Ph.D. DISSERTATION

Correlation-based Map Merging for Multi-Robot SLAM

다중 로봇 SLAM을 위한 상관관계 기반 지도병합 기술

BY

HEON-CHEOL LEE

AUGUST 2013

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

Abstracts

Multi-robot simultaneous localization and mapping (SLAM) is an advanced technique used by multiple robots and autonomous vehicles to build up a collective map within an unknown environment, or to update a collective map within a known environment, while at the same time keeping track of their current location. The collective map is obtained by merging individual maps built by different multiple robots exploring the environment. When robots do not know their initial poses one another, the problem of map merging becomes challenging because the robots have different coordinate systems. If robot-to-robot measurements are not available, the problem of map merging becomes more challenging because the map transformation matrix (MTM) among robots cannot be computed directly.

This dissertation presents novel map merging techniques based on the analysis of the correlation among the individual maps, which do not need the knowledge of the relative initial poses of robots and the robot-to-robot measurements. After the cross-correlation function among the spectrometric or tomographic information extracted from the individual maps is generated, the MTM is computed by taking the rotation angle and the translation amounts corresponding to the maximum cross-correlation values.

The correlation-based map merging techniques with spectral information presented in this dissertation are the extensions of a conventional map merging technique. One extension is spectrum-based feature map merging (SFMM), which extracts the spectral information of feature maps from virtual supporting lines and computes the MTM by matching the extracted spectral information. The other extension is enhanced-spectrum-based map merging (ESMM), which enhances grid maps using the locations of visual objects and computes the MTM by matching the spectral information extracted from the enhanced grid maps. The two extensions overcome successfully the limitation of the conventional map merging technique. The correlation-based map merging technique with tomographic information is a new map merging technique, which is named tomographic map merging (TMM). Since tomographic analysis can provide more detailed information on grid maps according to rotation and translation than spectral analysis, the more accurate MTM can be computed by matching the tomographic information. The TMM was tested on various pairs of partial maps from real experiments in indoor and outdoor environments. The improved accuracy was verified by showing smaller map merging errors than the conventional map merging technique and several existing map merging techniques.

Keywords: Map merging, Multi-robot SLAM, Correlation, Spectrometry, Tomography

Student Number: 2008-30882

Contents

List of Figures

List of Tables

Nomenclature

Notation	Description	Chapter
$\mathbf{M}_i$	The grid or feature map of the *i*-th robot	2,3,4
$\mathbf{T}_{\mathbf{M}_{ij}}$	Map transformation matrix (MTM) between $\mathbf{M}_i$ and $\mathbf{M}_j$	2,3,4
Δ_θ	Map rotation angle	2,3,4
Δ_x	Map translation amount with respect to *x*-coordinate	2,3,4
Δ_y	Map translation amount with respect to *y*-coordinate	2,3,4
$\omega_a(\mathbf{M}_i, \mathbf{M}_j)$	The acceptance index between two grid maps, $\mathbf{M}_i$ and $\mathbf{M}_j$	2
$\mathbf{M}_{HT}$	The grid map transformed by the Hough transform	3
$\mathbf{HS}_{\mathbf{M}_i}$	The Hough spectrum extracted from a grid map $\mathbf{M}_i$	3
$\mathbf{XS}_{\mathbf{M}_i}$	The *X*-spectrum extracted from a grid map $\mathbf{M}_i$	3,4
$\mathbf{YS}_{\mathbf{M}_i}$	The *Y*-spectrum extracted from a grid map $\mathbf{M}_i$	3,4
$\mathbf{CC}_{\mathbf{M}_i\mathbf{M}_j}$	The cross correlation between $\mathbf{HS}_{\mathbf{M}_i}$ and $\mathbf{HS}_{\mathbf{M}_j}$	3
$\mathbf{CCX}_{\mathbf{M}_i\mathbf{M}_j}$	The cross correlation between $\mathbf{XS}_{\mathbf{M}_i}$ and $\mathbf{XS}_{\mathbf{M}_j}$	3,4
$\mathbf{CCY}_{\mathbf{M}_i\mathbf{M}_j}$	The cross correlation between $\mathbf{YS}_{\mathbf{M}_i}$ and $\mathbf{YS}_{\mathbf{M}_j}$	3,4
$\omega_f(\mathbf{M}_i, \mathbf{M}_j)$	The acceptance index between two feature maps, $\mathbf{M}_i$ and $\mathbf{M}_j$	3
$\mathbf{\Lambda}_i$	The *i*-th virtual supporting line (VSL) of a feature map	3
$\mathbf{M}_i^*$	The feature map updated by VSLs of the *i*-th robot	3

$\mathbf{M}_{HT}^{*}$	The updated feature map transformed by the Hough transform	3
$\widehat{\mathbf{M}}_{i}^{*}$	The updated and rasterized feature map of the *i*-th robot	3
$\mathbf{HS}_{\mathbf{M}_i}^{f}$	The Hough spectrum extracted from a feature map $\mathbf{M}_i$	3
$\mathbf{XS}_{\mathbf{M}_i}^{f}$	The *X*-spectrum extracted from a feature map $\mathbf{M}_i$	3
$\mathbf{YS}_{\mathbf{M}_i}^{f}$	The *Y*-spectrum extracted from a feature map $\mathbf{M}_i$	3
$\mathbf{CC}_{\mathbf{M}_i\mathbf{M}_j}^{f}$	The cross correlation between $\mathbf{HS}_{\mathbf{M}_i}^{f}$ and $\mathbf{HS}_{\mathbf{M}_j}^{f}$	3
$\mathbf{CCX}_{\mathbf{M}_i\mathbf{M}_j}^{f}$	The cross correlation between $\mathbf{XS}_{\mathbf{M}_i}^{f}$ and $\mathbf{XS}_{\mathbf{M}_j}^{f}$	3
$\mathbf{CCY}_{\mathbf{M}_i\mathbf{M}_j}^{f}$	The cross correlation between $\mathbf{YS}_{\mathbf{M}_i}^{f}$ and $\mathbf{YS}_{\mathbf{M}_j}^{f}$	3
$\mathbf{M}_{i}^{d}$	The dual form of the map of the *i*-th robot	3
$\omega_E(\mathbf{M}_i,\mathbf{M}_j)$	The map merging error between two maps, $\mathbf{M}_i$ and $\mathbf{M}_j$	3
$\mathbf{M}_{i}^{eh}$	The enhanced map of the *i*-th robot	3
$\widetilde{\mathbf{M}}_{i}^{eh}$	The rotated enhanced map of the *i*-th robot	3
$\mathbf{I}_{i}^{eh}$	The image matrix generated from $\mathbf{M}_{i}^{eh}$	3
$\omega_a^d(\mathbf{M}_i,\mathbf{M}_j)$	The dual acceptance index between two maps, $\mathbf{M}_i$ and $\mathbf{M}_j$	4
$\mathbf{S}_{\mathbf{M}_i}$	The sinogram extracted by the Radon transform from $\mathbf{M}_i$	4
$\widetilde{\mathbf{M}}_i$, $\widetilde{\mathbf{M}}_i^{d}$	The rotated map and its dual form of the *i*-th robot	4
$\mathbf{DS}_{\mathbf{M}_i}^{\alpha'}$	The directional spectrum of $\mathbf{M}_i$ for the α' value of $\mathbf{S}_{\mathbf{M}_i}$	4

Chapter 1

Introduction

1.1 Background and Motivation

Simultaneous localization and mapping (SLAM) [2] is to build a map in unknown environments (*mapping*) and to simultaneously estimate their own pose in the coordinate system of the built map (*localization*). However, a robot should know its own pose to build a map of its surrounding environment, and the robot should be given a map to estimate its own pose. Due to the dependency between mapping and localization, SLAM has been considered as a *chicken or egg problem*. Moreover, because robot motion is affected by drift or slippage, and its perception has limitations such as corrupted noises and limited sensing ranges, it is highly difficult to conduct SLAM efficiently. Various techniques have been developed to overcome the difficulties and improve the performance of SLAM, which can be classified according to several criteria such as frameworks, sensors and the number of robots. First, the researches on SLAM frameworks are well-established, and many variants of EKF-SLAM [7] and FastSLAM [23-26, 61-62] have been developed. Because they have their own advantages and disadvantages, it is difficult to say which framework is more efficient than other frameworks. This dissertation has been conducted with the framework of FastSLAM, which will be described in Chapter 2. Second, the researches on sensor-based implementation techniques for SLAM have been developed. The two main categories are Scan-SLAM [45, 48] which uses range sensors such as ultrasonic range finders and laser range finders and Visual-SLAM [20, 46] which uses vision sensors such as monocular cameras, stereo cameras and infrared cameras. The key issue in Scan-SLAM is how well the scans obtained from range sensors at different time steps are matched. There are a variety of scan matching techniques such as ICP [51-53], IDC [45, 48], HSM [54-55], PSM [31, 56] and CSM [34]. Visual-SLAM has been developed in the context of visual feature extraction and matching algorithms [18-20, 46, 58], place recognition [28] and semantic mapping techniques [71-72]. Finally, researchers have been recently encouraged to develop multi-robot SLAM [1, 17] which conducts SLAM more efficiently based on the cooperation among multiple robots [5, 6]. Multi-robot SLAM is also called cooperative-SLAM (C-SLAM) [4] because of operating the cooperative robots.

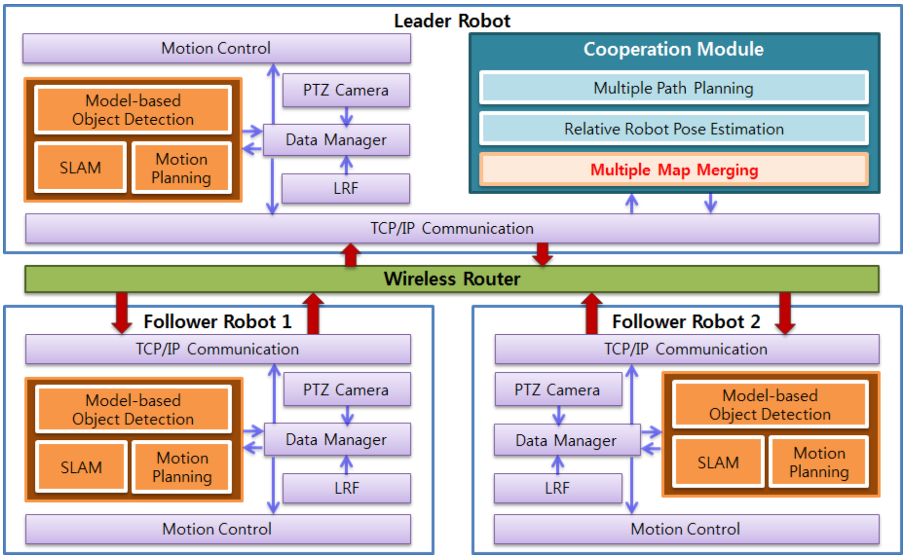


Figure 1.1 An example of the system for multi-robot SLAM. Each robot builds a map while estimating its own pose. In the cooperation module of the leader robot, multiple maps of the robots are merged into a collective map.

An example of the system for multi-robot SLAM can be configured as shown in Fig, which consists of various techniques. The most important component in multi-robot SLAM is map merging conducted in the cooperation module in the leader robot. Map merging in multi-robot SLAM is to fuse the individual maps built by the multiple robots to obtain a collective map [59, 64]. Since the performance of map merging affects the performance of the overall multi-robot system, it is important to develop an efficient map merging technique. For example, when two robots perform tasks, a robot can obtain the information on unexplored areas by merging the other robot map into its own map. But, if the merged map is not accurate, the robot cannot perform its own task efficiently since the path planned in the inaccurate merged map is inefficient. Map merging becomes challenging when the robots do not know their relative initial poses one another because the map transformation between the robot maps is derived from the relative poses between the robots [1, 3, 10, 12]. Practically, since the relative poses are not accurately given in most of real multi-robot systems, the development of map merging techniques should be conducted under unknown initial poses among robots. This dissertation also deals with the challenging map merging approaches under unknown initial poses for multi-robot systems.

1.2 Related Works

Map merging is to produce a collective map from multiple individual maps based on the map transformation matrix (MTM) between them. The methods which have been developed to acquire the MTM can be categorized into two groups according to the availability of the direct information on the relative poses among robots. One group is to directly acquire the MTM by measuring the relative distance and angle among the robots with vision and range sensors, which can be performed accidentally or intentionally on the predetermined rendezvous areas [41]. Konolige et al. [11] proposed a map merging technique based on the concept of hypothesis-and-verification. For hypothesis, robots try to meet one another with robot-to-robot measurements. If they fail to meet one another at estimated locations, the hypothesis is rejected. If they succeed, the hypothesis is accepted, which means that their individual maps are merged. However, they did not describe the more detailed description of acquiring the robot-to-robot measurements. Zhou et al. [12] proposed a map merging technique based on omnidirectional vision sensors to measure the relative distance and angle. Based on the measurements, the optimal geometry to compute the MTM was well established. However, in their work, the predetermined rendezvous areas were required. Lee et al. [1] proposed a probabilistic framework to solve the problem of map merging for multi-robot SLAM using particle filters was proposed. Especially, they focused on the bases for map merging, which represent the poses of the robots. The most probable map merging bases were obtained by Gaussian processes, which were used for map merging with the robot-to-robot measurement obtained by fusing range data by a laser scan sensor and visual data by a camera. However, it also required predetermined rendezvous areas, and the quality of the measurements depended on the environmental conditions such as color and lighting effects. Consequently, the direct map merging techniques may cause the additional costs to establish the sensor systems to obtain the robot-to-robot measurements or highly depend on the quality of the robot-to-robot measurements.

The other group is to indirectly acquire the MTM by finding and matching the overlapping regions of the individual maps without any external systems for the relative poses among robots. Zhou et al. [12] applied the nearest neighbor test (NNT) to matching the individual feature maps of robots to update the MTM which is initially computed by robot-to-robot measurements. Although the NNT was helpful to update the initially merged map, it caused much more computation time to conduct the NNT. Howard [3] proposed the concept of the virtual robot traveling backwards in time to build a complete map. The virtual robots traveled backward in time and update iteratively the past information to update the initially merged map produced in the forward process in time. However, inaccurate past data could cause inaccurate map merging. Differently from the above works, Carpin [9] proposed a map merging technique based on the spectral information of individual maps which tackled the conventional iterative process that may cause much computation time and the risk of being caught on local minima or maxima. The rotation angle was estimated by matching the Hough spectra of robot maps, and the translation displacements were estimated by matching *X* and *Y* spectra of robot maps. His technique reduced significantly computation time but required the sufficient overlapping areas between individual maps, which may decrease the flexibility of using multiple robots. Except for the above well-known indirect map merging techniques, there are several map merging techniques which are difficult to be categorized into the two groups. The techniques are respectively based on topological information [14, 16], manifold representations [15], sparse information filters [17] and Robot Technology Middleware (RTM) [60].

1.3 Contributions

This dissertation presents novel map merging techniques which do not need both the knowledge of the relative initial poses among robots and the direct robot-to-robot measurements. Firstly, this dissertation presents two extensions of the spectrum-based map merging (SMM) which is a recently developed grid map merging technique.

- The SMM cannot be applied to feature map merging due to its own properties. This dissertation presents a useful extension of the SMM, which is a non-iterative feature map merging technique using virtual supporting lines (VSLs). The presented extension extracts the spectral information of multiple feature maps using VSLs and obtains the MTM using the circular cross-correlation between the extracted spectral information on feature maps. The proposed technique was tested on feature maps produced by experiments with vision sensors, which was performed non-iteratively. In addition, it consistently showed high acceptance indices which indicate the degree of accuracy of feature map merging.
- The SMM requires a sufficient amount of overlapping regions between individual maps to find an accurate MTM. If the overlapping regions are insufficient, the performance of map merging degenerates. Common features or landmarks in individual maps can significantly reduce the required overlapping region. This dissertation presents a

novel extension of the SMM, which can be performed with smaller overlapping areas using visual objects. The more accurate MTM can be computed by matching the spectra of maps enhanced by the normal distribution with the locations of the visual objects. In environments where map merging using only geometric spectra or only visual landmarks fails, the presented extension finds the MTM more accurately without being caught on local maxima.

Secondly, this dissertation presents a new grid map merging technique, which does not require any information on the initial poses among robots or any robot-to-robot measurements.

- If the common visual features or landmarks are not available to reduce the size of the overlapping regions, the most important issue in map merging is to compute the accurate MTM with relatively small overlapping regions. This dissertation presents a novel map merging technique based on tomographic analysis. Because tomographic analysis can provide more detailed information on grid maps according to rotation and translation than spectral analysis, the more accurate MTM can be computed by matching the tomographic information. The tomographic information is extracted by an integral transform consisting of the integral of a function over straight lines such as Radon transform. The tomographic map merging technique (TMM) was tested on various different pairs of partial maps from real experiments in both indoor and outdoor environments. The improved accuracy was verified by showing the smaller map merging errors than the SMM. Especially, the TMM was robustly capable of merging maps despite inevitable map errors in outdoor experiments. Besides, the TMM was applicable to not only merging two maps but also more maps in successive rounds.

1.4 Organization

This dissertation is organized as follows. Chapter 2 describes the fundamental formulations for multi-robot SLAM and the map merging problem. Chapter 3 presents the extensions of the conventional map merging technique, which are spectrum-based feature map merging (SFMM) using virtual supporting lines and enhanced-spectrum-based map merging (ESMM). Chapter 4 presents a new map merging technique, which is named tomographic map merging (TMM). Chapter 5 shows the experimental results of the SFMM, the ESMM and the TMM in various environments. Finally, the conclusions of this dissertation are given in Chapter 6.

Chapter 2

Multi-Robot SLAM and Map Merging

In this section, the frameworks of SLAM using particle filters and multi-robot SLAM (MR-SLAM) using particle filters, which are used in this dissertation, are briefly described. According to the knowledge about the relative poses among robots, the framework of MR-SLAM is classified into MR-SLAM with known initial correspondences and MR-SLAM with unknown initial correspondences. Next, the formulations for map merging which is the most important component in MR-SLAM are described. According to the availability of direct observation measurements among robots, the formulations are classified into map merging with robot-to-robot measurements and map merging with overlapping area matching.

2.1 SLAM using Particle Filters

Due to the uncertainty of robot motion and sensor measurements, SLAM [7] have been approached within various probabilistic frameworks based on estimation theories. The posterior over a robot pose and a map with known data association by

$$p(x_{1:t}, \mathbf{M} \mid z_{1:t}, u_{1:t}, c_{1:t}) \qquad (2.1)$$

where $x_{1:t}$ is the robot path till time t, **M** is the map, and $z_{1:t}$, $u_{1:t}$ and $c_{1:t}$ are the measurements, controls, correspondences up to time t, respectively. This posterior has been mainly estimated by Kalman filters and particle filters. When Kalman filters are applied to SLAM, the computational cost for a joint map covariance matrix increases quadratically, which is a crucial drawback for MR-SLAM which requires the relatively large-scale map. In particle filters, a sample of a posterior distribution, which is frequently called a particle, represents a hypothesis as to what the true world state may be at that time. The intuition behind particle filters is to approximate the posterior by the set of particles. Unfortunately, particle filters may be also affected by the size of a map. A straightforward implementation of particle filters for SLAM would absolutely fail in a large environment due to the large number of variables representing the map.

However, particle filters can be more efficiently applied to SLAM using characteristics of the SLAM problem which can be approached by conditional independence between two disjoint sets of a robot pose and a map [7]. Rao-

Blackwellized particle filter (RBPF) which is a specialized version of the particle filter enables particle filters to be applicable to SLAM which uses a set of features as a map as shown in Fig. 2.1. FastSLAM [2] is an instance of SLAM based on RBPF, which uses particle filters to estimate the robot pose. The estimation of a map can be factored into many separate low-dimensional extended Kalman filters (EKFs) or unscented Kalman filters (UKFs) using the conditional independence among the features in the map.

The key mathematical insight of FastSLAM is based on the fact that the full SLAM posterior can be factorized with two parts which are respectively for robot path estimation and map estimation when the correspondences, $c_{1:t} = c_1, \cdots, c_t$ are given [2]:

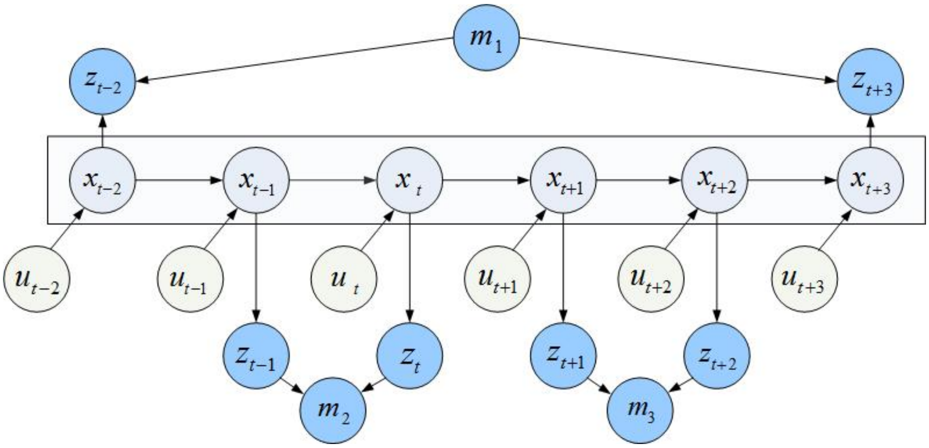


Figure 2.1 Factorization of the SLAM problem. If the consecutive poses of a robot from x_{t-2} to x_{t+3} are given, the locations of the features m_1, m_2 and m_3 are conditionally independent where z_t is a measurement at time t. This Bayes network graph indicates that the pose variables for the robot can separate the individual features in the map.

$$\begin{aligned} p(x_{1:t}, M \mid z_{1:t}, u_{1:t}, c_{1:t}) &= p(x_{1:t} \mid z_{1:t}, u_{1:t}, c_{1:t})\, p(M \mid x_{1:t}, z_{1:t}, u_{1:t}, c_{1:t}) \\ &= p(x_{1:t} \mid z_{1:t}, u_{1:t}, c_{1:t}) \prod_{n=1}^{N_f} p(m_n \mid x_{1:t}, z_{1:t}, u_{1:t}, c_{1:t}) \end{aligned} \tag{2.2}$$

where m_n is the *n*-th feature in the map **M**, and N_f is the number of features. In the framework of FastSLAM, the posterior over robot paths are estimated by RBPFs, which is denoted by $p(x_{1:t}|z_{1:t}, u_{1:t}, c_{1:t})$. The locations of the features in **M** are separately estimated by EKF or UKF, which is denoted by $\prod_{n=1}^{N_f} p(m_n|x_{1:t}, z_{1:t}, u_{1:t}, c_{1:t})$. The estimators for feature locations are conditioned on the robot path, which means there is a separate copy of each feature estimator for each particle. Due to this factorization, FastSLAM can maintain a separate EKF or UKF for each feature, which makes the update more efficient than that in EKF-SLAM. A particle at time *t*, $Y_t^{[k]}$ in FastSLAM is denoted by

$$Y_t^{[k]} = \left\langle x_t^{[k]}, \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]}, \dots, \mu_{N_f,t}^{[k]}, \Sigma_{N_f,t}^{[k]} \right\rangle \tag{2.3}$$

where the [*k*] indicates the index of the particle, and $x_t^{[k]}$ is the pose estimate of the robot at time t. Only the most recent pose $x_t^{[k]}$ is used in FastSLAM, so a particle keeps only the most recent pose. $\mu_{n,t}^{[k]}, \Sigma_{n,t}^{[k]}$ are the mean and covariance representing the *n*-th feature location relative to the *k*-th particle. Because the RBPF used in FastSLAM has the problem of particle depletion, we have developed several extension of the standard FastSLAM to overcome the problem. The detailed procedures of the standard FastSLAM and our previous extensions are respectively described in [2] and [39, 40].

2.2 Multi-Robot SLAM (MR-SLAM)

MR-SLAM is formulated by extending the framework of SLAM with particle filters to deal with multiple robots. Because the knowledge about the initial poses among robots enables the robots to share their coordinate systems, MR-SLAM is differently formulated as the availability of the knowledge. In this dissertation, the formulation is divided into MR-SLAM with known initial correspondences and MR-SLAM with unknown initial correspondences.

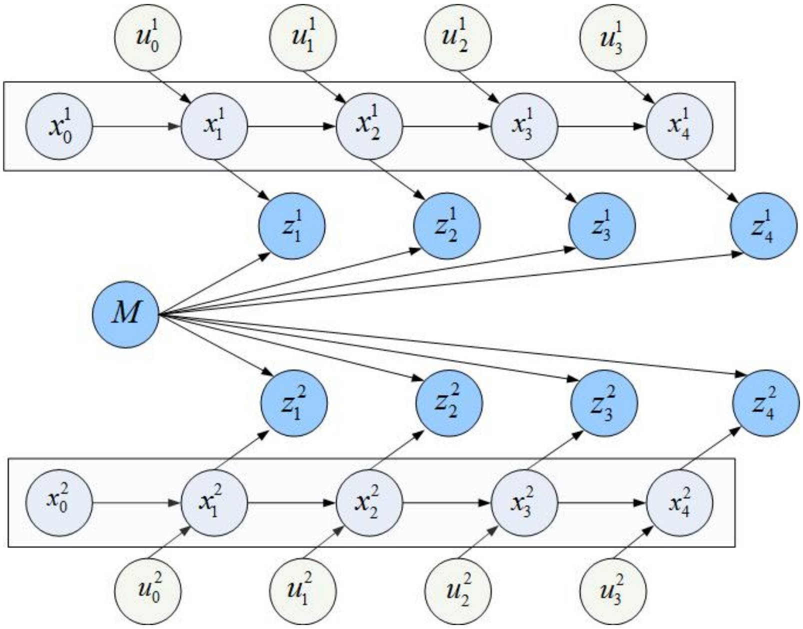


Figure 2.2 Bayes network for MR-SLAM with known initial correspondence, which means the relative initial poses between two robots, x_0^1 and x_0^2, are known. The trajectories of two robots are indicated by sequences $(x_0^1, x_1^1, \cdots)$ and $(x_0^2, x_1^2, \cdots)$, respectively. Here, any dependencies between the observation $(z_1^2, z_2^2, \cdots)$ made by robot 2 and the pose $(x_1^1, x_2^1, \cdots)$ of robot 1 (and vice versa) are ignored.

