven Loop Closure in Multi-Robot Mapping," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, pp. 1535-1540, 2009.

[75] T. A. Vidal-Calleja, C. Berger and S. Lacroix, "Large Scale Multiple Robot Visual Mapping with Heterogeneous Landmarks in Semi-structured Terrain," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 654-674, 2011.

[76] S. H. Lee, W. S. Yoo, and B. H. Lee, "Rao-Blackwellized Particle Filter based Efficient CSLAM Framework," *Proceedings of Korea Robotics Society Annual Conference*, Buyeo, Korea, pp. 333-336, 2014. (In Korean)

[77] S. H. Lee, W. S. Yoo, and B. H. Lee, "A Network based Multi-agent SLAM Framework for Accurate Map-Building," *Proceedings of Electronics and Information Engineers Conference*, Jeju, Korea, pp. 1829-1832, 2014. (In Korean)

[78] C. Yunfei. T. Zhenmin, and Z. Chunxia, "Multi-robots cooperative online FastSLAM," *Chinese Journal of Electronics*, vol. 20, no. 2, pp. 223-227, 2011.

[79] S. H. Lee and B. H. Lee, "Kalman Consensus Based Multi-robot SLAM

with A Rao-Blackwellized Particle Filter," *Proceedings of the International Conference on Robotics and Computer Vision (ICRCV 2014)*, Beijing, China, Oct. 2014.

[80] http://www.personal.acfr.usyd.edu.au/tbailey/software/slam_simulations.h tm.

[81] E. Nebot, Victoria Park dataset, Website, http://www-personal.acfr.usyd.edu.au/nebot/d ataset.htm, 2008.

[82] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Communications of the ACM*, vol. 15, pp. 11-15, 1972.

[83] Y. Choi, T. Lee, and S. Oh, "A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot," Autonomous Robots, vol. 24, pp. 13-27, 2008.

[84] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," *Proceedings of International Conference on Computer Vision*, pp. 2548-2555, 2011.

[85] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," *Proceedings of European Conference on Computer Vision (ECCV)*, 2010.

[86] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, "Evaluating the

Efficiency of Frontier-based Exploration Strategies," *Proceedings of International Symposium on Robotics and 6th German Conference on Robotics 2010*, 2010.

[87] SLAM summer school. http://www.cas.kth.se/SLAM/, 2002.

[88] S. Carpin, "Fast and accurate map merging for multi-robot system," *Autonomous Robots*, vol. 25, no. 3, pp. 305-316, 2008.

[89] H. C. Lee, Y. J. Cho, and B. H. Lee, "Accurate map merging with virtual emphasis for multi-robot systems," *Electronics Letters*, vol. 49, no. 15, pp. 952-953, 2013.

[90] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[91] K. Y. K. Leung, T. D. Barfoot, H. H. T. Liu, "Decentralized Cooperative SLAM for Sparsely-Communicating Robot Networks: A Centralized-Equivalent Approach," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 66, no. 3, pp. 321-342, 2012.

[92] S. M. Chen, J. F. Yuan, F. Zhang, and H.J. Fang, "Multirobot FastSLAM Algorithm Based on Landmark Consistency Correction," *Mathematical Problems in Engineering*, vol. 2014, pp. 1-7, 2014.

[93] R. Sharma, C. Taylor, D. W. Casbeer, and R. W. Beard, "Distributed

Cooperative SLAM using an Information Consensus Filter," *Proceedings of AIAA Guidance Navigation and Control Conference*, Toronto, pp. 8334–8342, 2010.

[94] N. Kwak, B. H. Lee, and K. Yokoi, "Representation of the Results from Rao-Blackwellized Particle Filtering for SLAM," *Proceedings of International Conference on Control, Automation and Systems*, Seoul, Republic of Korea, pp. 698-703, 2008.

[95] R. Martinez-Cantin and J. A. Castellanos, "Unscented SLAM for Large-Scale Outdoor Environments," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 328-333, 2005.

[96] G. Grisetti, R. Kummerle, C. Stachniss, W. Burgard, "A Tutorial on Graph-Based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31-43, 2010.

초 록

미지의(unknown) 환경에서 광물탐사 또는 인명구조를 위해 기본적으로 다중로봇 SLAM 기술이 필요하다. 이 기술은 여러대의 로봇들이 주변 환경을 인식하고 동시에 자신들의 위치를 추정할 수 있는 능력을 의미한다. 먼저 SLAM(Simultaneous Localization and Mapping)을 위한 방법에는 다양한 방법이 있지만, 계산량 측면 및 추정 능력 측면에서 RBPF(Rao-Blackwellized Particle Filter) 기반의 SLAM 기술이 탁월하다. 이 기술은 SLAM 사후 확률(posterior probability)을 로봇의 위치 추정에 대한 확률과 지도 추정에 대한 확률로 인수분해(factorization)한다. 각각의 확률은 파티클 필터(particle filter)와 최적 필터(optimal filter)를 이용하여 추정이 가능하다.

본 논문에서는 파티클 간 협업 및 로봇 간 협업을 통해 협조적 RBPF-SLAM 프레임워크를 구축하였다. 파티클 간 협업은 RBPF-SLAM 과정 중 가중치 연산 및 리샘플링 파트에 활용이 되었다. 데이터 연관(data association)의 실패로 파티클들에 잘못 할당된 가중치는 파티클 간 기하학적 정보와 믿을 만한 가중치(reliable weight)정보를 이용하여 보정하였다. 기존 RBPF-SLAM의 문제점인 파티클 고갈 문제(particle depletion problem)는 파티클들이 자신의 가중치와 기하학적 정보를 이용하여 대형을 이루어 해결하였다. 이 때 기존 파티클들의 분포를 최대한 유지시키기 위해 삼각대형(triangular formation) 기법이 이용되었다. 로봇 간 협업은 RBPF-SLAM 과정 중 샘플링 및 측정치 갱신 파트에 활용이 되었다. 로봇과 로봇이 만나는 경우(rendezvous event), 각 로봇은 서로의 상대적 거리 및 각도를 측정할 수 있다. 이 측정치와 서로의 위치 정보를 공유하여 자신의 위치 정보를 보정할 수 있다. 공유된 정보를 융합하기 위해, 칼만 일치 필터(Kalman consensus filter)가 이용되었고, 이 과정은 RBPF-SLAM 프레임워크에 포함되었다. 또한 로봇과 로봇이 공통의 특징점(common

feature)을 공유하는 경우, 현재 자신이 추정한 정보와 다른 로봇에 의해 추정된 정보를 융합할 수 있고, 이 또한 칼만 일치 필터를 이용하여 적용이 가능하다. 본 논문에서는 공식 시뮬레이터를 이용한 다양한 시뮬레이션을 수행하였고, Car park dataset, Victoria park dataset, indoor dataset 그리고 outdoor dataset와 같은 실제 환경에도 적용하였다. 동시에 기존 기술인 FastSLAM 2.0, PSO (particle swarm optimization) 기반 FastSLAM, UFastSLAM, 파티클 분열(fission) 기반 UFastSLAM, PSO 기반 UFastSLAM 방법등과 비교하여 제안된 RBPF-SLAM 프레임워크의 SLAM 성능을 검증하였다.