2.2.1 MR-SLAM with Known Initial Correspondence

If the initial robot correspondence is known, the maps which have built by robots can be readily merged into a collective map. Therefore, the formulation for MR-SLAM is achieved by simply extending the formulation for a single-robot SLAM. The graphical structure of MR-SLAM with known initial correspondence [3] is shown in Fig. 2.2. In the literature [3], the posterior probability of MR-SLAM with known initial correspondence based on two robots can be analyzed as follows:

$$\begin{aligned} &P(x_{1:t}^1, x_{1:t}^2, \mathbf{M} \mid z_{1:t}^1, u_{0:t-1}^1, x_0^1, z_{1:t}^2, u_{0:t-1}^2, x_0^2) = \\ &\qquad P(\mathbf{M} \mid x_{1:t}^1, z_{1:t}^1, x_{1:t}^2, z_{1:t}^2) \\ &\qquad P(x_{1:t}^1 \mid z_{1:t}^1, u_{0:t-1}^1, x_0^1) P(x_{1:t}^2 \mid z_{1:t}^2, u_{0:t-1}^2, x_0^2) \end{aligned} \tag{2.4}$$

where robot i is denoted by R_i, and **M** is a collective map. x_0^i is the initial pose for R_i. $x_{1:t}^i$ is the trajectory for R_i, at times 1, 2, … , t. $z_{1:t}^i$ is a corresponding sequence of observations, and $u_{0:t-1}^i$ is the sequence of actions executed by R_i. To efficiently estimate the posterior, the conditional dependency between the robot trajectories is ignored. Then, since the robot trajectories are treated as independent variables, each term of the posterior probability can be independently estimated. Thus, the full posterior can be obtained by using the estimated terms. Particle filters are used to compute the posterior probability over the trajectory of a robot, and a separate estimator is used over the current location of the robot for each feature in **M**. In MR-SLAM using RBPFs, a particle of R_i at time t is denoted by

$$Y_t^{i,[k]} = \left\langle x_t^{i,[k]}, \mu_{1,t}^{i,[k]}, \Sigma_{1,t}^{i,[k]}, \ldots, \mu_{N_f,t}^{i,[k]}, \Sigma_{N_f,t}^{i,[k]} \right\rangle \tag{2.5}$$

where the $[k]$ indicates the index of the particle, and $x_t^{i,[k]}$ is the pose estimate of the robot *i* at time *t*. Because only the most recent pose $x_t^{i,[k]}$ is used in MR-SLAM using RBPFs, each particle keeps only the most recent pose. $\mu_{n,t}^{i,[k]}$ and $\Sigma_{n,t}^{i,[k]}$ are mean and covariance of the Gaussian, which represent the probabilistic location of the *n*-th feature relative to the *k*-th particle. Using the set of the particles, each robot can simultaneously localize its own pose and build its own map using its own particle set. Consequently, the individual maps are easily merged because robots know their relative initial poses one another.

2.2.2 MR-SLAM with Unknown Initial Correspondence

If the initial robot correspondence is unknown, the formulation for MR-SLAM should carefully consider the relative position and orientation among the coordinate systems of the robots. The graphical structure of MR-SLAM with unknown initial correspondence [1, 3] is shown in Fig. 2.3. The posterior probability of MR-SLAM with unknown initial correspondence based on two robots [3] can be analyzed as follows:

$$\begin{aligned} P(x_{1:t}^1, x_{1:t}^2, \mathbf{M} \mid z_{1:t}^1, u_{0:t-1}^1, x_0^1, z_{1:t}^2, u_{0:t-1}^2, \Delta_s^{21}) = \\ P(\mathbf{M} \mid x_{1:t}^1, z_{1:t}^1, x_{1:s-1}^2, z_{1:s-1}^2, x_{s+1:t}^2, z_{s+1:t}^2) \\ P(x_{1:t}^1 \mid z_{1:t}^1, u_{0:t-1}^1, x_0^1) P(x_{1:s-1}^2 \mid z_{1:s-1}^2, u_{0:s-1}^2, x_s^1, \Delta_s^{21}) \\ P(x_{s+1:t}^2 \mid z_{s+1:t}^2, u_{s:t-1}^2, x_s^1, \Delta_s^{21}) \end{aligned} \tag{2.6}$$

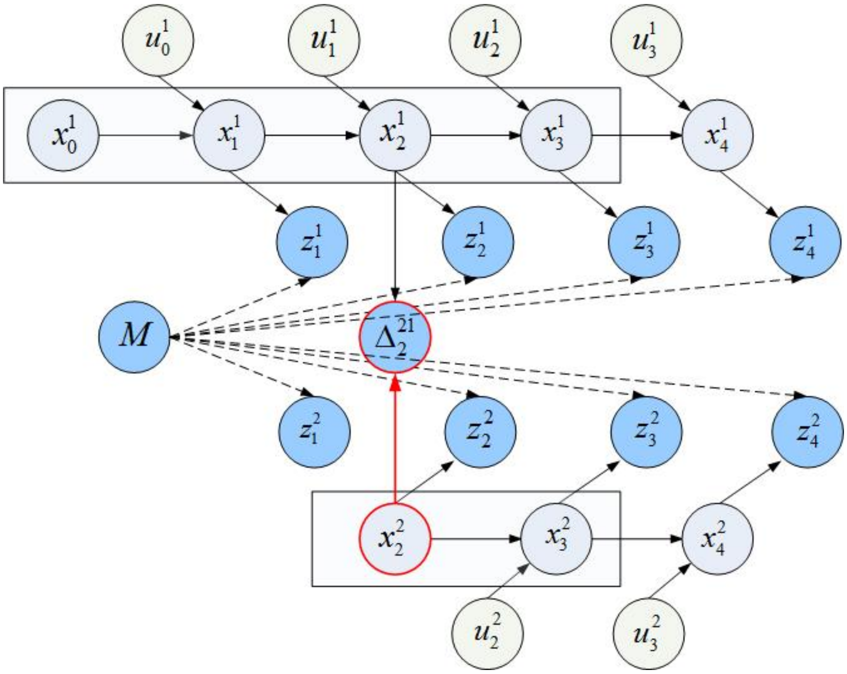


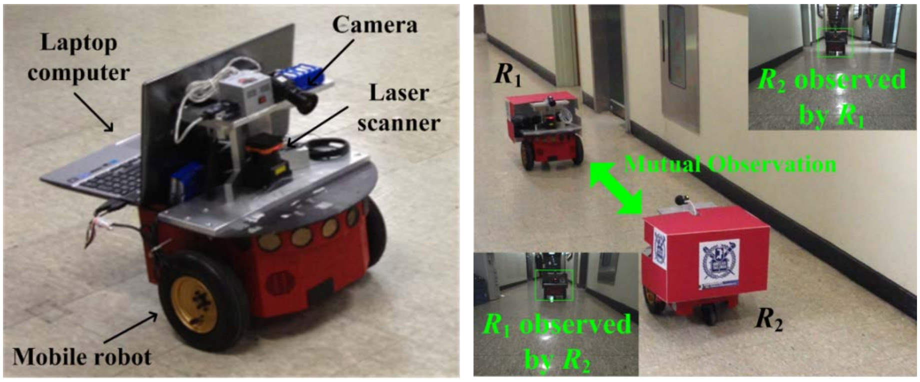
Figure 2.3 Bayes network for MR-SLAM with unknown initial correspondence. The robots first encounter each other at time t=2 and measure their relative pose Δ_2^{21}, (the pose of robot 2 relative to robot 1 at time t=2). Using the relative pose, robot 2's map is merged into robot 1's map.

where robot i is denoted by R_i, and **M** is a map. Δ_s^{21} is the relative pose between R_1 and R_2 at the time s. $x_{k:t}^i$ is the trajectory for R_i, at times k, k+1, … , t. $z_{k:t}^i$ is a corresponding sequence of observations, and $u_{k-1:t-1}^i$ is the sequence of actions executed by R_i.

Similarly to MR-SLAM with known initial correspondence, for efficient estimation of the posterior, we have assumed that the conditional dependency between the robot trajectories is ignored. Then, the robot trajectories are treated as independent variables, and each term of the posterior probability can be independently estimated. Consequently, the full posterior can be obtained by using the estimated terms. Using (2.6), each robot computes its own posterior probability using RBPFs. However, map merging in MR-SLAM with unknown initial correspondences becomes highly challenging because the coordinate transformation between the robots is initially unknown. In this case, map merging can be conducted when the robots encounter each other or the sufficient amount of the overlapping area between the individual maps of the robots is detected.

2.3 Map Merging

Map merging is to obtain a collective map by merging the individual maps of the different robots exploring the different parts of an environment. The most important part of the map merging is to compute the MTM among the robots. When the initial correspondences among the robots is known for one another as described in Section 2.2.1, the MTM can be easily computed because the robots can share the coordinate system by simply computing the difference among the initial poses of the robots. However, when the initial correspondences among robots is unknown for one another as described in Section 2.2.2, the computation of the MTM becomes challenging because the robots cannot share the coordinate system. This dissertation deals with only map merging with unknown initial correspondences.



(a) Sensor configuration of each robot

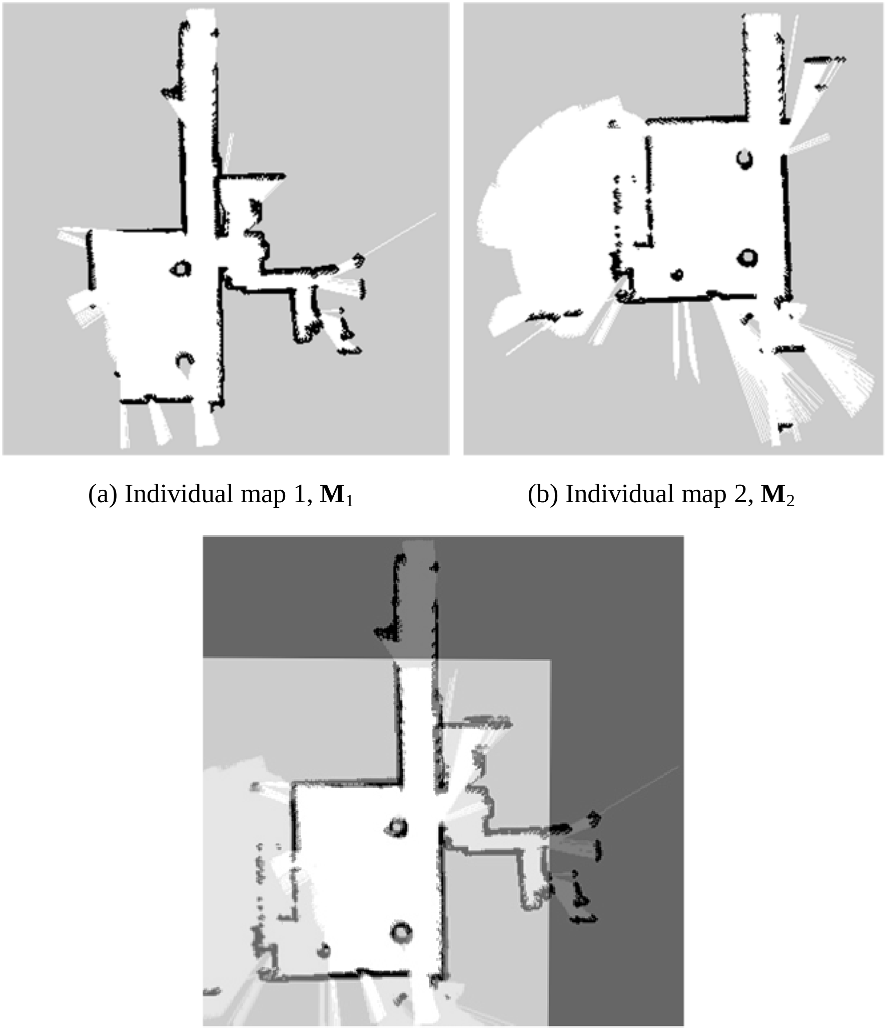
(b) Mutual observation

Figure 2.4 An example of a multi-robot system with robot-to-robot observation measurements for map merging. (a) Each robot is equipped with sensors such as a camera and a laser scanner to perceive another robot. (b) When two robots, R_1 and R_2, encounter each other, the mutual observation measurements are acquired.

Under unknown initial correspondences among robots, the approaches to computing MTM are classified into two categories. The first category is to measure the relative position and orientation between robots when the robots encounter each other or meet at predetermined places. Thus, these approaches require not only additional hardware such as range sensors and vision sensors but also the corresponding algorithms to process the sensor data. The relative position between robot positions can be measured by fusing the data obtained from a range sensor and a vision sensor [1, 12].

Figure 2.4 shows an example of a multi-robot system with robot-to-robot measurements for map merging. The perception of another robot can be conducted by applying local feature extraction algorithms such as SIFT [18] and SURF [19] to classification algorithms like support vector machine (SVM). But, the quality of estimating the orientation of the perceived robot is not enough to be used as a measurement. Thus, one-way observation measurements are difficult to be directly applied to map merging. Zhou et al. [12] presented a multi-robot system using mutual observation measurements which consist of only mutual relative positions instead of the orientation of at least one of the robots. Their work has relaxed the limitation of requiring the high-quality perception processing which may cause much computational cost in multi-robot systems. However, the approaches in the first category have been suffered from the inevitable errors in observation measurements due to the uncertainty of sensor models. Moreover, the assumption that two robots should meet each other at the same time to obtain the mutual observation measurements may decline the flexibility of operating multi-robot systems as shown in Fig. 2.4(b). Therefore, to resolve the two limitations, the approaches in the first category should be helped by the other techniques.

The second category is to compute the MTM by finding and matching the overlapping areas among the individual maps of robots, which can be not only applied to map merging with the techniques in the first category to improve the accuracy but also applied solely to map merging. Because the purpose of this dissertation is to improve the accuracy of map merging without any loss of the flexibility of operating multi-robot systems, this dissertation focuses on the second category.



(a) Individual map 1, $\mathbf{M}_1$

(b) Individual map 2, $\mathbf{M}_2$

(c) Merged map

Figure 2.5 An example of merging two grid maps by matching the overlapping areas between them. In each map, the black, white and gray cells represent respectively occupied, unoccupied and unknown grids. Since the overlapping areas are sufficient, $\mathbf{M}_1$ and $\mathbf{M}_2$ are successfully merged.

An example of merging two grid maps by one of the techniques in the second category is shown in Fig. 2.5. The overlapping areas are autonomously found and matched to compute the MTM between the two maps. The general formulation of the MTM which is the key for map merging is as follows. Given two-dimensional maps, $\mathbf{M}_1$ and $\mathbf{M}_2$, which are feature maps or grid maps, the MTM which translates Δ_x in the direction of *x*-coordinate and Δ_y in the direction of *y*-coordinate, and rotates Δ_θ in a counter-clockwise is defined as follows:

$$\mathbf{T}(\Delta_x, \Delta_y, \Delta_\theta) = \begin{bmatrix} \cos\Delta_\theta & -\sin\Delta_\theta & \Delta_x \\ \sin\Delta_\theta & \cos\Delta_\theta & \Delta_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

Let $\mathbf{m}_{1,i}$ and $\mathbf{m}_{2,j}$ be the *i*-th element in $\mathbf{M}_1$ and the *j*-th element in $\mathbf{M}_2$, respectively, which are represented by 3×1 vectors which consist of two rows representing the *x*-*y* coordinates of features or occupancy grids and a redundant constant that is usually set to 1. The transformation of $\mathbf{m}_{2,j}$ can be computed as follows:

$$\mathbf{m}'_{2,j} = \mathbf{T}(\Delta_x, \Delta_y, \Delta_\theta) \cdot \mathbf{m}_{2,j} \tag{2.8}$$

Then, $\mathbf{m}'_{2,j}$ is represented in the same coordinate system with $\mathbf{m}_{1,i}$. Consequently, the two maps share the same coordinate systems. This general formulation can be modified according to the type and dimension of the maps.

The classical approach to matching two maps is to find the MTM which maximizes the similarity between them in the iterative manner. As a measure of the similarity, Birk et al. [13] proposed an acceptance index which is computed as follows:

$$\omega_a(\mathbf{M}_1,\mathbf{M}_2)=\begin{cases}0 & \text{if } agr(\mathbf{M}_1,\mathbf{M}_2)=0\\ \dfrac{agr(\mathbf{M}_1,\mathbf{M}_2)}{agr(\mathbf{M}_1,\mathbf{M}_2)+dis(\mathbf{M}_1,\mathbf{M}_2)} & \text{if } agr(\mathbf{M}_1,\mathbf{M}_2)\neq 0\end{cases} \tag{2.9}$$

where $agr(\mathbf{M}_1,\mathbf{M}_2)$ and $dis(\mathbf{M}_1,\mathbf{M}_2)$ represent the agreement and disagreement between $\mathbf{M}_1$ and $\mathbf{M}_2$, respectively. $agr(\mathbf{M}_1,\mathbf{M}_2)$ is computed by the number of cells in $\mathbf{M}_1$ and $\mathbf{M}_2$ that are both free or both occupied, and $dis(\mathbf{M}_1,\mathbf{M}_2)$ is computed by the number of cells such that $\mathbf{M}_1$ is free and $\mathbf{M}_2$ is occupied or vice-versa.

According to the definition of the conventional acceptance index, $\mathbf{M}_1$ and $\mathbf{M}_2$ should have the same size in the same coordinate system. However, this is a strong assumption in real multi-robot systems, especially with unknown initial relative poses between the multiple robots. To utilize the acceptance index, the size of each map and the coordinates of each grid in the map should be carefully reconfigured to follow the definition of the acceptance index. Therefore, the formulation of the acceptance index can be modified according to the type and dimension of the maps. If the overlapping areas are sufficient as shown in Fig. 2.5, the conventional acceptance index can be a reliable measure. But, if the overlapping areas are not sufficient, the conventional acceptance index tends to fail to indicate the similarity between maps due to the excessive amount of outliers for matching them.

In this case, the similarity should be measured by an alternative index, which can be applied to the classical approaches based on the iterative manner. Otherwise, the MTM can be also computed without the similarity by tackling the iterative manner [8, 9]. Then, the similarity is used as a performance index which indicates the error in map merging. The similarity like the acceptance index can be also used as a supplementary measure to select the best MTM among several candidate MTMs. Because this dissertation does not follow the iterative manner either, the similarity does not become the crucial factor for map matching. Instead, the map merging techniques in this dissertation use their own indices to measure the error in map merging, which can be a modified version of the conventional acceptance index or not.

Chapter 3

Map Merging based on Spectral Correlation

In this section, after the spectrum-based map merging (SMM) technique [9] which is the conventional correlation-based map merging technique is briefly described, its two extensions are presented. The SMM is a nice approach to grid map merging based on the analysis of spectral information on maps. However, it has two limitations: non-applicability to feature map merging and need for a sufficient amount of overlapping regions. The first extension is capable of resolving the first limitation using virtual supporting lines (VSLs) generated from the geometric relation of feature. The second extension is capable of resolving the second limitation by enhancing the spectral information based on the locations of common visual objects.

3.1 Spectrum-based Map Merging (SMM)

Carpin [9] proposed the spectra-based map merging (SMM) algorithm to overcome the iterative property of the conventional map merging algorithms. In his work, the map merging problem was considered as the binary image registration problem. The SMM consists of two phases: the estimation of the rotation angle and the estimation of the displacements between two individual maps. Firstly, the rotation angle, Δ_θ, is estimated by matching the Hough spectra extracted from the individual maps with Hough transform [21, 22]. The standard Hough transform focused on the detection of lines which consists of consecutive occupied grids, which can be represented in Hough domains, i.e. $x\cos\theta + y\sin\theta = \rho$. Here, θ is the angle between x-axis and the normal from the line to the origin, and ρ is the shortest distance between the line and the origin. Let $\mathbf{M}_{HT}$ be the transformed map by Hough transform. $\mathbf{M}_{HT}$ is a matrix with ρ_g rows and θ_g columns, and its associated Hough spectrum for $1 \le j \le \theta_g$ is defined as the following unidimensional signal:

$$\mathbf{HS}_{\mathbf{M}}(j) = \sum_{i=1}^{\rho_g} \mathbf{M}_{HT}(i,j)^2 \tag{3.1}$$

The signal has sampling period with $2\pi/\theta_g$ and indicates how frequently lines are detected along θ. Given Hough spectra, $\mathbf{HS}_{\mathbf{M}_1}$ and $\mathbf{HS}_{\mathbf{M}_2}$, circular cross correlation is computed to determine similarities along θ as follows:

$$\mathbf{CC}_{\mathbf{M}_1\mathbf{M}_2}(k) = \sum_{i=1}^{\theta_g} \mathbf{HS}_{\mathbf{M}_1}(i)\mathbf{HS}_{\mathbf{M}_2}(i+k) \tag{3.2}$$

Then, $\Delta_\theta = \mathrm{argmax}_k\, \mathbf{CC}_{\mathbf{M}_1\mathbf{M}_2}(k)$, where $1 \le k \le \theta_g$.

Next, x and y displacements, Δ_x and Δ_y, are estimated by matching *X*-spectra and *Y*-spectra, respectively. Let a grid map $\mathbf{M}$ be considered as a binary image with r rows and c columns, *X*-spectrum for $1 \leq j \leq c$ and *Y*-spectrum for $1 \leq i \leq r$ are respectively defined as the following signals:

$$\mathbf{XS}_{\mathbf{M}}(j) = \sum_{i=1}^{r} \mathbf{M}(i, j) \tag{3.3}$$

$$\mathbf{YS}_{\mathbf{M}}(i) = \sum_{j=1}^{c} \mathbf{M}(i, j) \tag{3.4}$$

Based on the estimated rotation angle, Δ_θ, $\widetilde{\mathbf{M}}_2 = \mathrm{T}(0,0,\Delta_\theta)\mathbf{M}_2$. Similarly to $\mathbf{CC}_{\mathbf{M}_1\mathbf{M}_2}$, circular cross correlations are respectively computed to determine *X* and *Y* displacements as follows:

$$\mathbf{CCX}_{\mathbf{M}_1\tilde{\mathbf{M}}_2}(\tau) = \sum_{j=-\infty}^{+\infty} \mathbf{XS}_{\mathbf{M}_1}(j+\tau)\mathbf{XS}_{\tilde{\mathbf{M}}_2}(j) \tag{3.5}$$

$$\mathbf{CCY}_{\mathbf{M}_1\tilde{\mathbf{M}}_2}(\nu) = \sum_{i=-\infty}^{+\infty} \mathbf{YS}_{\mathbf{M}_1}(i+\nu)\mathbf{YS}_{\tilde{\mathbf{M}}_2}(i) \tag{3.6}$$

Then, $\Delta_x = \mathrm{argmax}_\tau\, \mathbf{CCX}_{\mathbf{M}_1\widetilde{\mathbf{M}}_2}(\tau)$, and $\Delta_y = \mathrm{argmax}_\upsilon\, \mathbf{CCY}_{\mathbf{M}_1\widetilde{\mathbf{M}}_2}(\upsilon)$.

Thanks to the non-iterative property for matching the overlapping area between individual maps, the SMM improved the speed of grid map merging significantly, which enables real-time processes. However, the SMM has two limitations, and this dissertation presents two extensions of the SMM to resolve the limitations. In the following subchapters, the limitations are briefly described, and the corresponding extensions are described respectively.

3.2 Spectrum-based Feature Map Merging (SFMM)

The SMM is a nice grid map merging technique but cannot be applied to feature map merging due to its own properties. This dissertation presents a useful extension of the SMM, which is a non-iterative feature map merging technique using virtual supporting lines (VSLs). The extension extracts the spectral information of multiple feature maps using VSLs and obtains the MTM using the circular cross-correlation between the extracted spectral information of the multiple feature maps. The proposed technique was tested on feature maps produced by experiments with vision sensors, which was performed non-iteratively. Besides, it consistently showed a high acceptance index, which indicates the degree of accuracy for feature map merging.

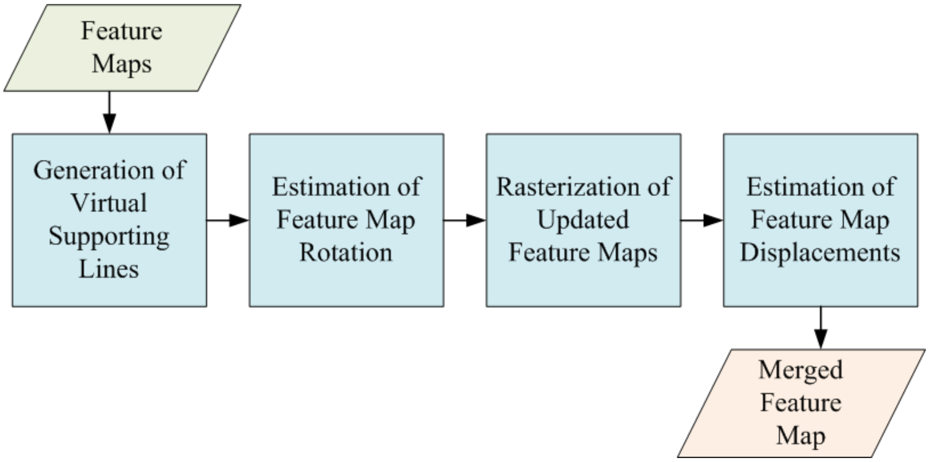

Feature Maps
Generation of Virtual Supporting Lines
Estimation of Feature Map Rotation
Rasterization of Updated Feature Maps
Estimation of Feature Map Displacements
Merged Feature Map

Figure 3.1 Whole structure of the SFMM. A merged feature map is obtained based on the MTM which consists of the estimated rotation angle and displacement amounts among the feature maps by the SFMM.

3.2.1 Overview of the SFMM

The SFMM[1] is an accurate feature map merging technique based on spectral correlation, which can be performed with indistinguishable features. The whole structure of the SFMM is shown in Fig. 3.1. The key process is the generation of virtual supporting lines (VSLs) from the original feature maps. Based on the updated feature maps with VSLs, the map rotation angle is firstly estimated by finding the maximum spectral correlation. Next, the feature maps are rasterized to extract the translational spectral information. Then, the map displacements are estimated with the rasterized feature maps. Finally, a collective merged feature map is obtained by the MTM which consists of the estimated rotation angle and displacement amounts.

3.2.2 Problem Formulation for the SFMM

A two-dimensional feature map $\mathbf{M}$ which consists of N_f features can be mathematically represented by a matrix with three rows and N_f columns. The first and second rows of $\mathbf{M}$ indicate the x-coordinate and y-coordinate of each feature, respectively. For the convenient computation of the MTM, the last row is filled with 1 in the SFMM. Given two feature maps, $\mathbf{M}_1$ and $\mathbf{M}_2$, the goal of feature map merging is to find the MTM, $\mathbf{T}_{\mathbf{M}_{12}}$, which enables $\mathbf{M}_1$ and $\mathbf{M}_2$ to be maximally overlapped. $\mathbf{T}_{\mathbf{M}_{12}}$ which translates Δ_x in the direction

[1] The SFMM which is one of the contributions in this dissertation has been published. Please refer to [8] for more detailed description.

of *x*-coordinate and Δ_y in the direction of *y*-coordinate, and rotates Δ_θ in a counter-clockwise is defined as follows:

$$\mathbf{T}_{\mathbf{M}_{12}}(\Delta_x,\Delta_y,\Delta_\theta)=\begin{bmatrix}\cos\Delta_\theta & -\sin\Delta_\theta & \Delta_x\\ \sin\Delta_\theta & \cos\Delta_\theta & \Delta_y\\ 0 & 0 & 1\end{bmatrix} \tag{3.7}$$

where $\mathbf{M_2}=\mathbf{T}_{\mathbf{M}_{12}}\mathbf{M_1}$. Practically, the computation of the optimal $\mathbf{T}_{\mathbf{M}_{12}}$ is difficult due to the inevitable errors in the estimation of the feature locations caused by imperfect sensors. Conventionally, many researchers have used iterative methods to find the MTM with the minimal mismatch between $\mathbf{M}_1$ and $\mathbf{M}_2$. In the case of grid map merging, to measure the mismatch between them, Birk et al. [13] proposed an acceptance index defined by the concept of the agreement and disagreement between $\mathbf{M}_1$ and $\mathbf{M}_2$, which has been widely used in grid map merging.

In the case of feature map merging, however, the acceptance index should be modified because a feature map consists of sparsely distributed points differently from a grid map. In this dissertation, the acceptance index to measure how two feature maps are overlapped is defined using the Euclidean distance between the N_{cf} common features in $\mathbf{M}_1$ and $\mathbf{M}_2$ as follows:

$$\omega_f(\mathbf{M}_1,\mathbf{M}_2)=1-\frac{\sum_{n=1}^{N_{cf}}\|\,\mathbf{m}_{n,1}-\mathbf{m}_{n,2}\,\|}{\Gamma_f N_{cf}} \tag{3.8}$$

where $\|\cdot\|$ is Euclidean norm. $\mathbf{m}_{n,1}$ and $\mathbf{m}_{n,2}$ are the *n*-th common feature in

$\mathbf{M}_1$ and $\mathbf{M}_2$, respectively. Γ_f is an user-defined constant to normalize the error in feature map merging. If the two feature maps are matched perfectly, $\omega_f(\mathbf{M}_1, \mathbf{M}_2)$ becomes 1. If the total difference between the locations of the common features is excessively larger than $\Gamma_f N_{cf}$, $\omega_f(\mathbf{M}_1, \mathbf{M}_2)$ becomes a negative value. If there are several possible MTMs, the best MTM for feature map merging can be obtained by

$$\mathbf{T}_{\mathbf{M}_{12}}^{f} = \arg\max_{\mathbf{T}_i} \omega_f(\mathbf{M}_1, \mathbf{T}_i\mathbf{M}_2) \quad , i = 1, \cdots, N_T \tag{3.9}$$

Note that the iterative property to check the acceptance index and obtain the MTM still remains, which causes the slowdown in feature map merging.

3.2.3 Virtual Supporting Lines (VSLs)

Differently from a grid map, since a feature map consists of the limited number of data points to extract the spectral information, this dissertation proposes a new supplementary method to extract the spectral information on a feature map, *virtual supporting line*, which is generated from the geometric relation among point features. Naive point-to-point matching methods with the common features between two feature maps tend to fail to find an accurate MTM due to the lack of the information on feature maps. The key idea of the proposed method is to combine a feature and its neighbor features together to find the more information on a feature map and connect it with the spectra-based approach in the SMM to improve the accuracy of feature map merging.

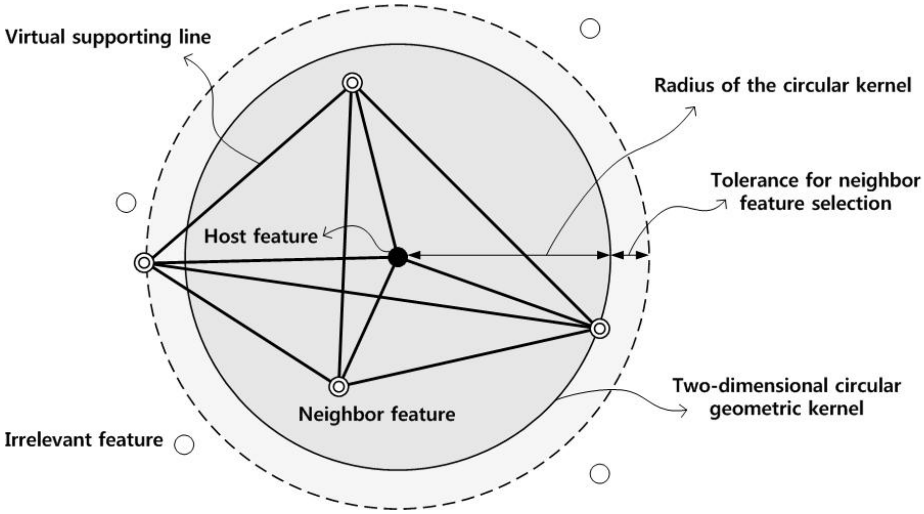

Virtual supporting line
Radius of the circular kernel
Tolerance for neighbor feature selection
Host feature
Two-dimensional circular geometric kernel
Neighbor feature
Irrelevant feature

Figure 3.2 The concept of virtual supporting lines. The neighbor features are selected by circular geometric kernel centered at the host feature. Virtual supporting lines are generated among the host feature and all the selected neighbor features.

Virtual supporting line (VSL) is a virtually generated one-dimensional connection between two neighboring features to enrich the feature map with undistinguishable points, which should meet the following requirements for feature map matching. First, a couple of the neighboring features should be closely located because the combination of the more distant features may cause the more errors in feature map matching. Second, the relative distance and angle between the neighboring features should be consistent regardless of map rotation or displacements since the inconsistent relative distance and angle may degenerate the correspondence for map matching. Finally, the combination of the neighboring features in a map built by a robot should be robustly associated with the same combination in maps built by other robots. As shown in Fig. 3.2, VSLs are generated by 2D circular geometric kernels to find neighbor features from a host feature to meet the first requirement. The selected features are connected to meet the second requirement. A tolerance factor for neighbor feature selection is used to meet the final requirement.

Given a feature map $\mathbf{M} = \{\mathbf{m}_1, \cdots, \mathbf{m}_n, \cdots, \mathbf{m}_N\}$ where N is the number of features, every feature becomes a host feature $\mathbf{m}_n^h$, and its neighbor feature $\mathbf{m}_n^r$ is extracted by the following circular geometric kernel with a tolerance κ.

$$K(\mathbf{m}_n^h, \mathbf{m}_n^r, \kappa) = \sqrt{\| \mathbf{m}_n^h - \mathbf{m}_n^r \| - \kappa^2} \tag{3.10}$$

Consequently, the set of the n-th host feature and the corresponding neighbor features are obtained as follows:

$$\begin{aligned} \mathbf{\Psi}_n &= \{\mathbf{m}_n^h\} \cup \{\mathbf{m}_n^r \mid K(\mathbf{m}_n^h, \mathbf{m}_n^r, \kappa) \le R\} \\ &= \{\psi_1, \cdots, \psi_i, \cdots, \psi_j, \cdots, \psi_{N_{f,n}}\} \end{aligned} \tag{3.11}$$

where R is the radius of the circular geometric kernel. VSLs are virtually generated by simply linking all the elements in $\mathbf{\Psi}_n$, which can be denoted as follows:

$$\begin{aligned} \mathbf{\Lambda}_n &= \{(\psi_i, \psi_j) \mid \psi_i \in \mathbf{\Psi}_n, \psi_j \in \mathbf{\Psi}_n\} \\ &= \{\lambda_1^n, \lambda_2^n, \cdots, \lambda_k^n, \cdots, \lambda_{K_n}^n\} \end{aligned} \tag{3.12}$$

where K_n is the number of VSLs with the n-th host feature. The k-th VSL which consists of (ψ_i, ψ_j) can be denoted by the following vector form of the closed line segment:

$$\lambda_k^n = \{\psi_i + c\psi_j \mid c \in [0,1]\} \tag{3.13}$$

Finally, **M** is updated as $\mathbf{\Lambda} = \{\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \cdots, \mathbf{\Lambda}_n, \cdots, \mathbf{\Lambda}_{N_f}\}$ which are generated by the algorithm shown in Table 3.1. Consequently, **M** consists of several line segments instead of point features.

TABLE 3.1

ALGORITHM FOR THE GENERATION OF VIRTUAL SUPPORTING LINES

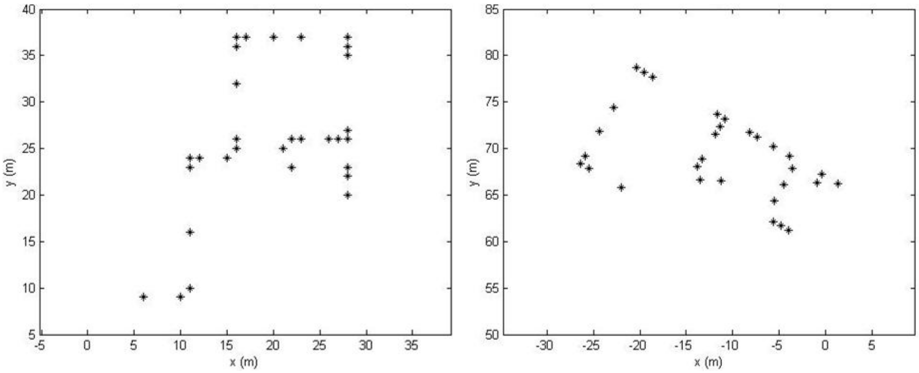
```
Algorithm 1:  The generation of virtual supporting lines
Input:        A feature map, $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_n, \cdots, \mathbf{m}_{N_f}\}$
Output:       The set of virtual supporting lines, $\mathbf{\Lambda} = \{\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \cdots, \mathbf{\Lambda}_n, \cdots, \mathbf{\Lambda}_{N_f}\}$

$\mathbf{\Lambda} \leftarrow \emptyset$
Set the radius of the circular geometric kernel, $R$
Set the tolerance of the circular geometric kernel, $\kappa$
for $n$=1 to $N_f$ do
    for $l$=1 to $N_f$ do
        $\mathbf{m}_n^h \leftarrow$ the $n$-th feature in M as a host feature
        $\mathbf{m}_n^{r*} \leftarrow$ the $l$-th feature in M as a candidate neighbor feature
        if $K(\mathbf{m}_n^h, \mathbf{m}_n^{r*}, \kappa) \leq R$, then
            $\mathbf{m}_n^r \leftarrow \mathbf{m}_n^{r*}$ as the selected neighbor feature by (3.11)
            $\mathbf{\Lambda}_n \leftarrow \mathbf{\Lambda}_n \cup (\mathbf{m}_n^h, \mathbf{m}_n^r)$ by (3.12)
        end if
    end for
    $\mathbf{\Lambda} \leftarrow \mathbf{\Lambda}_n$
end for
return $\mathbf{\Lambda}$
```

3.2.4 Estimation of Map Rotation with Hough Spectra

The rotation angle between two feature maps updated by VSLs, $\mathbf{M}_1^*$ and $\mathbf{M}_2^*$, is estimated with Hough spectra. First, the discretized Hough transform (DHT) is performed to detect the line segments in $\mathbf{M}_1^*$ and $\mathbf{M}_2^*$. Then, the geometric information in the Cartesian space is transformed into the spectral information in the Hough space, i.e. $x\cos\theta + y\sin\theta = \rho$, and discretized for ρ and θ. Therefore, the transformed feature maps, $\mathbf{M}_{1,HT}^*$ and $\mathbf{M}_{2,HT}^*$, are respectively represented by matrices with ρ_f rows and θ_f columns. The equation (3.1) is accordingly rewritten by the following signals with sampling period of $2\pi/\theta_f$:

$$\mathbf{HS}_{\mathbf{M}_1^*}^{f}(j) = \sum_{i=1}^{\rho_g} \mathbf{M}_{1,HT}^{*}(i,j)^2 \tag{3.14}$$

$$\mathbf{HS}_{\mathbf{M}_2^*}^{f}(j) = \sum_{i=1}^{\rho_g} \mathbf{M}_{2,HT}^{*}(i,j)^2 \tag{3.15}$$

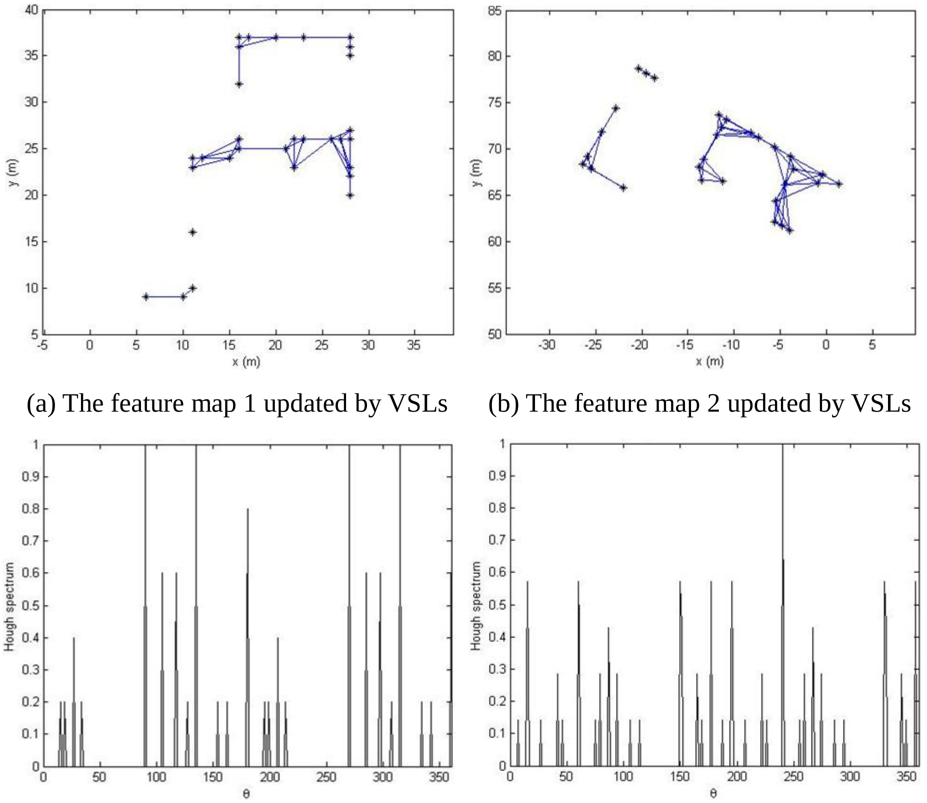


(a) The original feature map of robot 1 (b) The original feature map of robot 2

Figure 3.3 The original feature maps which have the same size with 25m×35m.

where $1 \leq j \leq \theta_f$. For example, two original feature maps shown in Fig. 3.3 are updated by VSLs as shown in Fig. 3.4(a)-(b). The corresponding Hough spectra extracted from the updated feature maps are shown in Fig. 3.4(c)-(d).

The circular cross correlation between $\mathbf{HS}_{\mathbf{M}_1^*}^{f}$ and $\mathbf{HS}_{\mathbf{M}_2^*}^{f}$ can measure how much they overlap each other. Thus, the circular cross correlation is computed to find the map rotation angle along θ as follows:



(a) The feature map 1 updated by VSLs (b) The feature map 2 updated by VSLs

(c) Hough spectrum of the feature map 1 (d) Hough spectrum of the feature map 2

Figure 3.4 The Hough spectra of the feature maps updated by VSLs generated with *R*=5. The Hough spectra indicate the spectral information of the feature maps.

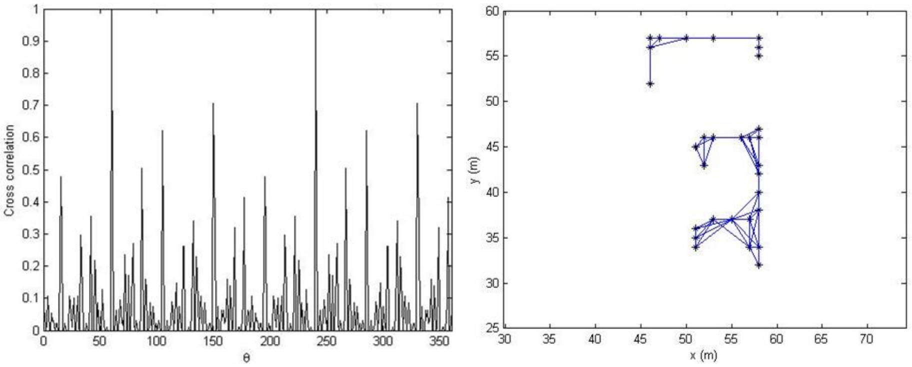
$$\mathbf{CC}^{f}_{\mathbf{M}_1^*\mathbf{M}_2^*}(k) = \sum_{i=1}^{\theta_f} \mathbf{HS}^{f}_{\mathbf{M}_1^*}(i)\mathbf{HS}^{f}_{\mathbf{M}_2^*}(i+k) \tag{3.16}$$

Then, the rotation angle between the two feature maps is estimated by taking the angle corresponding to the maximum cross-correlation as follows:

$$\Delta^{f}_{\theta} = \arg\max_{k} \mathbf{CC}^{f}_{\mathbf{M}_1^*\mathbf{M}_2^*}(k) \tag{3.17}$$

where $1 \le k \le \theta_f$.

The detailed algorithm to estimate the rotation angle between the two feature maps updated by VSLs is summarized in Table 3.2. The computed circular cross correlation is shown in Fig. 3.5(a), and the estimated rotation angle is 60°. The result of rotating the feature map 2 updated by VSLs is shown in Fig. 3.5(b).



(a) Cross correlation between Hough spectra (b) The rotated feature map 2

Figure 3.5 The result of the circular cross correlation between the two feature maps. Since the maximum correlation appears with the map rotation angle $\Delta^{f}_{\theta} = 60°$, the feature map 2 updated by VSLs is rotated by 60º.

TABLE 3.2

ALGORITHM FOR ESTIMATING THE ROTATION BETWEEN TWO FEATURE MAPS

```
Algorithm 2: The computation of the rotation between the two feature maps
Input:       Feature maps updated by VSLs, $\mathbf{M}_1^*$ and $\mathbf{M}_2^*$
Output:      The rotation angle between the feature maps, $\Delta_\theta^f$

$\Delta_\theta^f \leftarrow \emptyset$, and apply DHT to $\mathbf{M}_1^*$ and $\mathbf{M}_2^*$
for $j$=1 to $\theta_f$ do
    Compute the Hough spectrum of $\mathbf{M}_1^*$ by (3.14)
    Compute the Hough spectrum of $\mathbf{M}_2^*$ by (3.15)
end for
for $k$=1 to $\theta_f$ do
    Compute the correlation between $\mathbf{HS}_{\mathrm{M}_1^*}^f$ and $\mathbf{HS}_{\mathrm{M}_2^*}^f$ by (3.16)
    Compute the rotation, $\Delta_\theta^f$, maximizing the correlation (3.17)
end for
return $\Delta_\theta^f$
```

3.2.5 Rasterization of Updated Feature Maps with VSLs

For the estimation of x-y translations, one can consider the maximally voted distance r_{max}in order that $\Delta_x^f = r_{max}\cos\Delta_\theta^f$, and $\Delta_y^f = r_{max}\sin\Delta_\theta^f$. However, since the performance of the approach depends on the distribution and density of the features, the approach did not guarantee the robustness of feature map matching. Therefore, this dissertation utilizes not the approach but another approach like image registration methods. Before estimating *x*-*y* translations, Δ_x^f and Δ_y^f, the feature map updated by VSLs should be discretized to extract the spectral information on *x*-*y* coordinates, respectively.

If the updated feature map is regarded as a virtual image, the problem of discretization can be solved by rasterizing the virtual image. To rasterize the virtual image whose size is $S_{row} \times S_{col}$ with a fixed resolution η_r, the virtual image is converted by the discretized image with $\frac{S_{row}}{\eta_r} \times \frac{S_{col}}{\eta_r}$ cells.

For the k-th VSL, λ_k^n, generated from the n-th host feature in the updated feature map $\mathbf{M}^*$, each point on the λ_k^n is rounded off to the nearest quantized cell of the point as shown in Fig. 3.6. Therefore, the rasterized feature map $\widehat{\mathbf{M}}^*$ is represented by a matrix with $\hat{r}$ rows and $\hat{c}$ columns.

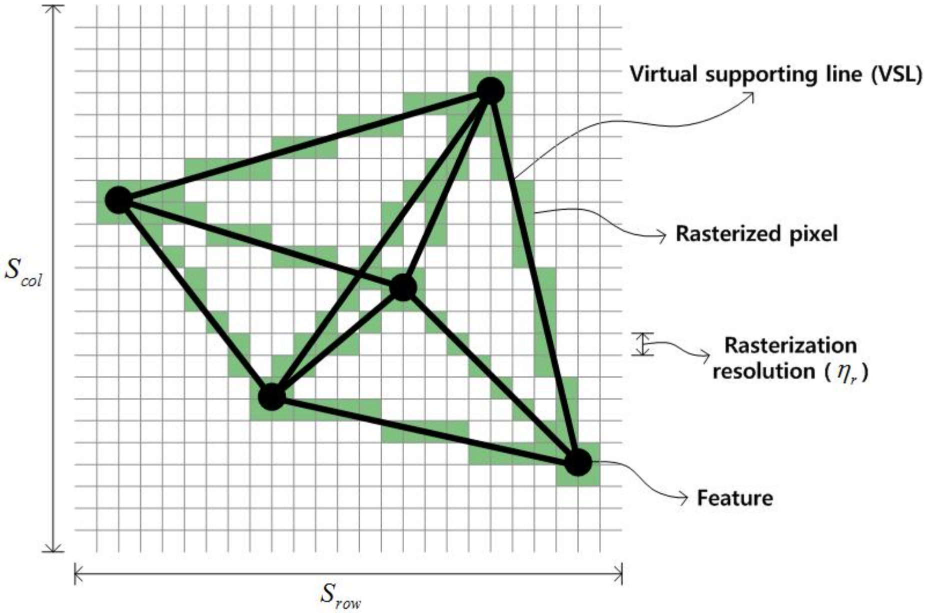

Virtual supporting line (VSL)
Rasterized pixel
Rasterization resolution (η_r)
Feature
S_{col}
S_{row}

Figure 3.6 Rasterization of the updated feature map with VSLs. The updated feature map regarded as a virtual image whose size is $S_{row} \times S_{col}$ is rasterized with a fixed resolution η_r.

TABLE 3.3

ALGORITHM FOR RASTERIZATION OF A UPDATED FEATURE MAP WITH VSLS

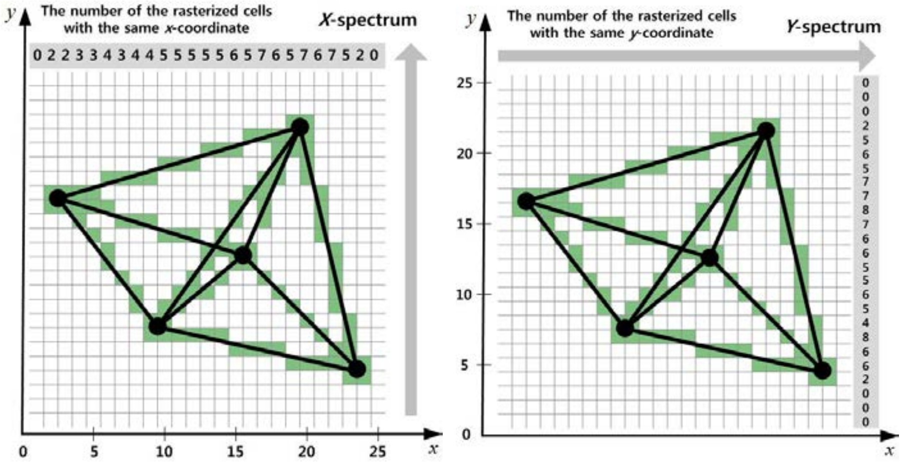
```
Algorithm 3: The rasterization of the updated feature map with VSLs
Input:       A updated feature map with VSLs, $\mathbf{M}^*$, whose size is $S_{row} \times S_{col}$
             A fixed resolution, $\eta_r$ and the total set of the VSLs, $\widehat{\mathbf{\Lambda}}$
Output:      The rasterized feature map, $\widehat{\mathbf{M}}^*$

$\widehat{\mathbf{M}}^* \leftarrow$ zero matrix with $\frac{S_{row}}{\eta_r}$ rows and $\frac{S_{col}}{\eta_r}$ columns
for $\hat{r} = 1$ to $\frac{S_{row}}{\eta_r}$ do
    for $\hat{c} = 1$ to $\frac{S_{col}}{\eta_r}$ do
        if $(\hat{r}, \hat{c}) \in \hat{\lambda}_k^n$ for $\hat{\lambda}_k^n \subset \widehat{\mathbf{\Lambda}}$, then
            $\widehat{\mathbf{M}}^*(\hat{r}, \hat{c}) = 1$ by (3.18)
        end if
    end for
end for
return $\widehat{\mathbf{M}}^*$
```

$$\widehat{\mathbf{M}}^*(\hat{r},\hat{c}) = \begin{cases} 1 & if\ (\hat{r},\hat{c}) \in \hat{\lambda}_k^n \quad for\ \hat{\lambda}_k^n \subset \widehat{\mathbf{\Lambda}} \\ 0 & otherwise \end{cases} \tag{3.18}$$

where $\widehat{\mathbf{\Lambda}}$ is the total set of the VSLs which are rounded off, and $\hat{\lambda}_k^n$ is its subset, and $n = 1, \cdots, N_f$, and $k = 1, \cdots, K_n$. The detailed algorithm for the rasterization of the updated feature map is summarized in Table 3.3.

3.2.6 Estimation of Map Displacements

Based on the map rotation angle Δ_θ^f estimated in Section 3.2.4 and the rasterized feature maps, $\widehat{\mathbf{M}}_1^*$ and $\widehat{\mathbf{M}}_2^*$, one can obtain $\widehat{\mathbf{M}}_3^* = \mathbf{T}(0,0,\Delta_\theta^f)\widehat{\mathbf{M}}_2^*$, which is the feature map rotated by Δ_θ^f. The map displacements are also estimated by correlating the spectral information on x-y coordinates, which extracted from the rasterized feature maps. The X-spectrum of the rasterized feature map is extracted by computing the number of the rasterized cells with the same x-coordinate as shown in Fig. 3.7(a). Similarly, the Y-spectrum of the rasterized feature map is extracted by computing the number of the rasterized cells with the same y-coordinate as shown in Fig. 3.7(b). If the rasterized feature map $\hat{\mathbf{M}}_1^*$ is considered as a binary image with $\hat{r}_1$ rows and $\hat{c}_1$ columns, X-spectrum for $1 \leq j \leq \hat{c}_1$ and Y-spectrum for $1 \leq i \leq \hat{r}_1$ are respectively computed by



(a) The extraction of the X-spectrum (b) The extraction of the Y-spectrum

Figure 3.7 The concept of extracting X-Y spectra. The X-spectrum and Y-spectrum of the rasterized feature map are extracted by computing the number of the rasterized cells with the same x-coordinate and the same y-coordinate, respectively.

$$\mathbf{XS}_{\widehat{\mathbf{M}}_1^*}(j) = \sum_{i=1}^{\hat{r}_1} \widehat{\mathbf{M}}_1^*(i, j) \tag{3.19}$$

$$\mathbf{YS}_{\widehat{\mathbf{M}}_1^*}(i) = \sum_{j=1}^{\hat{c}_1} \widehat{\mathbf{M}}_1^*(i, j) \tag{3.20}$$

Similarly, if the rasterized feature map $\widehat{\mathbf{M}}_3^*$ is considered as a binary image with $\hat{r}_3$ rows and $\hat{c}_3$ columns, *X*-spectrum for $1 \leq j \leq \hat{c}_3$ and *Y*-spectrum for $1 \leq i \leq \hat{r}_3$ are respectively computed as the following signals:

$$\mathbf{XS}_{\widehat{\mathbf{M}}_3^*}(j) = \sum_{i=1}^{\hat{r}_3} \widehat{\mathbf{M}}_3^*(i, j) \tag{3.21}$$

$$\mathbf{YS}_{\widehat{\mathbf{M}}_3^*}(i) = \sum_{j=1}^{\hat{c}_3} \widehat{\mathbf{M}}_3^*(i, j) \tag{3.22}$$

Figure 3.8 shows the *X*-spectra and the *Y*-spectra of the two feature maps. For $\hat{c}_1 > \hat{c}_3$ and $\hat{r}_1 > \hat{r}_3$, the circular cross correlation functions are respectively computed to find *X* and *Y* displacements as follows:

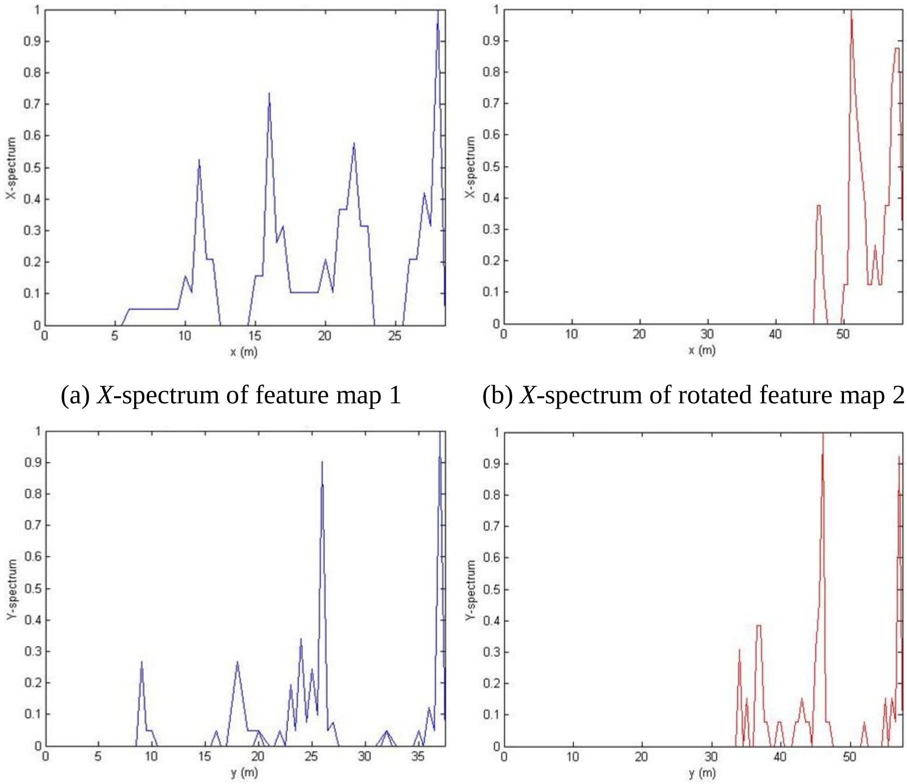
$$\mathbf{CCX}_{\widehat{\mathbf{M}}_1^* \widehat{\mathbf{M}}_3^*}(\tau) = \sum_{j=-\infty}^{+\infty} \mathbf{XS}_{\widehat{\mathbf{M}}_1^*}(j+\tau)\mathbf{XS}_{\widehat{\mathbf{M}}_3^*}(j) \tag{3.23}$$

$$\mathbf{CCY}_{\widehat{\mathbf{M}}_1^* \widehat{\mathbf{M}}_3^*}(\nu) = \sum_{i=-\infty}^{+\infty} \mathbf{YS}_{\widehat{\mathbf{M}}_1^*}(i+\nu)\mathbf{YS}_{\widehat{\mathbf{M}}_3^*}(i) \tag{3.24}$$

Finally, the *X*-*Y* displacements are obtained by respectively taking the row and column with the maximum *x* and *y* correlations as follows:

$$\Delta_x^f = \arg\max_{\tau} \mathbf{CCX}_{\widehat{\mathbf{M}}_1^* \widehat{\mathbf{M}}_3^*}(\tau) \tag{3.25}$$

$$\Delta_y^f = \arg\max_{\nu} \mathbf{CCY}_{\widehat{\mathbf{M}}_1^* \widehat{\mathbf{M}}_3^*}(\nu) \tag{3.26}$$

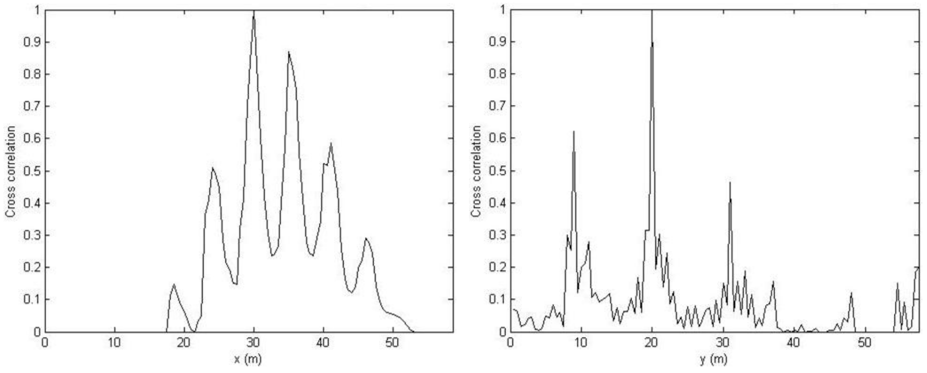


(a) *X*-spectrum of feature map 1 (b) *X*-spectrum of rotated feature map 2

(c) *Y*-spectrum of feature map 1 (d) *Y*-spectrum of rotated feature map 2

Figure 3.8 The *X*-*Y* spectra after the rotation Δ_θ^f is applied to the feature map 2. The *X*-*Y* spectra of the updated feature maps are extracted to find the *x*-*y* displacements.

The detailed algorithm to compute the *X* displacements between the two rasterized feature maps $\widehat{\mathbf{M}}_1^*$ and $\widehat{\mathbf{M}}_3^*$ updated by VSLs is summarized in Table 3.4. The detailed algorithm to compute the *Y* displacements between the two rasterized feature maps $\widehat{\mathbf{M}}_1^*$ and $\widehat{\mathbf{M}}_3^*$ updated by VSLs is summarized in Table 3.5. The cross correlation functions between the *X*-*Y* spectra extracted from the two rasterized feature maps are respectively shown in Fig. 3.9.



(a) Cross correlation between *X*-spectra (b) Cross correlation between *Y*-spectra

Figure 3.9 The results of the circular cross correlation for the *X*-*Y* displacements. The maximum of the circular cross correlation between the two *X*-spectra appears with the *X*-displacement $\Delta_x^f = 30\text{m}$. The maximum of the circular cross correlation between the two *Y*-spectra appears with the *Y*-displacement $\Delta_y^f = 20\text{m}$.

TABLE 3.4

ALGORITHM FOR ESTIMATING *X*-DISPLACEMENT BETWEEN FEATURE MAPS

Algorithm 4:	The computation of the *X*-displacement between two feature maps
Input:	The update and rasterized feature maps, $\widehat{\mathbf{M}}_1^*$ and $\widehat{\mathbf{M}}_3^*$
Output:	The *X*-displacement between two feature maps, Δ_x^f
1:	$\Delta_x^f \leftarrow \emptyset$
2:	**for** $j = 1$ **to** $\hat{c}_1$ **do**
3:	Compute the *X*- spectrum, which is $\mathbf{XS}_{\widehat{\mathbf{M}}_1^*}(j)$, by (3.19) **end for**
4:	**for** $j = 1$ **to** $\hat{c}_3$ **do**
5:	Compute the *X*- spectrum, which is $\mathbf{XS}_{\widehat{\mathbf{M}}_3^*}(j)$, by (3.21) **end for**
6:	**for** $\tau = 1$ **to** $\hat{c}_1$ **do**
7:	Compute the cross correlation $\mathbf{CCX}_{\widehat{\mathbf{M}}_1^*\widehat{\mathbf{M}}_3^*}(\tau)$ by (3.23) **end for**
8:	Compute the *X*-displacement, Δ_x^f, by (3.25)
9:	**return** Δ_x^f

TABLE 3.5

ALGORITHM FOR ESTIMATING *Y*-DISPLACEMENT BETWEEN FEATURE MAPS

```
Algorithm 5: The computation of the Y-displacement between two feature maps
Input:  The update and rasterized feature maps, $\widehat{\mathbf{M}}_1^*$ and $\widehat{\mathbf{M}}_3^*$
Output: The Y-displacement between two feature maps, $\Delta_y^f$

$\Delta_y^f \leftarrow \emptyset$
for $i = 1$ to $\hat{r}_1$ do
    Compute the Y- spectrum, which is $\mathbf{YS}_{\widehat{\mathbf{M}}_1^*}(i)$, by (3.20)
end for
for $i = 1$ to $\hat{r}_3$ do
    Compute the Y- spectrum, which is $\mathbf{YS}_{\widehat{\mathbf{M}}_3^*}(i)$, by (3.22)
end for
for $v = 1$ to $\hat{r}_1$ do
    Compute the cross correlation $\mathbf{CCY}_{\widehat{\mathbf{M}}_1^*\widehat{\mathbf{M}}_3^*}(v)$ by (3.24)
end for
Compute the Y-displacement, $\Delta_y^f$, by (3.26)
return $\Delta_y^f$
```

Because all elements in the MTM are obtained by the spectral information of the two feature maps, the feature maps can be merged by $\mathbf{T}_{\mathbf{M}_{12}}\left(\Delta_x^f, \Delta_y^f, \Delta_\theta^f\right)$. In the example for describing the SFMM, since the maximum values of the circular cross correlations appeared distinctively, the selection of the accurate MTM for feature map merging is easy. But, if there are several candidate MTMs with similar acceptance indices computed by (3.8) for the MTM, the most accurate MTM can be obtained by (3.9).

3.3 Enhanced-Spectrum-based Map Merging (ESMM)

Generally, the SMM requires a sufficient amount of overlapping regions between individual maps to find an accurate MTM. If the overlapping regions are not sufficient, the performance of map merging degenerates. Visual features [18, 19] or landmarks [28] can be used to reduce the required overlapping region. Their locations estimated by robots can be represented in the robot map. But, the estimated location of the visual landmark may contain inevitable errors due to the uncertainty of the measurements from the vision sensors. Thus, map merging by naively matching the location of common visual landmarks in different maps may be unsuccessful. This dissertation presents Enhanced-Spectra-based Map Merging (ESMM) which is an accurate map merging technique based on spectral correlation with visual objects.

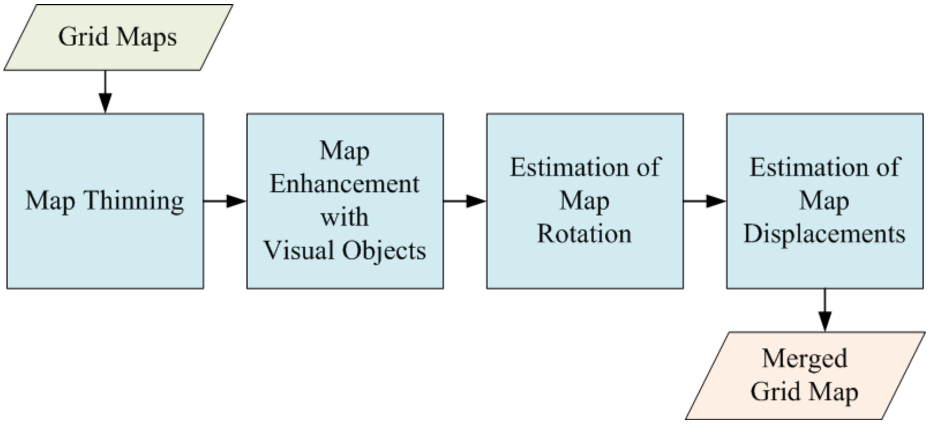

Grid Maps
Map Thinning
Map Enhancement with Visual Objects
Estimation of Map Rotation
Estimation of Map Displacements
Merged Grid Map

Figure 3.10 Whole structure of the ESMM. A merged grid map is obtained based on the MTM which consists of the estimated rotation angle and displacement amounts among the grid maps with visual objects by the ESMM.

3.3.1 Overview of the ESMM

The whole structure of the ESMM[2] is shown in Fig. 3.10. Firstly, the process of map thinning is performed to remove the irregular deviations of grid maps. Next, the map enhancement with visual objects from the original grid maps which is the key process of the ESMM is performed. Then, each grid is represented as three-dimensional cell which consists of *x*-coordinate, *y*-coordinate, and enhancement level, which is the main difference from the conventional map merging technique, SMM. Based on the enhanced grid maps with visual objects, the map rotation angle is estimated by finding the maximum rotational correlation between two rotational spectra. Then, the map displacements are estimated by finding the maximum translational correlation between two translational spectra. Finally, a collective merged grid map is obtained by the MTM.

3.3.2 Problem Formulation for the ESMM

Let a 2D grid map **M** be a matrix with n_r rows and n_c columns. Each grid contains the map information on the location represented by the grid in the global coordinate system. Generally, the grid information is defined as *occupied*, *unoccupied*, and *unknown*. In this dissertation, the occupied grids have additionally their own enhancement level.

[2] The ESMM which is one of the contributions in this dissertation has been published. Please refer to [73] for more detailed description.

For convenient map transformation, $\mathbf{M}$ is converted to its dual form $\mathbf{M}^d$ which is a matrix four rows and N_g columns. N_g is the number of occupied grids in $\mathbf{M}$. The first and second rows of $\mathbf{M}^d$ represent x-coordinate and y-coordinate of the occupied grids, respectively. The third row of $\mathbf{M}^d$ represents the enhancement level of the occupied grid. The last row is filled with 1 for just convenient computation. The map transformation matrix $\mathbf{T}$ which translates Δ_x in the direction of x-coordinate and Δ_y in the direction of y-coordinate, and rotates Δ_θ in a counter-clockwise is defined as follows:

$$\mathbf{T}(\Delta_x, \Delta_y, \Delta_\theta) = \begin{bmatrix} \cos\Delta_\theta & -\sin\Delta_\theta & 0 & \Delta_x \\ \sin\Delta_\theta & \cos\Delta_\theta & 0 & \Delta_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.27}$$

Then, the dual form of the transformed grid map can be simply obtained as follows:

$$\mathbf{M}_{tr}^d = \mathbf{T}\mathbf{M}^d \tag{3.28}$$

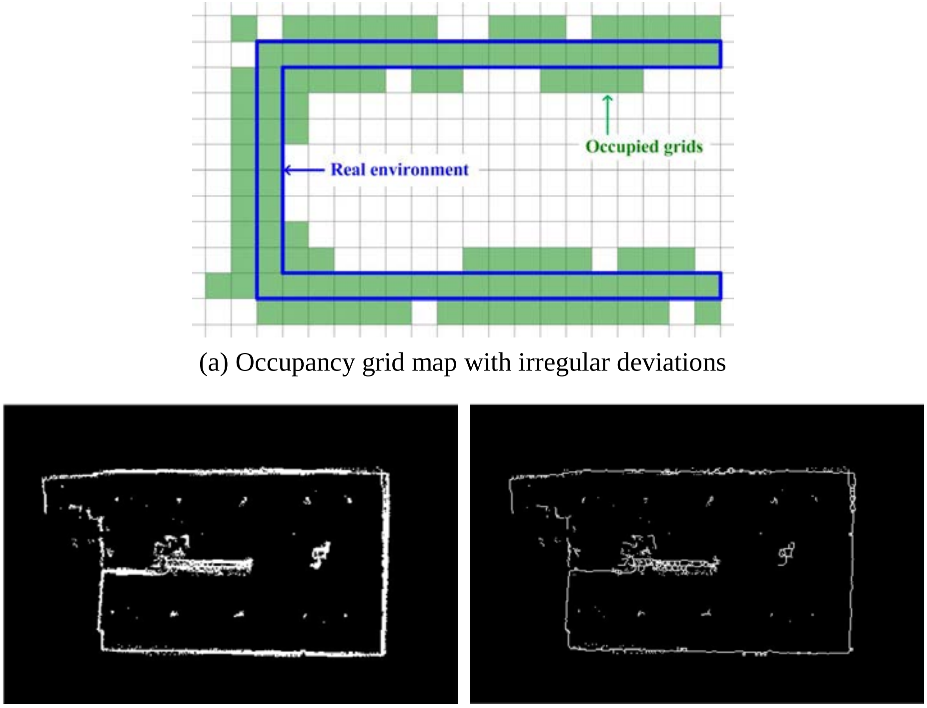
To evaluate the accuracy of the estimated MTM, Birk et al. [13] proposed an acceptance index to decide which one is better and to reject possible false positives. They computed the agreement and disagreement between two maps. The agreement was computed by the number of cells in two maps that are both free or both occupied, and the disagreement was computed by the number of cells such that one map is free and the other map is occupied or vice-versa. For the computation, the two maps should have the same size in the same coordinate system. However, this is a strong assumption in real multi-robot systems, especially with unknown initial relative poses between the multiple robots because the size of each map and the coordinates of each grid in the map should be carefully reconfigured to follow the definition of the acceptance index. Therefore, this paper defines a different metric to evaluate the accuracy of the estimated MTM. For given two maps, **M**$_i$ and **M**$_j$, the map merging error $\omega_E(\mathbf{M}_i, \mathbf{M}_j)$ is simply defined as follows:

$$\omega_E(\mathbf{M}_i, \mathbf{M}_j) = \sqrt{\| \Delta_x^{opt} \| - \| \widehat{\Delta_x} \|^2 + \Delta_y^{opt} - \widehat{\Delta_y}^{\;2} + L^2 \; \Delta_\theta^{opt} - \widehat{\Delta_\theta}^{\;2}} \tag{3.29}$$

where $\|\cdot\|$indicates an Euclidean norm. Δ_x^{opt} and $\widehat{\Delta_x}$ are the *x*-translation amount between **M**$_i$ and **M**$_j$. Δ_y^{opt} and $\widehat{\Delta_y}$ are the *y*-translation amount between **M**$_i$ and **M**$_j$. Δ_θ^{opt} and $\widehat{\Delta_\theta}$ are the optimal rotation angle and the estimated rotation angle between **M**$_i$ and **M**$_j$. L is a predetermined constant. The optimal values are given by a user through precise manual iterations in this dissertation.

3.3.3 Preprocessing – Map Thinning

The first step of the ESMM which is the presented map merging technique in this dissertation is the preprocessing: *map thinning*. The SMM extracts the spectral information from the lines massed in segments. However, the ESMM extracts the spectral information from piecewise consecutive occupied grid points. The locations of the occupied grid points are described in the dual forms of the original grid maps. Generally, a map has irregular deviations due to inevitable errors in sensors as shown in Fig. 3.11(a), which becomes worse in large environments. This deviated grid map should be refined to piecewise extract the spectral information. The ESMM utilizes the Zhang-Suen thinning algorithm [29] to refine the original grid map. The deviated grid map shown in Fig. 3.11(b) is refined by the thinning algorithm as shown in Fig. 3.11(c).

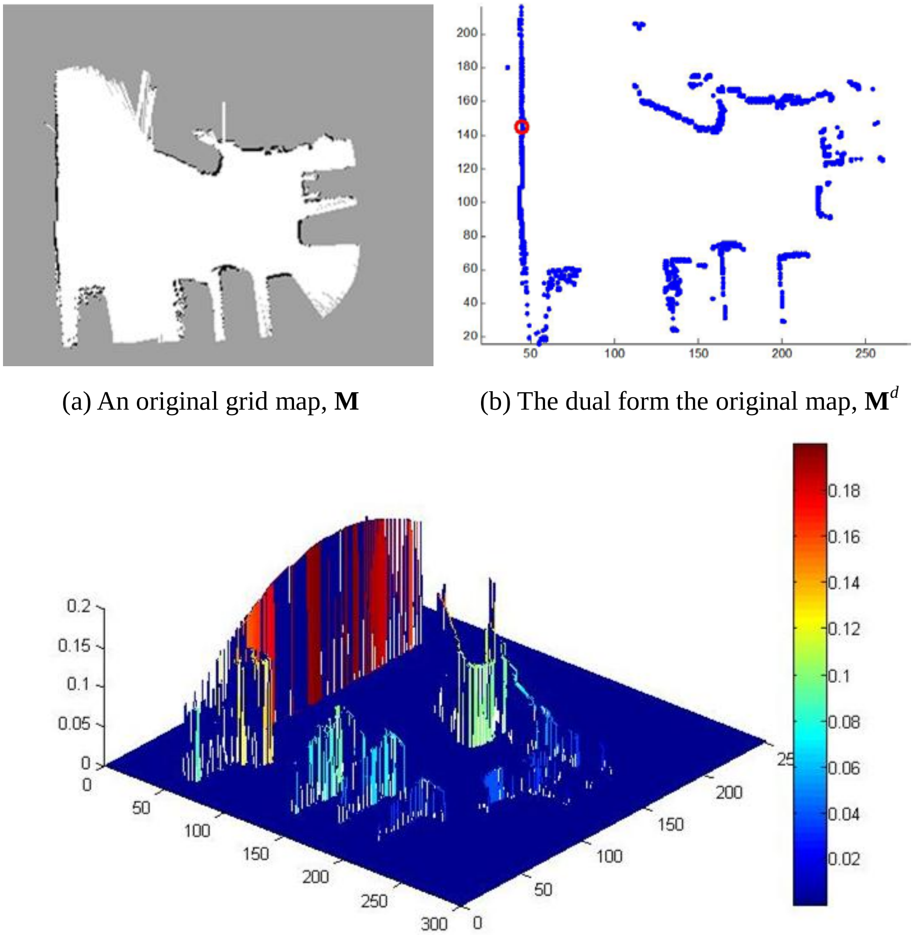


(a) Occupancy grid map with irregular deviations

(b) Grid map before map thinning

(c) Grid map after map thinning

Figure 3.11 Map thinning for irregularly deviated grid maps. (a) The region enclosed by blue solid lines represents real environments. Green squares represent the occupied grids in a map. (b) The map does not accurately describe the real environments due to irregular deviations. (c) The map is refined by the Zhang-Suen thinning algorithm.



(a) An original grid map, **M**

(b) The dual form the original map, $\mathbf{M}^d$

(c) The enhanced map, $\mathbf{M}^{eh}$

Figure 3.12 An example of map enhancement. (a) An original grid map is represented by occupied, unoccupied, and unknown grids. (b) The dual form of the original grid map is represented by the locations of the only occupied grids. The red circle located at (45,145) indicates the mean $\boldsymbol{\mu}$ for the location of a visual landmark where $\sigma = 100$. (c) The occupied grids in $\mathbf{M}^d$ are enhanced by computing the probabilistic density from the visual landmark. The *z*-axis indicates the enhancement level.

3.3.4 Map Enhancement

The next step is to enhance the original grid maps by the mean and covariance matrix for the location of a visual landmark. The enhancement is performed with the dual form $\mathbf{M}^d$ of a grid map $\mathbf{M}$ using a normal distribution. For the i-th occupied grid $\mathbf{g}_i = [x_i, y_i]^T$ in $\mathbf{M}$, its enhancement level l_i in the enhanced map $\mathbf{M}^{eh}$ is computed as follows:

$$l_i = \frac{S_n}{2\pi |\mathbf{\Sigma}_n|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{g}_i - \boldsymbol{\mu}_n)^T \mathbf{\Sigma}_n^{-1} (\mathbf{g}_i - \boldsymbol{\mu}_n) \right) \tag{3.30}$$

where $\mathbf{M}_i^{eh} = [x_i, y_i, l_i]^T$. $\boldsymbol{\mu}_n$ and $\mathbf{\Sigma}_n = [\sigma_n, 0;\ 0, \sigma_n]$ are the mean and covariance matrix for the location of the n-th visual landmark, respectively. S_n is a scaling factor. Figure 3.12 shows an example of map enhancement. After an original grid map $\mathbf{M}$ in Fig. 3.12(a) is converted to its dual form $\mathbf{M}^d$ in 3.12(b), map enhancement is performed with the visual landmark as shown in Fig. 3.12(c) where $S_n = 50$.

3.3.5 Estimation of Map Rotation

The map rotation angle between two maps can be estimated by matching their spectral information. First, the enhanced map $\mathbf{M}^{eh}$ in Cartesian space is transformed into three-dimensional Hough space, which is different from other map merging techniques based on spectral information. Figure 3.13 shows a geometry to describe the concept of Hough transform with the enhancement level indicated by the L-axis.

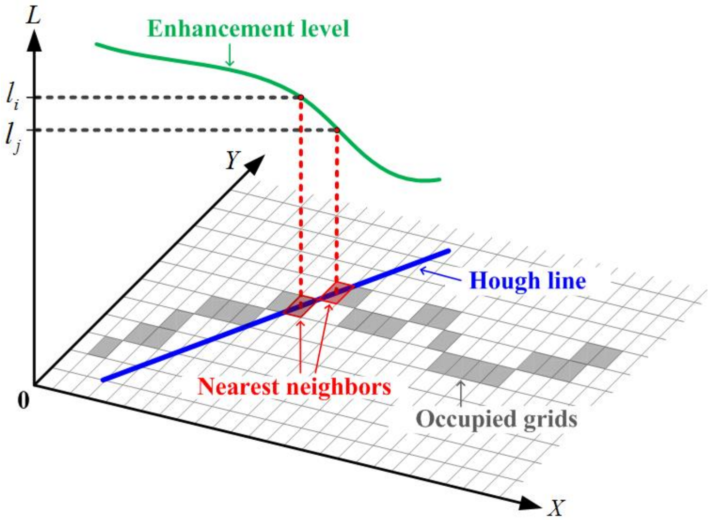


Figure 3.13 A geometry which describes the concept of Hough transform with the enhancement level. *X*-axis and *Y*-axis indicate Cartesian coordinates, and *L*-axis indicates the enhancement level. The gray squares represent occupied grids. The Hough line represented is generated from the two nearest neighbor grids represented by red squares. l_i and l_j are the enhancement levels of the two nearest neighbor grids.

For the i-th enhanced grid, $\mathbf{M}_i^{eh} = [x_i, y_i, l_i]^T$, its nearest neighbor enhanced grid, $\mathbf{M}_j^{eh} = \left[x_j, y_j, l_j\right]^T$, is selected. For the two enhanced grids, the corresponding component $\left[\rho_{ij}, \theta_{ij}, \kappa_{ij}\right]^T$ in three-dimensional Hough space is computed as follows:

$$\begin{bmatrix} \rho_{ij} \\ \theta_{ij} \\ \kappa_{ij} \end{bmatrix} = \begin{bmatrix} \dfrac{|(x_i - x_j)y_j - (y_i - y_j)x_j|}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \\ -\arctan\left(\dfrac{x_i - x_j}{y_i - y_j}\right) \\ \dfrac{l_i + l_j}{2} \end{bmatrix} \tag{3.31}$$

where ρ_{ij} and θ_{ij} represent the distance and angle from the origin to the Hough line generated by the two enhanced grids in Cartesian space. κ_{ij} represents the average enhancement level over the two nearest neighbor grids. Then, they are voted by κ_{ij} to the location (ρ_{ij}, θ_{ij}) in the transformed map $\mathbf{M}_H^*$ represented by a matrix with n_ρ rows and n_θ columns.

Let $\mathbf{M}_{1,H}^*$ and $\mathbf{M}_{2,H}^*$ be two enhanced and transformed maps from two original grid maps, $\mathbf{M}_1$ and $\mathbf{M}_2$. Then, their enhanced-Hough spectra are respectively extracted as follows:

$$\mathbf{HS}_{\mathbf{M}_1}(\theta) = \sum_{\rho=1}^{n_\rho} \mathbf{M}_{1,H}^*(\rho, \theta) \tag{3.32}$$

$$\mathbf{HS}_{\mathbf{M}_2}(\theta) = \sum_{\rho=1}^{n_\rho} \mathbf{M}_{2,H}^*(\rho, \theta) \tag{3.33}$$

where $1 \leq \theta \leq n_\theta$. The cross correlation between $\mathbf{HS}_{\mathbf{M}_1}$ and $\mathbf{HS}_{\mathbf{M}_2}$ can measure how much the two Hough spectra overlap each other. Therefore, the circular cross correlation is computed to find the map rotation angle along θ as follows:

$$\mathbf{CC}_{\mathbf{M}_1\mathbf{M}_2}(k) = \sum_{\theta=1}^{n_\theta} \mathbf{HS}_{\mathbf{M}_1}(i)\mathbf{HS}_{\mathbf{M}_2}(\theta + k) \tag{3.34}$$

Then, for $1 \leq k \leq n_\theta$, the map rotation angle is obtained by selecting the index with the maximum correlation between them as follows:

$$\Delta_\theta = \arg\max_k \mathbf{CC}_{\mathbf{M}_1\mathbf{M}_2}(k) \tag{3.35}$$

3.3.6 Estimation of Map Translations

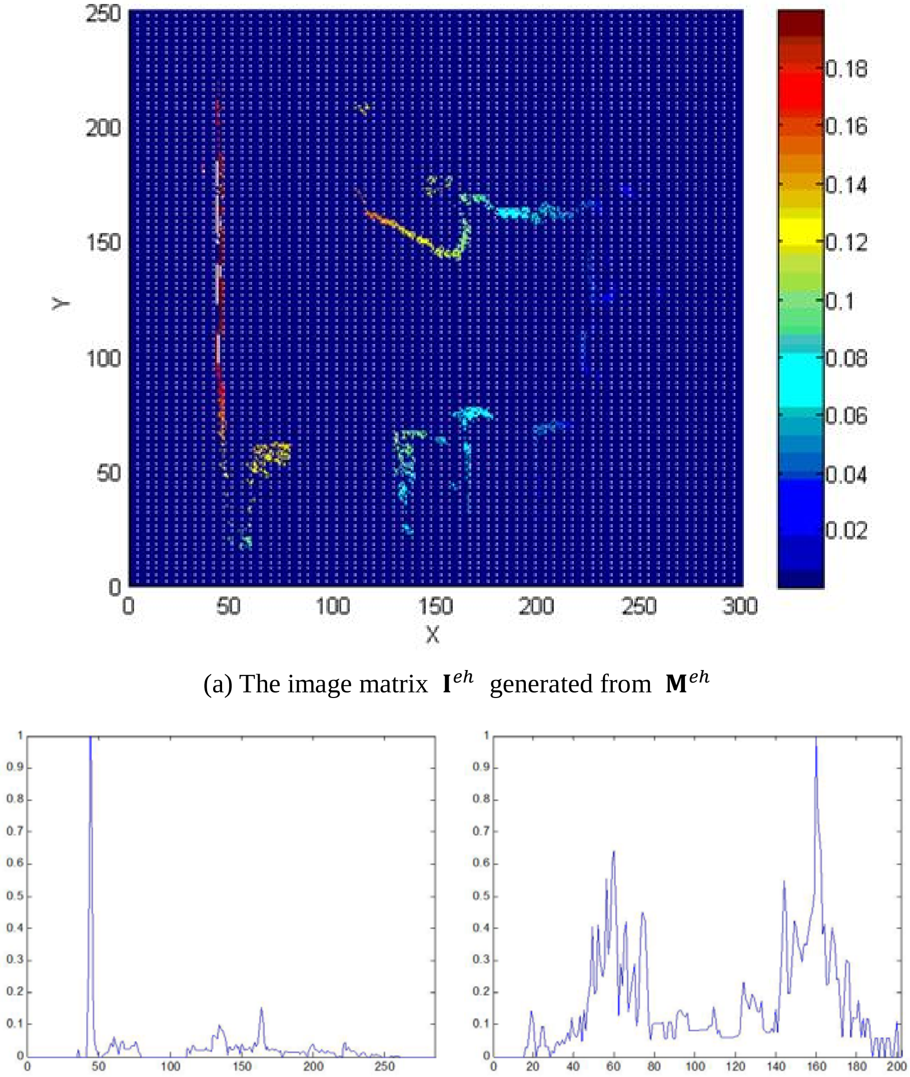
The estimated map rotation angle, $\varDelta_\theta$, is applied to the enhanced map $\mathbf{M}_2^{eh}$ as follows:

$$\tilde{\mathbf{M}}_2^{eh} = \mathbf{T}(0, 0, \Delta_\theta)\mathbf{M}_2^{eh} \tag{3.34}$$

where $\tilde{\mathbf{M}}_2^{eh}$ is the rotated enhanced map of $\mathbf{M}_2$. The estimation of map translations in the direction of *X*-axis and *Y*-axis is performed based on the image matrices from $\mathbf{M}_1^{eh}$ and $\tilde{\mathbf{M}}_2^{eh}$. The image matrix $\mathbf{I}^{eh}$ is generated by filling the location indicated by the first and second row of each grid in the enhanced dual form $\mathbf{M}^{eh}$ with the enhancement level indicated by the third row of each grid in the enhanced dual form $\mathbf{M}^{eh}$. The image matrix generated from the example enhanced map $\mathbf{M}^{eh}$ in Fig. 3.12(c) is shown in Fig. 3.14(a), which is equal to the top-view of $\mathbf{M}^{eh}$. The *X*-spectrum of an enhanced map is extracted from its image matrix by summing the values of the grids with the same *x*-coordinate. Similarly, the *Y*-spectrum of an enhanced map is extracted by summing the values of the grids with the same *y*-coordinate. Figure 3.14 shows the *X*-spectrum and the *Y*-spectrum of the example enhanced map $\mathbf{M}^{eh}$ in Fig. 3.12(c).

Given the two enhanced maps, $\mathbf{M}_1^{eh}$ and $\tilde{\mathbf{M}}_2^{eh}$, if the image matrix $\mathbf{I}_1^{eh}$ generated from $\mathbf{M}_1^{eh}$ is represented by n_{r_1} rows and n_{c_1} columns, its *X*-spectrum and *Y*-spectrum are computed as follows:

$$\mathbf{XS}_{\mathbf{M}_1^{eh}}(c) = \sum_{r=1}^{n_{r_1}} \mathbf{I}_1^{eh}(r, c) \tag{3.35}$$



(a) The image matrix $\mathbf{I}^{eh}$ generated from $\mathbf{M}^{eh}$

(b) *X*-spectrum

(c) *Y*-spectrum

Figure 3.14 Translational spectra of the example enhanced map $\mathbf{M}^{eh}$ in Fig. 3.12(c). (a) The image matrix $\mathbf{I}^{eh}$ generated from $\mathbf{M}^{eh}$ is equal to the top-view of $\mathbf{M}^{eh}$. (b) *X*-spectrum of $\mathbf{M}^{eh}$ is extracted from its image matrix $\mathbf{I}^{eh}$. (c) *Y*-spectrum of $\mathbf{M}^{eh}$ is extracted from its image matrix $\mathbf{I}^{eh}$.

$$\mathbf{YS}_{\mathbf{M}_1^{eh}}(r) = \sum_{c=1}^{n_{c_1}} \mathbf{I}_1^{eh}(r,c) \tag{3.36}$$

where $1 \leq r \leq n_{r_1}$ and $1 \leq c \leq n_{c_1}$. Similarly, if the image matrix $\tilde{\mathbf{I}}_2^{eh}$ generated from $\tilde{\mathbf{M}}_2^{eh}$ is represented by n_{r_1} rows and n_{c_1} columns, its *X*-spectrum and *Y*-spectrum are computed as follows:

$$\mathbf{XS}_{\tilde{\mathbf{M}}_2^{eh}}(c) = \sum_{r=1}^{n_{r_2}} \tilde{\mathbf{I}}_2^{eh}(r,c) \tag{3.37}$$

$$\mathbf{YS}_{\tilde{\mathbf{M}}_2^{eh}}(r) = \sum_{c=1}^{n_{c_2}} \tilde{\mathbf{I}}_2^{eh}(r,c) \tag{3.38}$$

where $1 \leq r \leq n_{r_2}$ and $1 \leq c \leq n_{c_2}$.

Then, the cross correlations are respectively computed to determine *X* and *Y* translations as follows:

$$\mathbf{CCX}_{\mathbf{M}_1^{eh}\tilde{\mathbf{M}}_2^{eh}}(\tau) = \sum_{c=-\infty}^{+\infty} \mathbf{XS}_{\mathbf{M}_1^{eh}}(c+\tau)\mathbf{XS}_{\tilde{\mathbf{M}}_2^{eh}}(c) \tag{3.39}$$

$$\mathbf{CCY}_{\mathbf{M}_1^{eh}\tilde{\mathbf{M}}_2^{eh}}(\nu) = \sum_{r=-\infty}^{+\infty} \mathbf{YS}_{\mathbf{M}_1^{eh}}(r+\nu)\mathbf{YS}_{\tilde{\mathbf{M}}_2^{eh}}(r) \tag{3.40}$$

where $n_{c_1} \geq n_{c_2}$ and $n_{r_1} \geq n_{r_2}$. The *X*-*Y* translations can be obtained by respectively selecting the column and row with the maximum *x*-correlation and *y*-correlation as follows:

$$\Delta_x = \arg\max_{\tau} \mathbf{CCX}_{\mathbf{M}_1^{eh}\tilde{\mathbf{M}}_2^{eh}}(\tau) \tag{3.41}$$

$$\Delta_y = \arg\max_{\nu} \mathbf{CCY}_{\mathbf{M}_1^{eh}\tilde{\mathbf{M}}_2^{eh}}(\nu) \tag{3.42}$$

Finally, by the MTM, $\mathbf{T}(\Delta_x, \Delta_y, \Delta_\theta)$, obtained from enhanced-spectral information of the two grid maps, we can obtain the map transformed from $\mathbf{M}_2^{eh}$, which is maximally matched with $\mathbf{M}_1^{eh}$ as follows:

$$\mathbf{M}_{2,tr}^{eh} = \mathbf{T}(\Delta_x, \Delta_y, \Delta_\theta)\mathbf{M}_2^{eh} \tag{3.43}$$

Chapter 4

Map Merging based on Tomographic Correlation

This section presents a new map merging technique based on tomographic correlation for multi-robot SLAM with unknown initial poses among robots. Tomography is a useful technique for medicine to extract the sectional images of a human body by computing the amounts of the penetrating waves such as X-rays, gamma rays, radio-frequency waves, electrons, and magnetic particles. This dissertation utilizes the concept of the X-ray computed axial tomography (CAT) scan. The key mathematical basis of the CAT scan is the Radon transform [30] which is an integral transform consisting of the integral of a function over straight lines, which is described in Section 4.3. The result of applying the Radon transform to two-dimensional data is called *sinogram* which is a three-dimensional representation. A grid map can be also regarded as the two-dimensional data which can be applied by the Radon transform. The presented map merging technique, Tomographic Map Merging (TMM), computes the MTM by matching the tomographic information, *sinograms*, extracted by the Radon transform. No predetermined rendezvous between robots or no common landmark between maps or no *a priori* information on overlapping regions between maps is not required. The TMM was tested with various pairs of grid maps built by real experiments and showed smaller map merging errors than a conventional map merging algorithm. Besides, the improved accuracy was consistent in spite of map noises.

4.1 Overview of the TMM

The TMM is an accurate grid map merging technique based on tomographic correlation. The whole structure of the TMM is shown in Fig. 4.1. Firstly, the original grid maps with *occupied* or *unoccupied* or *unknown* cells are changed to the grid maps with *occupied* or *unoccupied* cells. Next, the sinograms which are the tomographic information are extracted by applying the Radon transform to the changed grid maps, which is the key process of the TMM is performed. Based on the sinograms, the map rotation angle is estimated by finding the maximum rotational correlation. Then, the map displacements are estimated by finding the maximum translational correlation. Finally, a collective merged grid map is obtained by the MTM which consists of the estimated rotation angle and translation amounts.

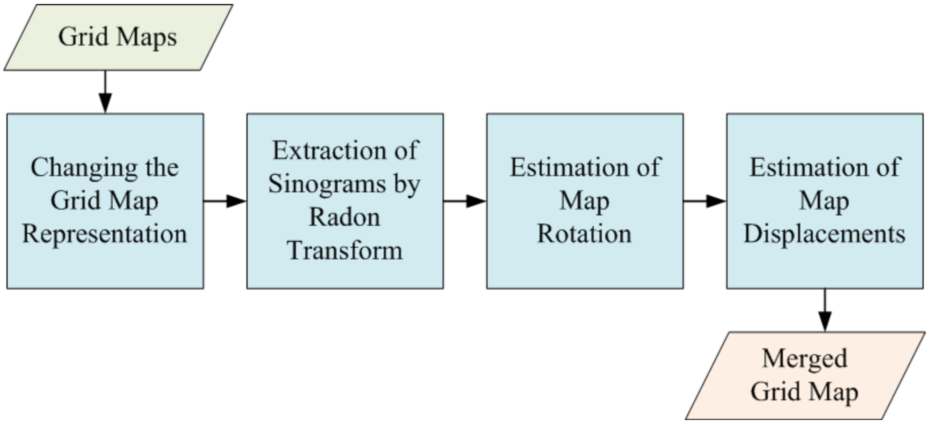

Grid Maps
Changing the Grid Map Representation
Extraction of Sinograms by Radon Transform
Estimation of Map Rotation
Estimation of Map Displacements
Merged Grid Map

Figure 4.1 The whole structure of the TMM. A merged grid map is obtained based on the MTM which consists of the estimated rotation angle and displacement amounts among the grid maps by the ESMM.

4.2 Problem Formulation for the TMM

A two-dimensional grid map **M** is assumed as a matrix with n_r rows and n_c columns, which can be regarded as a $n_r \times n_c$ binary image. Each grid contains map information on the location represented by the grid in the global coordinate system. The grid information in this dissertation is defined as *occupied* and *unoccupied* to compute the map transformation. However, the grid information is sometime defined as *occupied*, *unoccupied*, and *unknown*, which becomes also available after a simple conversion.

For convenient map transformation, $\mathbf{M}^d$ which is a dual form of **M** is defined as a matrix with three rows and N_{occ} columns. N_{occ} is the number of occupied grids in **M**. The first and second rows of $\mathbf{M}^d$ represent *x*-coordinate and *y*-coordinate of the occupied grids, respectively. The last row of $\mathbf{M}^d$ is filled with 1 so that the computation with a map transformation matrix is performed conveniently. Given two grid maps, $\mathbf{M}_1$ and $\mathbf{M}_2$, the map transformation matrix $\mathbf{T}$ which translates Δ_x in the direction of x-coordinate and Δ_y in the direction of y-coordinate, and rotates Δ_θ in a counter-clockwise is defined as follows:

$$\mathbf{T}(\Delta_x, \Delta_y, \Delta_\theta) = \begin{bmatrix} \cos\Delta_\theta & -\sin\Delta_\theta & \Delta_x \\ \sin\Delta_\theta & \cos\Delta_\theta & \Delta_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

where $\mathbf{M}_1' = \mathbf{T}\mathbf{M}_1$. Here, $\mathbf{M}_1'$ represented in the same coordinate system with $\mathbf{M}_2$ is a new map transformed from $\mathbf{M}_1$.

Although there exist only one optimal transformation, a map merging algorithm may produce multiple candidate transformations due to the errors in robot maps. To measure the error in the merged grid map, therefore, Birk et al. [13] proposed an acceptance index to decide which one is better and to reject possible false positives as follows:

$$\omega_a(\mathbf{M}_1, \mathbf{M}_2) = \begin{cases} 0 & \text{if } agr(\mathbf{M}_1, \mathbf{M}_2) = 0 \\ \dfrac{agr(\mathbf{M}_1, \mathbf{M}_2)}{agr(\mathbf{M}_1, \mathbf{M}_2) + dis(\mathbf{M}_1, \mathbf{M}_2)} & \text{if } agr(\mathbf{M}_1, \mathbf{M}_2) \neq 0 \end{cases} \tag{4.2}$$

where $agr(\mathbf{M}_1,\mathbf{M}_2)$ and $dis(\mathbf{M}_1,\mathbf{M}_2)$ represent the agreement and disagreement between $\mathbf{M}_1$ and $\mathbf{M}_2$, respectively. $agr(\mathbf{M}_1,\mathbf{M}_2)$ is computed by the number of cells in $\mathbf{M}_1$ and $\mathbf{M}_2$ that are both free or both occupied, and $dis(\mathbf{M}_1,\mathbf{M}_2)$ is computed by the number of cells such that $\mathbf{M}_1$ is free and $\mathbf{M}_2$ is occupied or vice-versa. According to their definition, $\mathbf{M}_1$ and $\mathbf{M}_2$ should have the same size in the same coordinate system. However, this is a strong assumption in real multi-robot systems, especially with unknown initial relative poses between the multiple robots. To utilize the acceptance index, the size of each map and the coordinates of each grid in the map should be carefully reconfigured to follow the definition of the acceptance index.

For the convenience of finding the best transformation, therefore, we define a new index to decide which one is better transformation as follow. Let $\mathbf{M}_1^d$ and $\mathbf{M}_2^d$ be the dual forms of two maps $\mathbf{M}_1$ and $\mathbf{M}_2$, respectively. Let N_{occ}^1 and N_{occ}^2 be the numbers of the occupied grids of $\mathbf{M}_1^d$ and $\mathbf{M}_2^d$, respectively. The *agreement* between $\mathbf{M}_1$ and $\mathbf{M}_2$, $agr(\mathbf{M}_1,\mathbf{M}_2)$, is the number of columns in $\mathbf{M}_1^d$ which represents the same coordinates with one of the columns in $\mathbf{M}_2^d$. Then, the acceptance index is defined as follows:

$$\omega_a^d(\mathbf{M}_1,\mathbf{M}_2) = \frac{agr^d(\mathbf{M}_1,\mathbf{M}_2)}{N_{occ}^1}. \tag{4.3}$$

As the overlapping region of $\mathbf{M}_1$ and $\mathbf{M}_2$ becomes larger, $\omega_a^d(\mathbf{M}_1,\mathbf{M}_2)$ becomes larger. If N_{occ}^1 and N_{occ}^2 are equal, and the two maps are matched perfectly, then $\omega_a^d(\mathbf{M}_1,\mathbf{M}_2) = 1$. If the two maps are not matched at all, $\omega_a^d(\mathbf{M}_1,\mathbf{M}_2) = 0$. Finally, for the possible transformation matrices $\mathbf{T}_i$ where $i = 1,\cdots,N_T$, the best transformation matrix is obtained by

$$\mathbf{T}_M = \arg\max_{\mathbf{T}_i} \omega_a^d(\mathbf{M}_1,\mathbf{T}_i\mathbf{M}_2) \tag{4.4}$$

where $\mathbf{M}_1' = \mathbf{T}_M\mathbf{M}_1$.

4.3 Extraction of Sinograms by the Radon Transform

The Radon transform [30] is an integral transform which consists of the integral of a function over straight lines. It is applicable to tomography which is the creation of an image from the scattering data associated to cross-sectional scans of an object. The inverse of the Radon transform can be used to reconstruct the original density from the scattering data, and thus it forms the mathematical underpinning for tomographic reconstruction, also known as image reconstruction. The Radon transform data is often called a *sinogram* because the Radon transform of a Dirac delta function is a distribution supported on the graph of a sine wave. Consequently, the Radon transform of a number of small objects appears graphically as a number of blurred sine waves with different amplitudes and phases.

Let $f(x,y)$ be a continuous function vanishing outside some large disc in the Euclidean plane $\mathbf{R}^2$ as shown in Fig. 4.2. The straight line L can be parameterized by

$$\left(x(t),y(t)\right)=\left((t\sin\alpha+s\cos\alpha),(-t\cos\alpha+s\sin\alpha)\right) \tag{4.5}$$

where t is a parameter for the parametric form of L, s is the distance of L from the origin, and α is the angle the normal vector to L makes with the x-axis. It follows that the quantities (s,α) can be considered as coordinates on the space of all lines in $\mathbf{R}^2$. The Radon transform, RT_f, is a function defined on the space of straight lines L in $\mathbf{R}^2$ by the line integral along each such line:

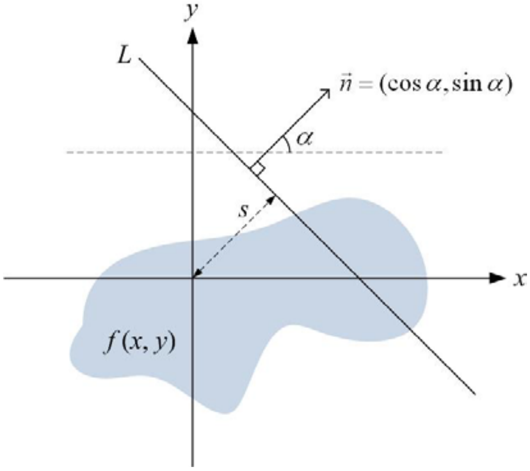


Figure 4.2 Concept of the Radon transform. A function $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})$ on the two-dimensional Euclidean space $\mathbf{R}^2$ is transformed into a function on a new two-dimensional space with α and *s* which defines the straight line *L* in $\mathbf{R}^2$.

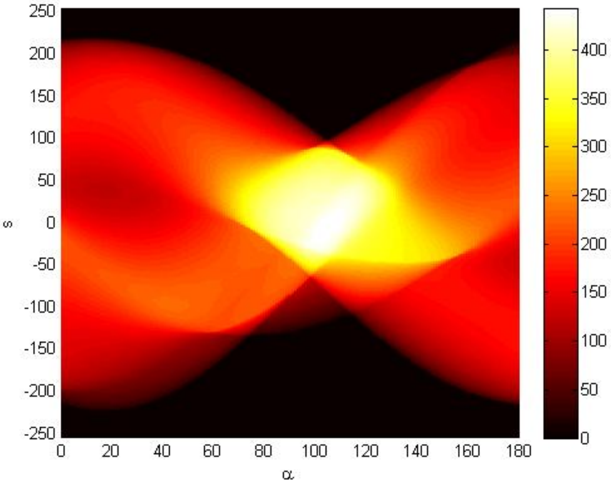


Figure 4.3 An example of applying the Radon transform. *s*-axis indicates the shortest distance from the origin to *L* in $\mathbf{R}^2$. α-axis indicates the angle between *x*-axis and the normal vector of *L* in Fig. 4.1. The maximum level is located at $(3,106°)$.

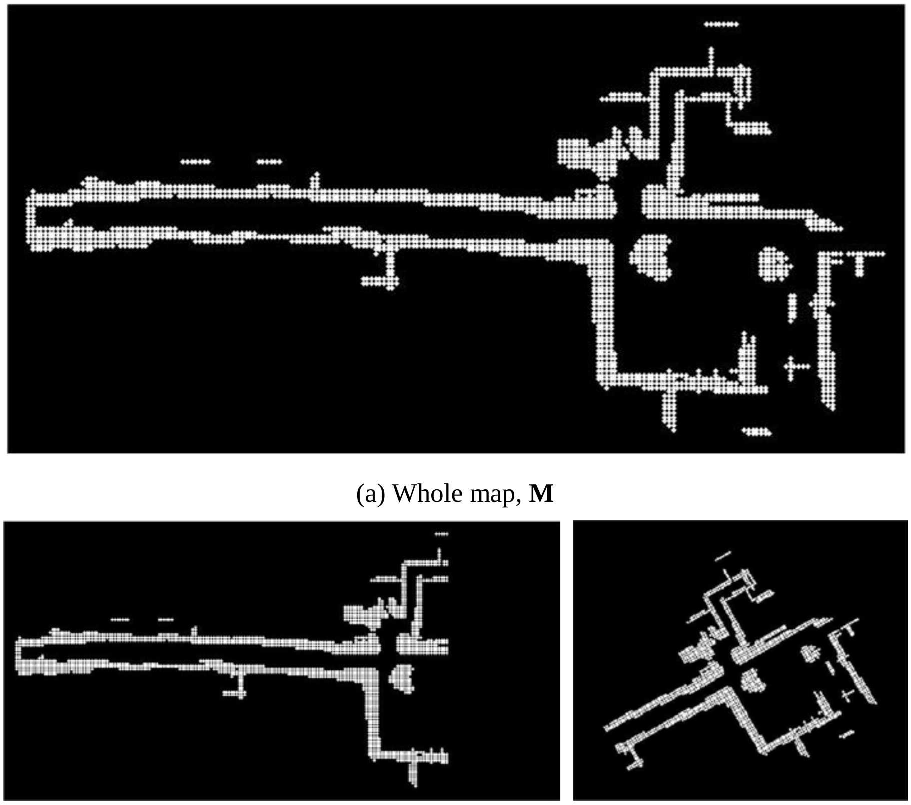
$$\begin{aligned} RT_f(s,\alpha) &= \int_{-\infty}^{\infty} f(x(t), y(t))\mathrm{d}t \\ &= \int_{-\infty}^{\infty} f\left((t\sin\alpha + s\cos\alpha), (-t\cos\alpha + s\sin\alpha)\right)\mathrm{d}t \\ &\equiv \mathbf{S}_f \end{aligned} \tag{4.6}$$

The example of applying the Radon transform to $f(x, y)$ is shown in Fig. 4.3, which is called *sinogram*. The horizontal and vertical axes indicate α-axis and s-axis, respectively. The level at each point (s, α) indicates the length of the intersection between $f(x, y)$ and L defined by s and α. Because the sinogram of a function is unique, it can be used as a feature set. Especially, the location (s, α) indicating the maximum level of the sinogram is a significant feature point. Because a grid map can be regarded as a two-dimensional continuous function in $\mathbf{R}^2$, the Radon transform can be applied to its feature extraction. Therefore, we uses the sinograms of two grid maps to find the best transformation $\mathbf{T}_M$ which consist of a rotation angle Δ_θ, a x-translation Δ_x, and a y-translation Δ_y between them.

4.4 Estimation of a Rotation Angle

In the sinogram of a grid map, the variation of column-wise levels according to α can indicate the shape of the grid map. Especially, α_{max} at the location of the maximum level indicates that the most elongated direction of the grid map is $\alpha_{max} - 90°$. For example, since the maximum level of $\mathbf{S}_f$ is located at $(3, 106°)$ as shown in Fig. 4.3, the most elongated direction of

$f(x,y)$ is 16°, which can be confirmed by Fig. 4.2. Therefore, the rotation angle between two grid maps can be estimated by comparing the locations indicating the maximum levels in their sinograms.



(a) Whole map, **M**

(b) Partial map 1, $\mathbf{M}_1$

(c) Partial map 2, $\mathbf{M}_2$

Figure 4.4 A whole map and its two partial maps. Each map consists of *occupied* white grids and *unoccupied* black grids. (a) **M** is a whole map which is unknown for a map merging system. (b) $\mathbf{M}_1$ is a part of **M**. (c) $\mathbf{M}_2$ is also a part of **M** and is rotated by Δ_θ. There is an overlapped region between $\mathbf{M}_1$ and $\mathbf{M}_2$.

Given two grid maps **M**$_1$ and **M**$_2$ which are parts of a whole map **M** as shown in Fig. 4.4 where the sizes of **M**$_1$ and **M**$_2$ are respectively $r_1 \times c_1$ and $r_2 \times c_2$, their sinograms are generated for $0° \leq \alpha \leq 180°$ as follow:

$$\mathbf{S}_{\mathbf{M}_1} = RT_{\mathbf{M}_1}(s,\alpha), \qquad -\rho_1 \leq s \leq \rho_1 \tag{4.7}$$

$$\mathbf{S}_{\mathbf{M}_2} = RT_{\mathbf{M}_2}(s,\alpha), \qquad -\rho_2 \leq s \leq \rho_2 \tag{4.8}$$

where ρ_1 and ρ_2 are respectively the maximum *s*-values of $\mathbf{S}_{\mathbf{M}_1}$ and $\mathbf{S}_{\mathbf{M}_2}$, which are computed as follows:

$$\rho_1 = \left\lceil \sqrt{\left(r_1 - r_1^*\right)^2 + \left(c_1 - c_1^*\right)^2} \right\rceil + 1 \tag{4.9}$$

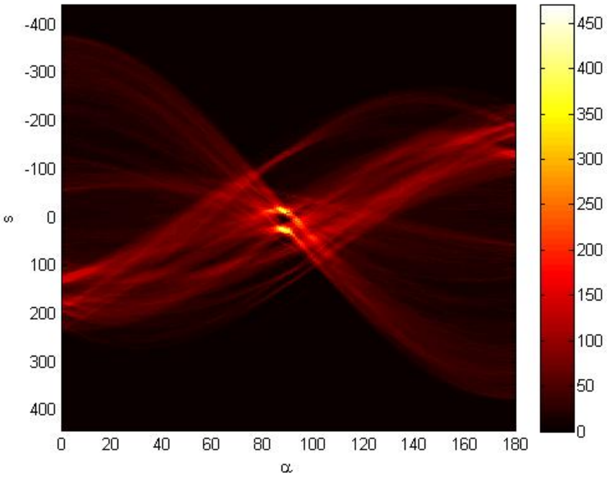
$$\rho_2 = \left\lceil \sqrt{\left(r_2 - r_2^*\right)^2 + \left(c_2 - c_2^*\right)^2} \right\rceil + 1 \tag{4.10}$$

where the center of **M**$_1$ is $(r_1^*, c_1^*) = \left(\left\lfloor \frac{r_1+1}{2} \right\rfloor, \left\lfloor \frac{c_1+1}{2} \right\rfloor\right)$, and the center of **M**$_2$ is $(r_2^*, c_2^*) = \left(\left\lfloor \frac{r_2+1}{2} \right\rfloor, \left\lfloor \frac{c_2+1}{2} \right\rfloor\right)$. Here, $\lceil \cdot \rceil$ denotes a ceiling function, and $\lfloor \cdot \rfloor$ denotes a floor function.

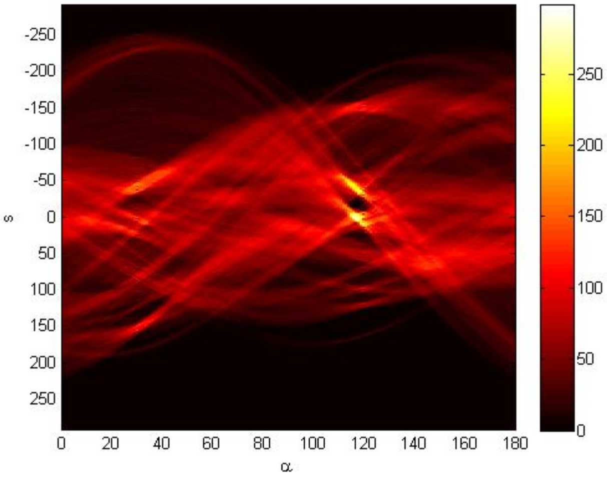
Let $(s_{1,max}, \alpha_{1,max})$ and $(s_{2,max}, \alpha_{2,max})$ be the locations of the maximum levels in $\mathbf{S}_{\mathbf{M}_1}$ and $\mathbf{S}_{\mathbf{M}_2}$. Then, the rotation angle Δ_θ between **M**$_1$ and **M**$_2$ can be estimated as follows:

$$\Delta_\theta = \alpha_{2,max} - \alpha_{1,max} \tag{4.11}$$

For example, the given maps and their sinograms as shown in Fig. 4.5, since $\alpha_{2,max} = 121°$ and $\alpha_{1,max} = 91°$, the rotation angle $\Delta_\theta = 30°$.



(a) The sinogram of **M**$_1$, denoted by $\mathbf{S}_{\mathbf{M}_1}$



(b) The sinogram of **M**$_2$, denoted by $\mathbf{S}_{\mathbf{M}_2}$

Figure 4.5 The sinograms of the partial maps, **M**$_1$ and **M**$_2$. The sinograms are generated by applying the Radon transform to the partial maps. (a) $\mathbf{S}_{\mathbf{M}_1}$ is the sinogram of **M**$_1$. The maximum level of $\mathbf{S}_{\mathbf{M}_1}$ is located at (472,91º). (b) $\mathbf{S}_{\mathbf{M}_2}$ is the sinogram of **M**$_2$. The maximum level of $\mathbf{S}_{\mathbf{M}_2}$ is located at (303,121º). The range of *s* in $\mathbf{S}_{\mathbf{M}_2}$ is smaller than the range of *s* in $\mathbf{S}_{\mathbf{M}_1}$ because the size of **M**$_2$ is smaller than the size of **M**$_1$.

4.5 Estimation of *X*-*Y* Translations

Because the estimation of translations in this paper is performed after map rotation is completed, one of the partial maps is rotated by the rotation angle estimated at the previous step as described in Sec. 4.4. Given the estimated rotation angle Δ_θ, $\widetilde{\mathbf{M}}_2^d$ is obtained by rotating $\mathbf{M}_2$ with Δ_θ as follows:

$$\widetilde{\mathbf{M}}_2^d = \mathbf{T}(0,0,\Delta_\theta)\mathbf{M}_2^d \tag{4.12}$$

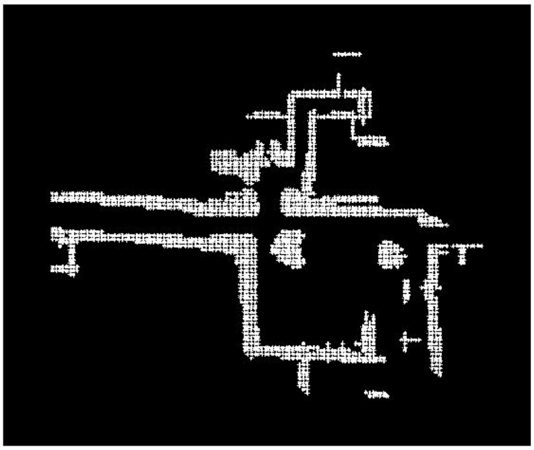
where $\widetilde{\mathbf{M}}_2^d$ and $\mathbf{M}_2^d$ are the dual forms of $\widetilde{\mathbf{M}}_2$ and $\mathbf{M}_2$, respectively. The size of $\widetilde{\mathbf{M}}_2$ is $\tilde{r}_2 \times \tilde{c}_2$, which is shown in Fig. 4.6(a). For $0° \leq \alpha \leq 180°$, the sinogram of $\widetilde{\mathbf{M}}_2^d$ is generated as follows:

$$\mathbf{S}_{\mathbf{M}_2} = RT_{\tilde{\mathbf{M}}_2}(s,\alpha), \qquad -\tilde{\rho}_2 \leq s \leq \tilde{\rho}_2 \tag{4.13}$$

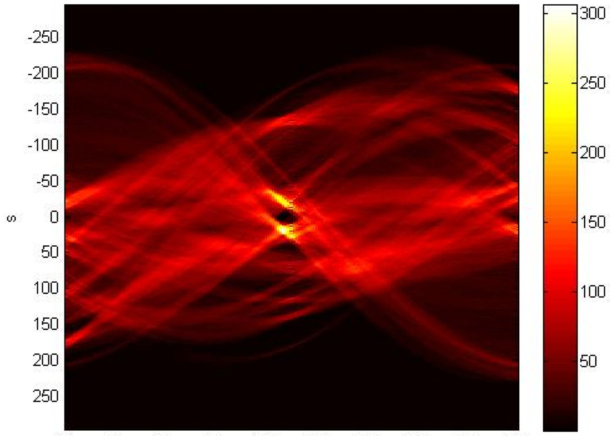
where $\tilde{\rho}_2$ is the maximum *s*-value of $\mathbf{S}_{\widetilde{\mathbf{M}}_2}$, which is computed as follows:

$$\tilde{\rho}_2 = \left\lceil \sqrt{\left(\tilde{r}_2 - \tilde{r}_2^*\right)^2 \tilde{+} \left(c_2 - \tilde{c}_2^*\right)^2} \right\rceil + 1 \tag{4.14}$$

The estimation of each translation can be obtained by finding the *s*-value which maximizes the cross correlation between two spectral vectors corresponding to a certain α-value. This spectral vector indicates the distribution of grids along the α direction, which means that the spectral vector is the distribution of *s* corresponding to the α-value. Given $\mathbf{S}_{\mathbf{M}_n}$ with $2\rho_n + 1$ rows and ϕ_n columns generated from the *n*-th partial map $\mathbf{M}_n$ with r_n rows and c_n columns as shown in Fig. 4.6(b), the spectral vector corresponding to the α'-value is defined as the *directional spectrum* of α':



(a) The rotated partial map, $\tilde{\mathbf{M}}_2$



(b) The sinogram of $\tilde{\mathbf{M}}_2$, denoted by $\mathbf{S}_{\tilde{\mathbf{M}}_2}$

Figure 4.6 The rotated partial map and its sinogram. To estimate the *x*-*y* translations between $\mathbf{M_1}$ and $\mathbf{M_2}$, $\tilde{\mathbf{M}}_\mathbf{2}$ is obtained by rotating $\mathbf{M_2}$ of Δ_θ, and its sinogram is generated by the Radon transform.

$$\mathbf{DS}_{\mathbf{M}_n}^{\alpha'}(t) = \mathbf{S}_{\mathbf{M}_n}(t - \rho_n - 1, \alpha'), \qquad 0 < t \leq 2\rho_n + 1 \tag{4.15}$$

where ρ_n is the maximum *s*-value of $\mathbf{S}_{\mathbf{M}_n}$, which can be computed by (4.9) or (4.10). Especially, the directional spectra for *x* and *y* axes can be computed by $\mathbf{DS}_{\mathbf{M}_n}^{180^\circ}(s)$ and $\mathbf{DS}_{\mathbf{M}_n}^{90^\circ}(s)$, respectively.

However, since the sizes of the partial maps are generally different like $\mathbf{M}_1$ and $\widetilde{\mathbf{M}}_2$, the directional spectra are redefined to compute the cross correlation between them. The spectra should be represented with the same dimension with r_m rows and c_m columns, which is defined as the maximum size of the partial maps $\mathbf{M}_1$ and $\widetilde{\mathbf{M}}_2$ as follows: $r_m = \max(r_1, \tilde{r}_2)$ and $c_m = \max(c_1, \tilde{c}_2)$. The *X* and *Y* spectra of $\mathbf{M}_1$ are redefined as follows:

$$\mathbf{XS}_{\mathbf{M}_1}(i) = \begin{cases} \mathbf{DS}_{\mathbf{M}_1}^{180^\circ}(i + \rho_1^* - c_m^*) & 1 \leq i \leq 2c_m^* + 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

$$\mathbf{YS}_{\mathbf{M}_1}(j) = \begin{cases} \mathbf{DS}_{\mathbf{M}_1}^{90^\circ}(j + \rho_1^* - r_m^*) & 1 \leq j \leq 2r_m^* + 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.17}$$

where $\rho_1^* = \left\lceil \frac{2\rho_1 + 1}{2} \right\rceil$, $c_m^* = \left\lceil \frac{2c_m + 1}{2} \right\rceil$, and $r_m^* = \left\lceil \frac{2r_m + 1}{2} \right\rceil$. Similarly, the *X*-spectrum and *Y*-spectrum of $\widetilde{\mathbf{M}}_2$ are redefined as follows:

$$\mathbf{XS}_{\tilde{\mathbf{M}}_2}(i) = \begin{cases} \mathbf{DS}_{\tilde{\mathbf{M}}_2}^{180^\circ}(i + \tilde{\rho}_2^* - c_m^*) & 1 \leq i \leq 2c_m^* + 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.18}$$

$$\mathbf{YS}_{\tilde{\mathbf{M}}_2}(j) = \begin{cases} \mathbf{DS}_{\tilde{\mathbf{M}}_2}^{90^\circ}(j + \tilde{\rho}_2^* - r_m^*) & 1 \leq j \leq 2r_m^* + 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.19}$$

The *x*-*y* translations between the first partial map $\mathbf{M}_1$ and the second rotated partial map $\tilde{\mathbf{M}}_2$ can be estimated by finding the arguments which maximize the cross correlation between their spectra. The cross correlation between them for *x*-*y* axes are respectively defined as follows:

$$\mathbf{CCX}_{\mathbf{M}_1\tilde{\mathbf{M}}_2}(\tau) = \begin{cases} \sum_{k=-\infty}^{\infty} \mathbf{XS}_{\mathbf{M}_1}(k+\tau)\mathbf{XS}_{\tilde{\mathbf{M}}_2}(k) & 1 \le \tau \le c_m \\ 0 & \text{otherwise} \end{cases} \tag{4.20}$$

$$\mathbf{CCY}_{\mathbf{M}_1\tilde{\mathbf{M}}_2}(\upsilon) = \begin{cases} \sum_{l=-\infty}^{\infty} \mathbf{YS}_{\mathbf{M}_1}(l+\upsilon)\mathbf{YS}_{\tilde{\mathbf{M}}_2}(l) & 1 \le \upsilon \le r_m \\ 0 & \text{otherwise} \end{cases} \tag{4.21}$$

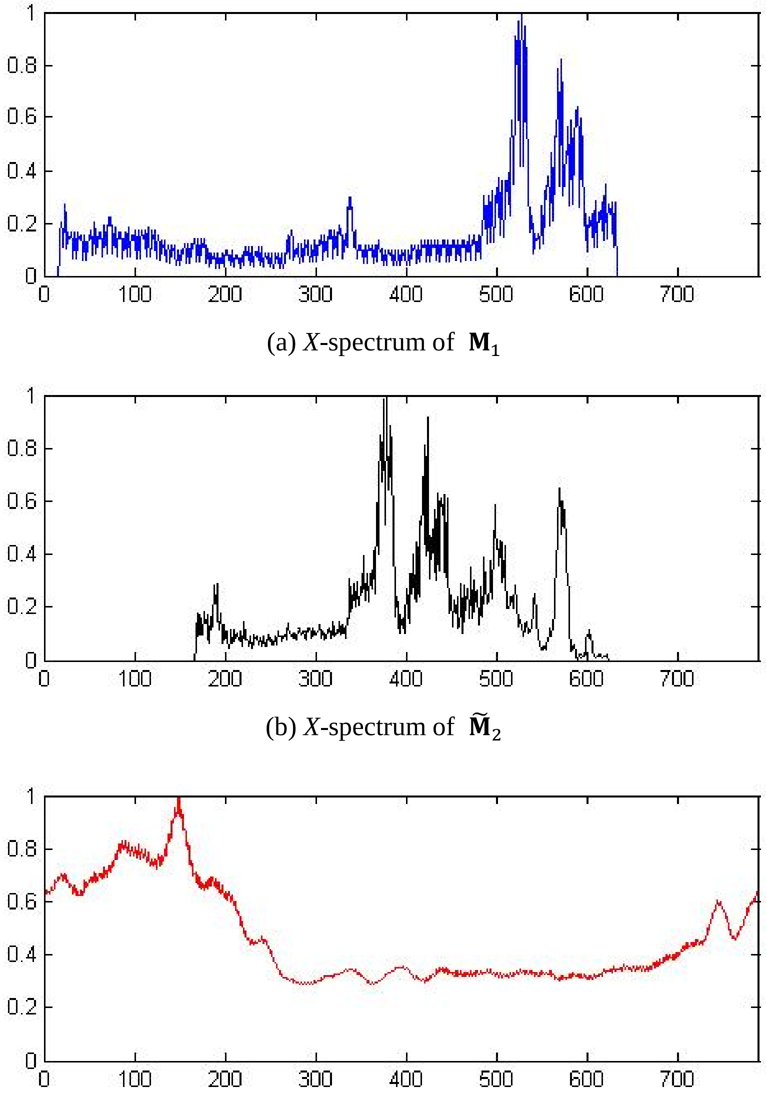
Finally, the *x*-translation between the partial maps is obtained by selecting the argument which maximizes the cross correlation of *X*-spectra as follows:

$$\Delta_x = \arg\max_{\tau} \mathbf{CCX}_{\mathbf{M}_1\tilde{\mathbf{M}}_2}(\tau) \tag{4.22}$$

Similarly, the *y*-translation between the partial maps is obtained by selecting the argument which maximizes the cross correlation of *Y*-spectra as follows:

$$\Delta_y = \arg\max_{\upsilon} \mathbf{CCY}_{\mathbf{M}_1\tilde{\mathbf{M}}_2}(\upsilon) \tag{4.23}$$

Figure 4.7 shows the *X*-spectra and their cross correlations, which are represented in the same dimensions. The *x*-translation between the two partial maps was obtained by (4.22) where $\Delta_x = 148$. Figure 4.8 shows the *Y*-spectra and their cross correlations, which are represented in the same dimensions. The *y*-translation between the two partial maps was obtained by (4.23) where $\Delta_y = 4$.

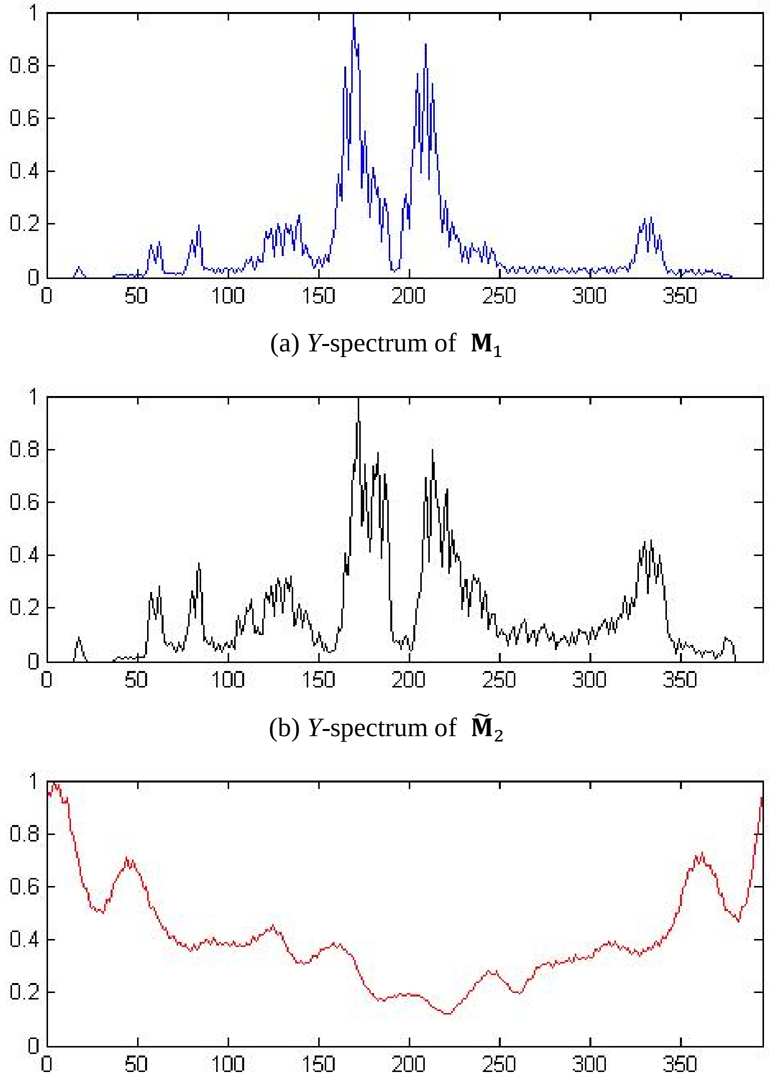


(a) X-spectrum of $\mathbf{M}_1$

(b) X-spectrum of $\widetilde{\mathbf{M}}_2$

(c) Cross correlation between the two X-spectra

Figure 4.7 The directional spectra for x-axis and their cross correlation. The x amount corresponding to the maximum cross-correlation is selected as Δ_x.



(a) Y-spectrum of $\mathbf{M}_1$

(b) Y-spectrum of $\widetilde{\mathbf{M}}_2$

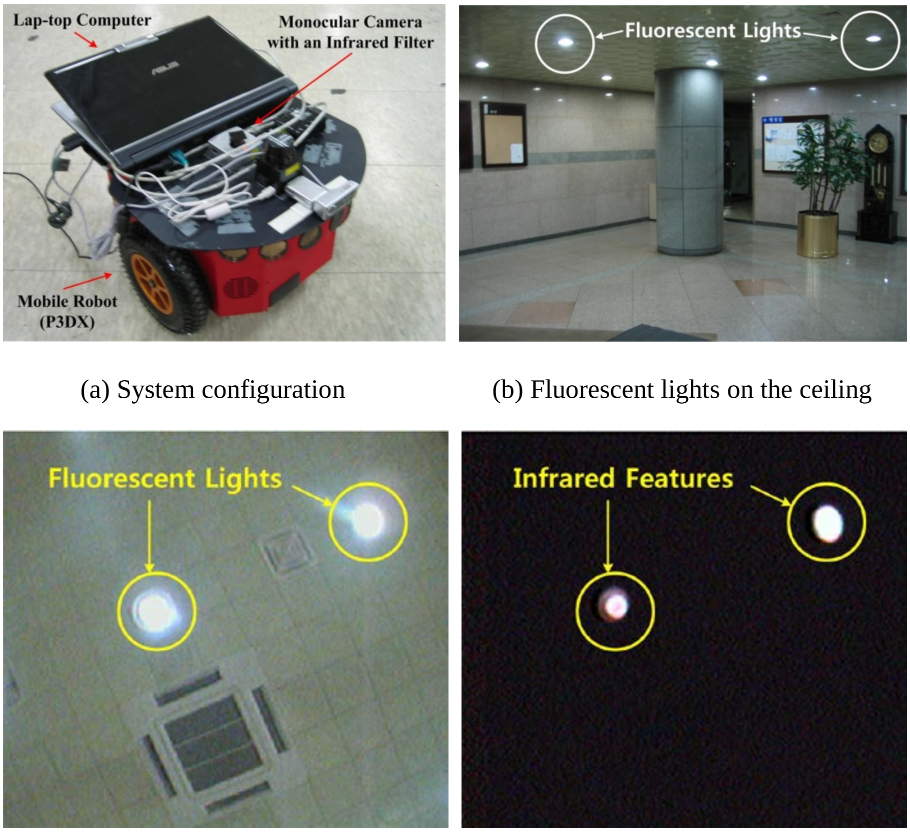
(c) Cross correlation between the two Y-spectra

Figure 4.8 The directional spectra for y-axis and their cross correlation. The y amount corresponding to the maximum cross-correlation is selected as Δ_y.

Chapter 5

Experiments

This section shows the experimental results of the correlation-based map merging techniques which are presented in this dissertation. Because the experimental setups for them are different one another, the experimental results are shown separately. Section 5.1 shows the experimental results of the SFMM with two mobile robots in indoor environments with indistinguishable infrared features whose locations are unknown for robots. Section 5.2 shows the experimental results of the ESMM with three mobile robots in outdoor environments with distinguishable visual objects whose locations are also unknown for robots. Section 5.3 shows the experimental results of the TMM with three mobile robots in both an outdoor environment and an indoor environment without any robot-to-robot measurements or common objects.



(a) System configuration

(b) Fluorescent lights on the ceiling

(c) Original image obtained from the camera

(d) Infrared-filtered image

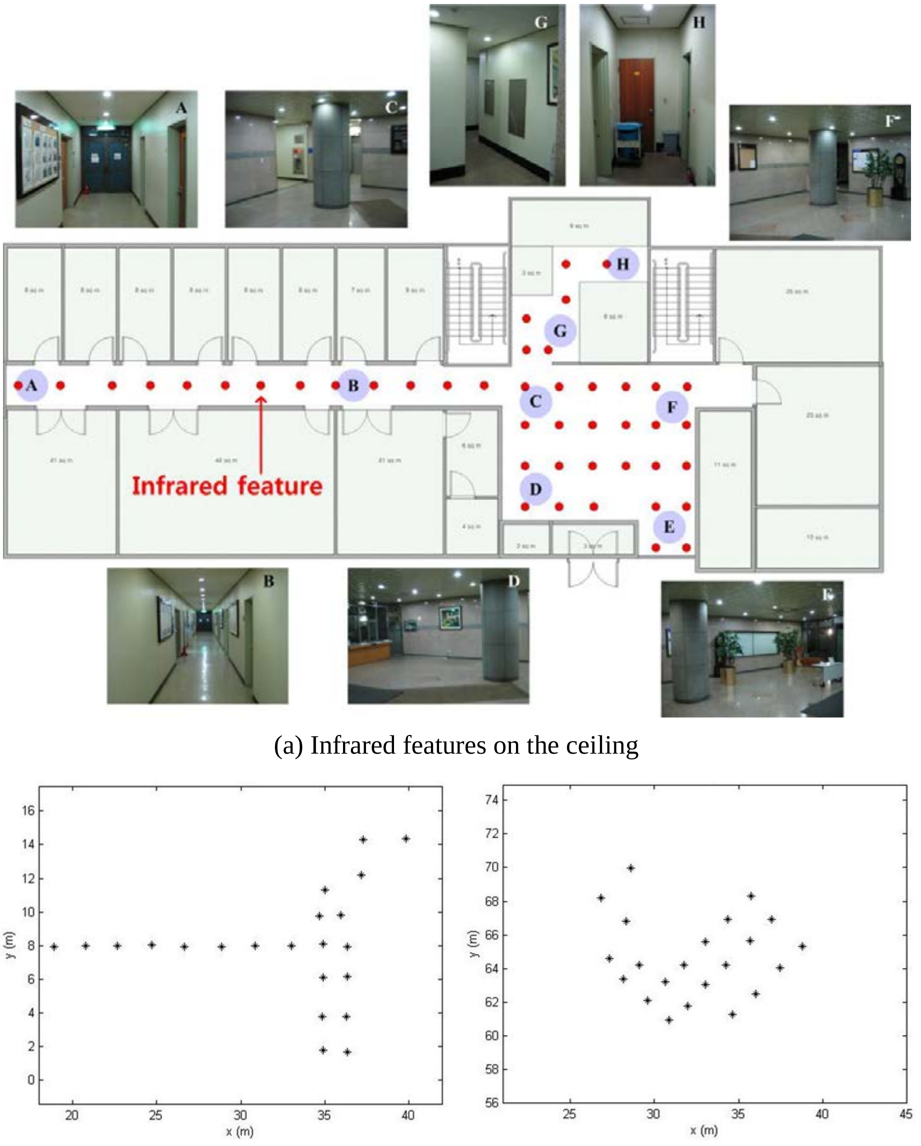
Figure 5.1 The observation system for building a feature map in experiments. The fluorescent lights as the infrared features were detected by the monocular camera with infrared band-pass filter.

5.1 Experimental Results of the SFMM

To verify the performance of the SFMM, we tested it in a real indoor environment whose size was 17m×48m. Each robot estimated its own pose and the locations of the infrared features on the ceiling, simultaneously.

Figure 5.1 shows an example of the infrared feature and the robot system for observing it. An upward monocular camera with an infrared band-pass filter was equipped at the center of the mobile robot P3DX as shown in Fig 5.1(a), and a fluorescent light shown in Fig. 5.1(b) as an infrared feature was detected by filtering the original image from the camera as shown in Fig 5.1(c)-(d). The most distinguished advantage of this system is to minimize the computational cost and the number of outliers. The feature location on the infrared-filtered image is estimated by taking the center of mass of the white blob shown in Fig. 5.1(d). Then, the real location of the feature is estimated with a simple observation model which projects the infrared features on the ceiling onto the image plane. The maximum likelihood (ML) data association was employed to distinguish the feature measurements, which was simple but sufficient for our observation system which produced the sparsely distributed feature measurements. The more detailed observation model is described in [1, 47].

Figure 5.2(a) shows the global map with infrared features whose locations are unknown for robots. Figure 5.2(b) and Figure 5.2(c) show the two feature maps built by an improved version of FastSLAM [23]. The two feature maps have been built with different coordinate systems because the initial locations and orientations of the robots were unknown for each other. The errors due to slips and the inaccuracy of measurements were satisfactorily reduced, and thus nice feature maps were produced.

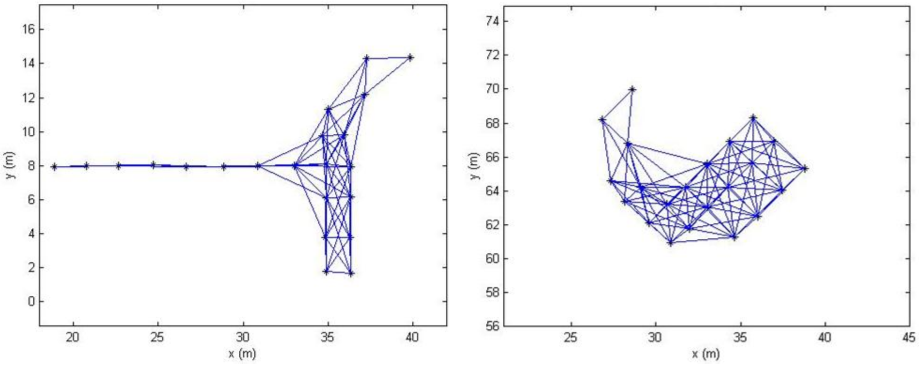


(a) Infrared features on the ceiling

(b) Infrared features on the ceiling

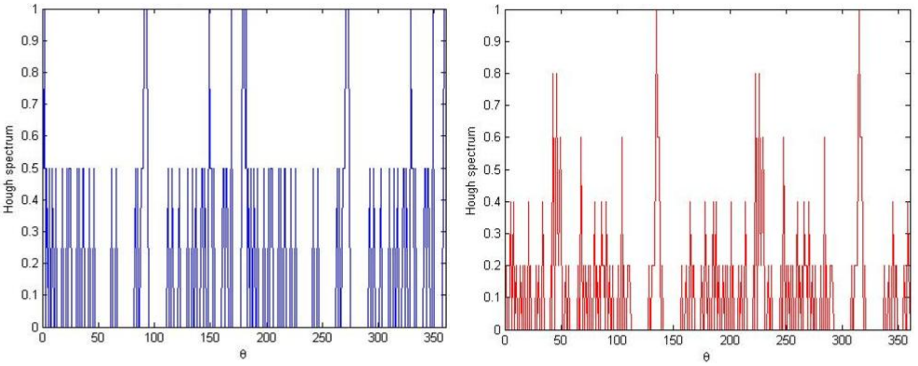
(c) System configuration

Figure 5.2 The global feature map and partial feature maps built by different robots. (a) The red points indicate the locations of the infrared features. (b)-(c) The two robots built their own feature maps by SLAM algorithms. The robot-to-robot measurements were not needed for the SFMM.



(a) Feature map 1 updated with VSLs (b) Feature map 2 updated with VSLs

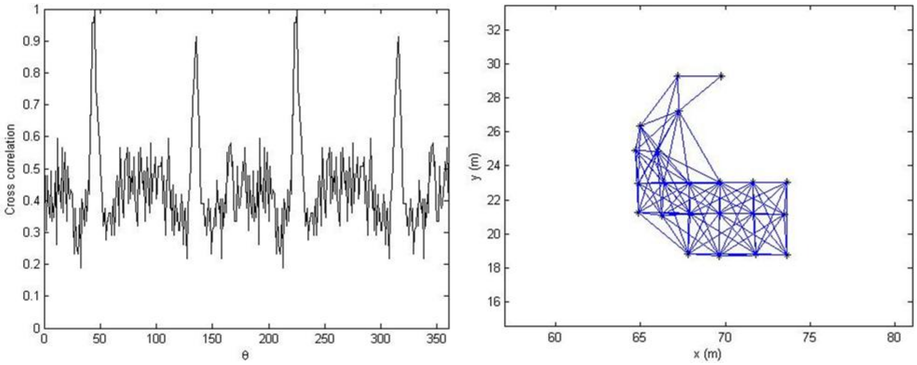
Figure 5.3 The updated feature maps with VSLs in experiments. To extract the spectral information by the Hough transform, the two feature maps were updated with VSLs which uses the circular geometric kernel with *R*=5.



(a) Hough spectrum of feature map 1 (b) Hough spectrum of feature map 2

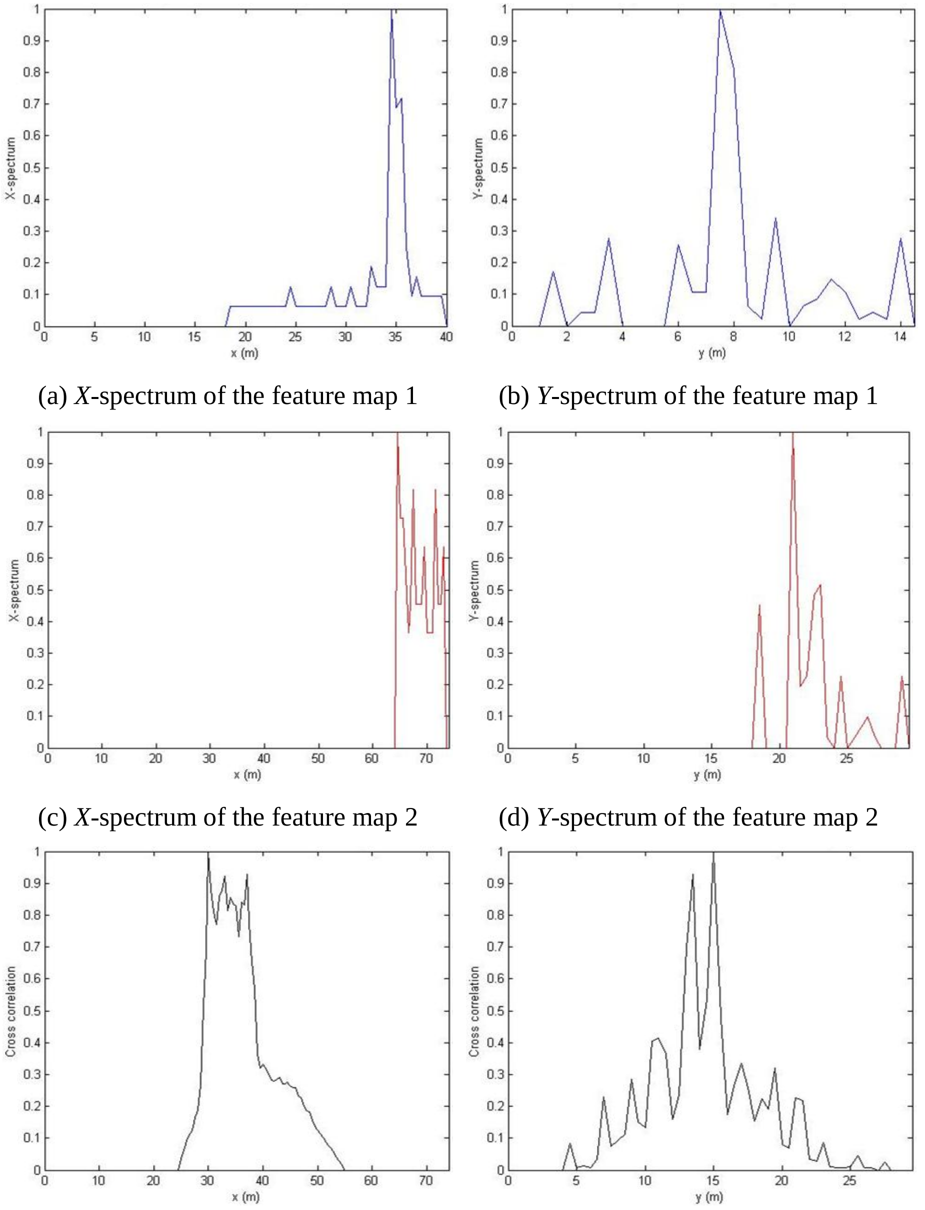
Figure 5.4 The Hough spectra extracted from the updated feature maps in experiments. The spectral information on the map rotation was extracted by the Hough transform.

To extract the spectral information of the two feature maps, each feature map was updated with VSLs generated by Algorithm 1 described in Table 3.1 where *R*=5 as shown in Fig. 5.3. Then, the DHT was applied to the updated feature maps, and the spectral information was extracted from the transformed maps by Algorithm 2 in Table 3.2. Figure 5.4 shows the results of extracting the Hough spectra of the two feature maps with VSLs. The cross correlation function between the two Hough spectra is shown in Fig. 5.5(a). There were several local maxima, but the SFMM accurately estimated the map rotation angle as $\Delta_{\theta}^{f} = 45°$ by (3.17). After the feature map 2 was rotated by Δ_{θ}^{f}, the two feature maps updated with VSLs were rasterized by Algorithm 3 described in Table 3.3. Based on the rasterized maps, the spectral information on *x* and *y* coordinates was extracted. Since the resolution affects the accuracy and computation time of the SFMM, it should be determined appropriately. In these experiments, the resolution of the rasterization, η_r, was set to 0.5.



(a) Cross correlation between Hough spectra (b) Rotated feature map 2 with VSLs

Figure 5.5 The result of estimating the rotation angle in experiments. The rotation angle between the updated feature maps was estimated to the angle $\Delta_{\theta}^{f} = 45°$ which maximizes the circular cross correlation.



(a) *X*-spectrum of the feature map 1

(b) *Y*-spectrum of the feature map 1

(c) *X*-spectrum of the feature map 2

(d) *Y*-spectrum of the feature map 2

(e) Cross-correlation between *X*-spectra

(f) Cross-correlation between *Y*-spectra

Figure 5.6 The results of estimating the *x*-*y* translation amounts in experiments. The *x*-*y* translations between the feature maps were respectively estimated to $\Delta_x^f = 30\text{m}$ and $\Delta_x^f = 15\text{m}$ which maximize the circular cross correlation functions.

The *x*-*y* displacements were estimated by the *X*-*Y* spectra of the two feature maps with Algorithm 4 and Algorithm 5 described in Table 3.4 and Table 3.5, respectively. Note that the SFMM does not depend on the order in the estimation of *x*-*y* displacements. Figure 5.6 shows the results of estimating the *x*-*y* displacements with the spectral information. The *X*-spectra of the two feature maps were respectively extracted as shown in Fig. 5.6(a) and Fig. 5.6(c). The *Y*-spectra of them were respectively extracted as shown in Fig. 5.6(b) and Fig. 5.6(d). Finally, the *x*-displacement Δ_x^f was estimated to 30m which maximizes the cross correlation between two *X*-spectra as shown in Fig. 5.6(e). The *y*-displacement Δ_y^f was estimated to 15m which maximizes the cross-correlation between two *Y*-spectra as shown in Fig. 5.6(f).

To evaluate the performance of the SFMM, the accuracy of feature map merging was measured by the acceptance index defined by (3.2) according to the map rotation angles from 0º to 360º. The 500 map rotation angles were randomly sampled from the uniform distribution in [0º, 360º]. Figure 5.7 shows the result of measuring the acceptance index ω_f according to the randomly sampled map rotation angles. The tolerance Γ_f was 0.5m. The maximum ω_f of feature map merging was 0.9999, and the average $\overline{\omega}_f$ over the acceptance index of feature map merging was 0.9456. The acceptance indices according to Γ_f are summarized in Table 5.1. The smaller Γ_f causes the smaller $\overline{\omega}_f$ because the small Γ_f causes the negative ω_f by the definition in (3.8),

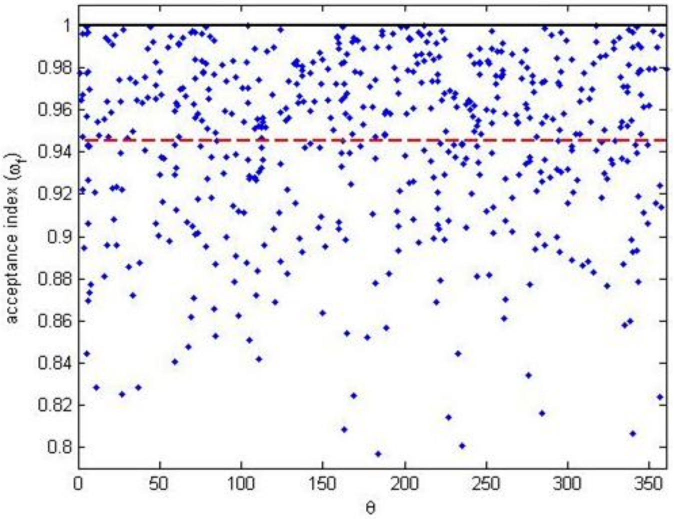


Figure 5.7 The acceptance index ω_f according to the map rotation angle in experiments. The acceptance index was measured with the 500 angles randomly sampled from uniform distribution in [0º, 360º]. The average acceptance index $\bar{\omega}_f$ was 0.9456 as described by the dashed red line.

TABLE 5.1

AVERAGE ACCEPTANCE INDICES ACCORDING TO THE TOLERANCE Γ_f

Γ_f	0.1	0.3	0.5	0.7	0.9
$\bar{\omega}_f$	0.9298	0.9380	0.9456	0.9516	0.9607

Note that there are several candidates for Δ_θ^f, Δ_x^f, and Δ_y^f in the circular cross correlation functions, respectively. For example, in the case of the circular cross correlation function between the two Hough spectra shown in Fig. 5.5(a), the rotation angle θ where $\mathbf{CC}_{\widehat{\mathbf{M}}_1^*\widehat{\mathbf{M}}_3^*}(\theta) \geq 0.9$ could be chosen as the candidates, which are 43º, 44º, 45º, and 136º. The acceptance indices of the four candidates for Δ_θ^f are summarized in Table 5.2, where the tolerance $\Gamma_f = 1.0°$. Obviously, the best rotation angle θ was 45º, and the SFMM found it consistently in spite of the existence of the several candidates. For the circular cross correlation between the two X-spectra shown in Fig. 5.6(e), the x-coordinates where $\mathbf{CCX}_{\widehat{\mathbf{M}}_1^*\widehat{\mathbf{M}}_3^*}(x) \geq 0.9$ could be chosen as the candidates, which are 30.0m, 33.0m, and 37.0m. The acceptance indices of the three candidates for Δ_x^f are summarized in Table 5.2, where the tolerance $\Gamma_f = 1.0$m. The best x-displacement was 30.0m, and the proposed technique found it consistently despite the existence of the several candidates which have similar cross correlations. For the circular cross correlation between the two Y-spectra shown in Fig. 5.6(f), the y-coordinates where $\mathbf{CCY}_{\widehat{\mathbf{M}}_1^*\widehat{\mathbf{M}}_3^*}(y) \geq 0.9$ could be chosen as the candidates, which are 13.5m and 15.0m. The acceptance indices of the two candidates for Δ_y^f are summarized in Table 5.2, where the tolerance $\Gamma_f = 1.0$m. The best y-displacement was 15.0m, and the SFMM found it consistently despite the existence of the several candidates which have similar cross correlations.

TABLE 5.2

AVERAGE ACCEPTANCE INDICES OF THE CANDIDATES IN EXPERIMENTS

Type	Δ_θ^f (º)				Δ_x^f (m)			Δ_y^f (m)	
Candidates	43	44	45	136	30.0	33.0	37.0	13.5	15.0
ω_f	-0.79	-0.27	0.99	-10.8	0.99	-0.73	-1.65	-0.22	0.99

5.2 Experimental Results of the ESMM

The ESMM was tested by real experiments with three mobile robots in two outdoor environments. One was the parking area in the front of Automation and Systems Research Institute at Seoul National University, where its size is 25m×30m. The other one was the roof of a building, where its size is 35m×20m. The used robots were Pioneer3-DX with laser scan sensors (Hokuyo UTM-30LX) and cameras (Sony PTZ SNC-RZ50) as shown in Fig. 5.8. The detectable range of the laser scan sensor was 30m within ±135°. Each robot started at different locations in the same time where the robots did not know the initial locations of the other robots. While each robot built its own map by our SLAM techniques [23]-[26], visual landmarks were recognized by cameras and localized in the built map.

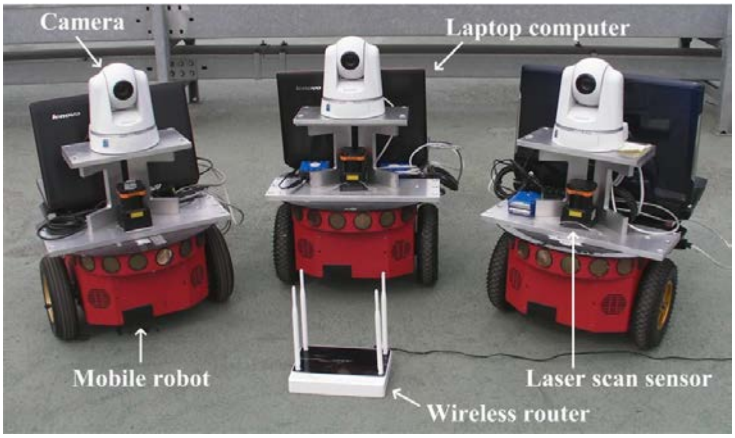
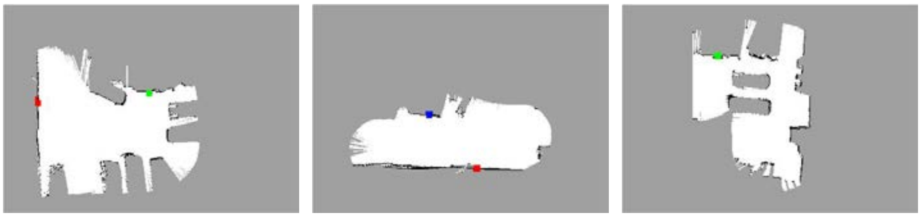
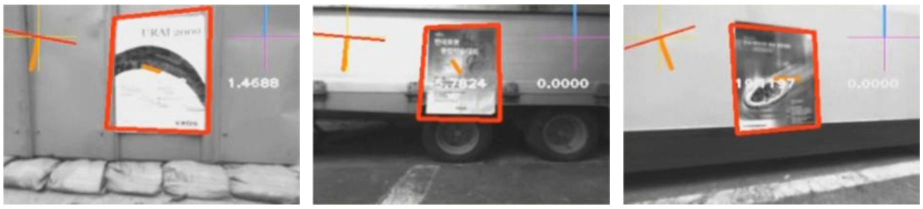


Figure 5.8 The hardware system for the ESMM. Each robot performs SLAM with a laser scan sensor while recognizing and localizing visual objects with a camera. Their maps are sent to one of the robots via a wireless router and merged by the ESMM.



(a) Map 1, $\mathbf{M}_1$ (b) Map 2, $\mathbf{M}_2$ (c) Map 3, $\mathbf{M}_3$

Figure 5.9 The individual maps built by different robots which started at different locations in the parking area. The red, green and blue squares represent the means of the locations of the visual landmarks.

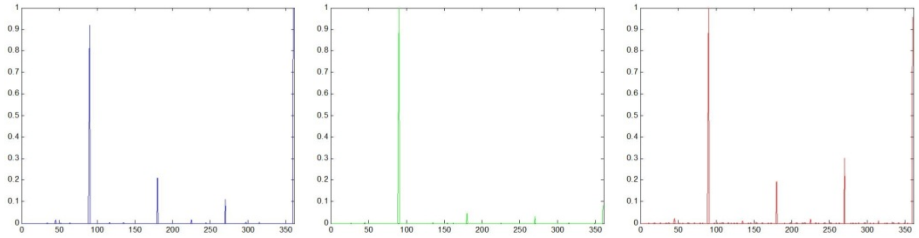


(a) Visual object 1 (b) Visual object 2 (c) Visual object 3

Figure 5.10 The visual objects localized in individual maps. (a) Visual object 1 is represented by red squares in $\mathbf{M}_1$ and $\mathbf{M}_2$. (b) Visual object 2 is represented by green squares in $\mathbf{M}_1$ and $\mathbf{M}_3$. (c) Visual object 3 is represented by a blue square in $\mathbf{M}_2$.

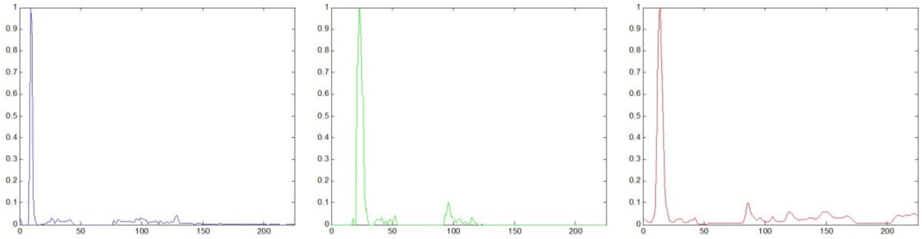
5.2.1 Results in a Parking Area

The individual maps of the parking area are shown in Fig. 5.9, which are built by SLAM with scan matching [31, 34], and the capture images when the visual objects were recognized are shown in Fig. 5.10. Their models are obtained by GPU-SIFT (Graphics Processing Unit based Scale-Invariant Feature Transform) [27], which was also used to recognize the visual objects. $\mathbf{M}_1$ and $\mathbf{M}_2$ have the visual object 1, and $\mathbf{M}_1$ and $\mathbf{M}_3$ have the visual object 2. Therefore, $\mathbf{M}_2$ and $\mathbf{M}_3$ are consecutively merged into $\mathbf{M}_1$ by the ESMM.



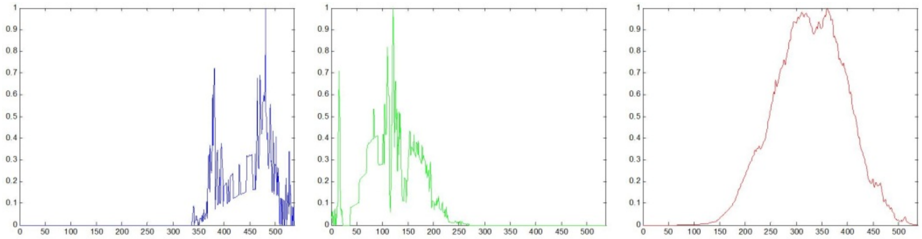
(a) $\mathbf{HS}_{\mathbf{M}_1}$ (b) $\mathbf{HS}_{\mathbf{M}_2}$ (c) $\mathbf{CC}_{\mathbf{M}_1\mathbf{M}_2}$

Figure 5.11 The enhanced Hough spectra and their cross correlation of $\mathbf{M}_1$ and $\mathbf{M}_2$. The cross-correlation function $\mathbf{CC}_{\mathbf{M}_1\mathbf{M}_2}$ was derived with the enhanced Hough spectra $\mathbf{HS}_{\mathbf{M}_1}$ and $\mathbf{HS}_{\mathbf{M}_2}$ extracted from $\mathbf{M}_1$ and $\mathbf{M}_2$, respectively.



(a) $\mathbf{XS}_{\mathbf{M}_1}$ (b) $\mathbf{XS}_{\mathbf{M}_2}$ (c) $\mathbf{CCX}_{\mathbf{M}_1\mathbf{M}_2}$

Figure 5.12 The enhanced *X*-spectra and their cross correlation of $\mathbf{M}_1$ and $\mathbf{M}_2$. The cross-correlation function $\mathbf{CCX}_{\mathbf{M}_1\mathbf{M}_2}$ was derived with the enhanced *X* spectra $\mathbf{XS}_{\mathbf{M}_1}$ and $\mathbf{XS}_{\mathbf{M}_2}$ extracted from $\mathbf{M}_1$ and $\mathbf{M}_2$, respectively.

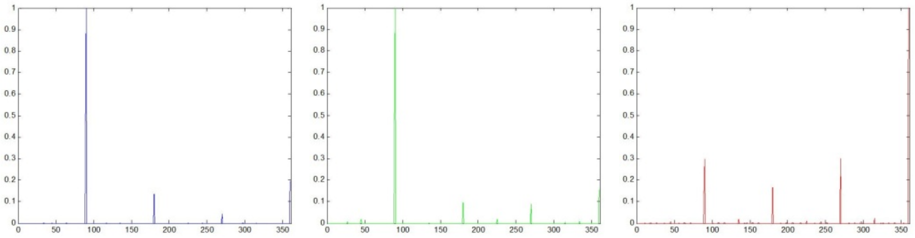


(a) $\mathbf{YS}_{\mathbf{M}_1}$ (b) $\mathbf{YS}_{\mathbf{M}_2}$ (c) $\mathbf{CCY}_{\mathbf{M}_1\mathbf{M}_2}$

Figure 5.13 The enhanced *Y*-spectra and their cross correlation of $\mathbf{M}_1$ and $\mathbf{M}_2$. The cross-correlation function $\mathbf{CCY}_{\mathbf{M}_1\mathbf{M}_2}$ was derived with the enhanced *Y* spectra $\mathbf{YS}_{\mathbf{M}_1}$ and $\mathbf{YS}_{\mathbf{M}_2}$ extracted from $\mathbf{M}_1$ and $\mathbf{M}_2$, respectively.

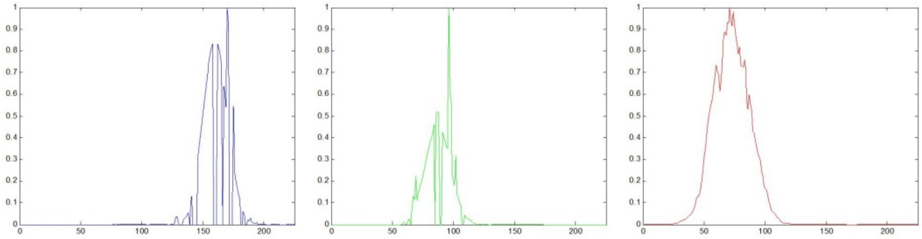
For merging $\mathbf{M}_1$ and $\mathbf{M}_2$, the ESMM extracted their enhanced Hough spectra and computed the cross correlation between them as shown in Fig. 5.11. The ESMM found the accurate rotation angle between $\mathbf{M}_1$ and $\mathbf{M}_2$ as $\Delta_\theta^{12} = 88°$ by (3.35). Then, the ESMM extracted the enhanced *X*-spectra and computed the cross correlation between the two maps as shown in Fig. 5.12. The ESMM found the accurate the *x*-translation amount between $\mathbf{M}_1$ and $\mathbf{M}_2$ as $\Delta_x^{12} = -22$ by (3.41). Finally, the ESMM extracted the enhanced *Y*-spectra and computed the cross correlation between the two maps as shown in Fig. 5.13. The ESMM found the accurate the *y*-translation amount between $\mathbf{M}_1$ and $\mathbf{M}_2$ as $\Delta_y^{12} = 355$ by (3.42). The map merging error between two maps, $\omega_E(\mathbf{M}_1, \mathbf{M}_2)$, was 0.5000, which indicates the ESMM was successful.

For merging $\mathbf{M}_1$ and $\mathbf{M}_3$, the ESMM extracted their enhanced Hough spectra and computed the cross correlation as shown in Fig. 5.14. The ESMM found the accurate rotation angle between $\mathbf{M}_1$ and $\mathbf{M}_3$ as $\Delta_\theta^{13} = 360°$ by (3.35). Then, the enhanced *X*-spectra were extracted and used to compute the cross correlations between the two maps in *x*-direction as shown in Fig. 5.15. The ESMM found the accurate the *x*-translation amount between $\mathbf{M}_1$ and $\mathbf{M}_3$ as $\Delta_x^{13} = 76$ by (3.41). Finally, the enhanced *Y*-spectra were extracted and used to compute the cross correlations between maps in *y*-direction as shown in Fig. 5.16. The ESMM found the accurate the *y*-translation amount between $\mathbf{M}_1$ and $\mathbf{M}_3$ as $\Delta_y^{13} = 50$ by (3.42). The map merging error, $\omega_E(\mathbf{M}_1, \mathbf{M}_3)$, was 0.8608 which indicates that the ESMM was performed successfully.



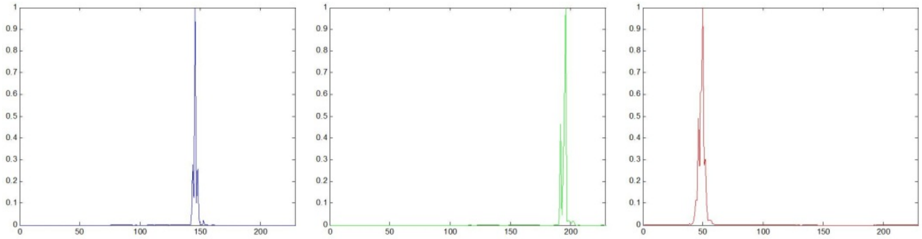
(a) $\mathbf{HS}_{\mathbf{M}_1}$ (b) $\mathbf{HS}_{\mathbf{M}_3}$ (c) $\mathbf{CC}_{\mathbf{M}_1\mathbf{M}_3}$

Figure 5.14 The enhanced Hough spectra and their cross correlation of $\mathbf{M}_1$ and $\mathbf{M}_3$. The cross-correlation function $\mathbf{CC}_{\mathbf{M_1M_3}}$ was derived with the enhanced Hough spectra $\mathbf{HS}_{\mathbf{M_1}}$ and $\mathbf{HS}_{\mathbf{M_3}}$ extracted from $\mathbf{M}_1$ and $\mathbf{M}_3$, respectively.



(a) $\mathbf{XS}_{\mathbf{M}_1}$ (b) $\mathbf{XS}_{\mathbf{M}_3}$ (c) $\mathbf{CCX}_{\mathbf{M}_1\mathbf{M}_3}$

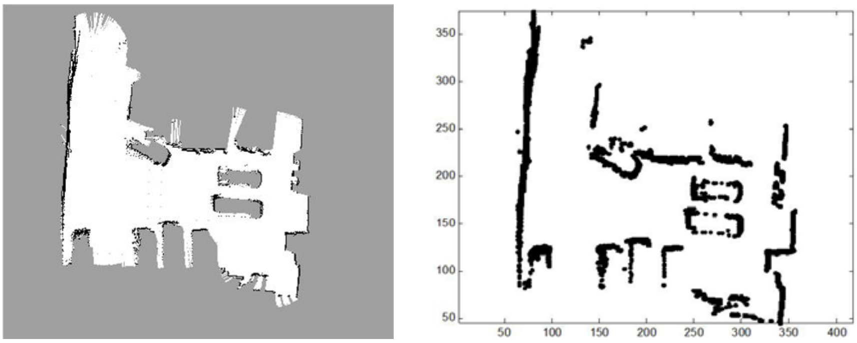
Figure 5.15 The enhanced *X*-spectra and their cross correlation of $\mathbf{M}_1$ and $\mathbf{M}_3$. The cross-correlation function $\mathbf{CCX}_{\mathbf{M_1M_3}}$ was derived with the enhanced *X* spectra $\mathbf{XS}_{\mathbf{M_1}}$ and $\mathbf{XS}_{\mathbf{M_3}}$ extracted from $\mathbf{M}_1$ and $\mathbf{M}_3$, respectively.



(a) $\mathbf{YS}_{\mathbf{M}_1}$ (b) $\mathbf{YS}_{\mathbf{M}_3}$ (c) $\mathbf{CCY}_{\mathbf{M}_1\mathbf{M}_3}$

Figure 5.16 The enhanced *Y*-spectra and their cross correlation of $\mathbf{M}_1$ and $\mathbf{M}_3$. The cross-correlation function $\mathbf{CCY}_{\mathbf{M_1M_3}}$ was derived with the enhanced *Y* spectra $\mathbf{YS}_{\mathbf{M_1}}$ and $\mathbf{YS}_{\mathbf{M_3}}$ extracted from $\mathbf{M}_1$ and $\mathbf{M}_3$, respectively.

The original form and its dual form of the final merged map of $\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{M}_3$ are successfully acquired as shown in Fig. 5.17(a) and Fig. 5.17(b), respectively. The SMM and the ESMM were applied to the different pairs instead of new maps, and the map merging errors defined by (3.29) were calculated. Table 5.3 shows the comparison result of the errors in map merging with the SMM and the ESMM. Obviously, the errors in map merging by the ESMM were smaller than the errors in map merging by the SMM. Since the enhanced spectra in the ESMM improved the accuracy of the cross-correlation between maps, the map merging errors were significantly reduced.



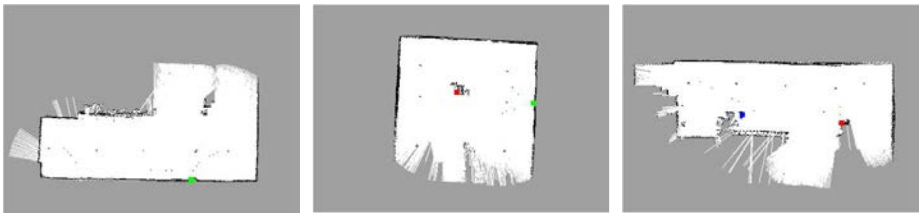
(a) Merged map by the ESMM (b) Dual form of the merged map

Figure 5.17 The final merged map of $\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{M}_3$. The final merged map and its dual form were successfully acquired by the ESMM.

TABLE 5.3

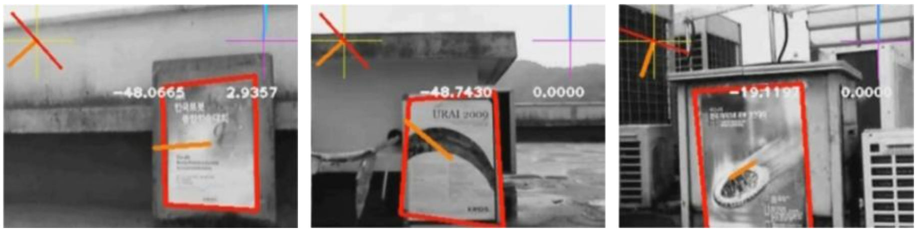
COMPARISON OF THE MAP MERGING ERRORS IN THE PARKING AREA

Type	SMM (conventional.)	ESMM (proposed.)
Merging $\mathbf{M}_1$ and $\mathbf{M}_2$	6.4033	0.4668
Merging $\mathbf{M}_1$ and $\mathbf{M}_3$	17.6489	0.4156



(a) Map 4, $\mathbf{M}_4$ (b) Map 5, $\mathbf{M}_5$ (c) Map 6, $\mathbf{M}_6$

Figure 5.18 The individual maps built by different robots which started at different locations in the building roof.

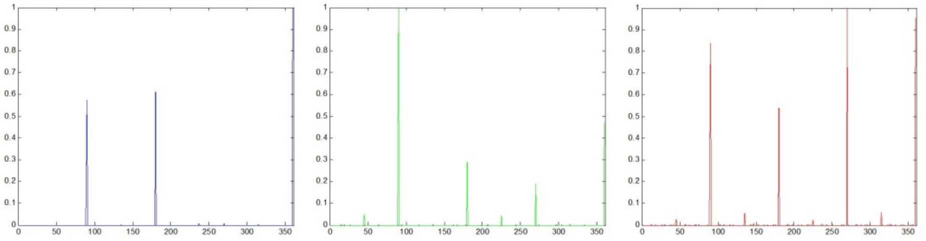


(a) Visual object 4 (b) Visual object 5 (c) Visual object 6

Figure 5.19 The visual objects localized in the individual maps of the building roof. (a) Visual object 4 is represented by green in $\mathbf{M}_4$ and $\mathbf{M}_5$. (b) Visual object 5 is represented by red in $\mathbf{M}_5$ and $\mathbf{M}_6$. (c) Visual object 6 is represented by blue in $\mathbf{M}_6$.

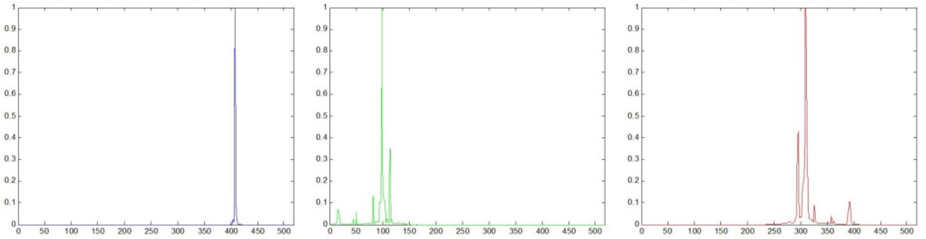
5.2.2 Results in a Building Roof

The individual maps of a building roof are shown in Fig. 5.18, and the images when the visual objects were recognized are shown in Fig. 5.19. $\mathbf{M}_4$ and $\mathbf{M}_5$ have the visual object 4, and $\mathbf{M}_5$ and $\mathbf{M}_6$ have the visual object 5. Therefore, $\mathbf{M}_6$ can be merged into $\mathbf{M}_5$ with the visual object 5. Then, $\mathbf{M}_5$ and $\mathbf{M}_6$ are together merged into $\mathbf{M}_4$ with the visual object 4. After the MTM between $\mathbf{M}_5$ and $\mathbf{M}_6$ is computed by the ESMM, the MTM between $\mathbf{M}_4$ and $\mathbf{M}_5$ is computed by the ESMM.



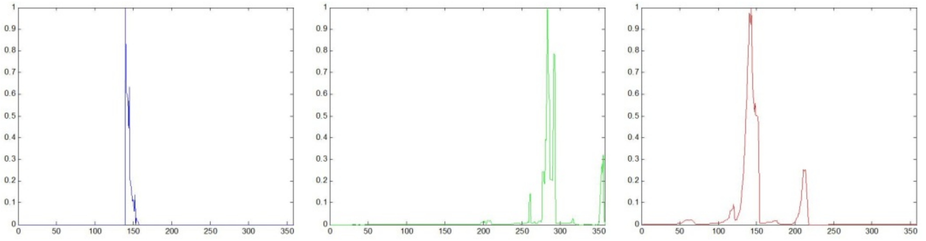
(a) $\mathbf{HS}_{\mathbf{M}_5}$ (b) $\mathbf{HS}_{\mathbf{M}_6}$ (c) $\mathbf{CC}_{\mathbf{M}_5\mathbf{M}_6}$

Figure 5.20 The enhanced Hough spectra and their cross correlation of $\mathbf{M}_5$ and $\mathbf{M}_6$. The cross-correlation function $\mathbf{CC}_{\mathbf{M}_5\mathbf{M}_6}$ was derived with the enhanced Hough spectra $\mathbf{HS}_{\mathbf{M}_5}$ and $\mathbf{HS}_{\mathbf{M}_6}$ extracted from $\mathbf{M}_5$ and $\mathbf{M}_6$, respectively.



(a) $\mathbf{XS}_{\mathbf{M}_5}$ (b) $\mathbf{XS}_{\mathbf{M}_6}$ (c) $\mathbf{CCX}_{\mathbf{M}_5\mathbf{M}_6}$

Figure 5.21 The enhanced *X*-spectra and their cross correlation of $\mathbf{M}_5$ and $\mathbf{M}_6$. The cross-correlation function $\mathbf{CCX}_{\mathbf{M}_5\mathbf{M}_6}$ was derived with the enhanced *X* spectra $\mathbf{XS}_{\mathbf{M}_5}$ and $\mathbf{XS}_{\mathbf{M}_6}$ extracted from $\mathbf{M}_5$ and $\mathbf{M}_6$, respectively.

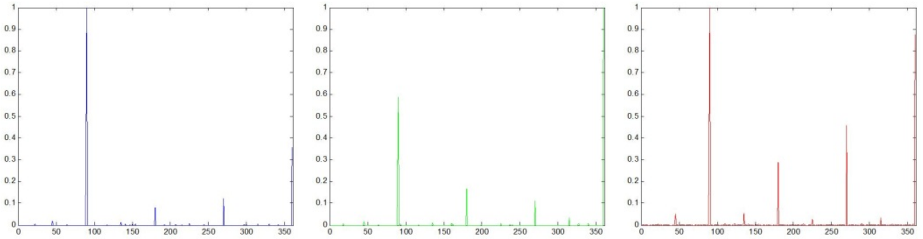


(a) $\mathbf{YS}_{\mathbf{M}_5}$ (b) $\mathbf{YS}_{\mathbf{M}_6}$ (c) $\mathbf{CCY}_{\mathbf{M}_5\mathbf{M}_6}$

Figure 5.22 The enhanced *Y*-spectra and their cross correlation of $\mathbf{M}_5$ and $\mathbf{M}_6$. The cross-correlation function $\mathbf{CCY}_{\mathbf{M}_5\mathbf{M}_6}$ was derived with the enhanced *Y* spectra $\mathbf{YS}_{\mathbf{M}_5}$ and $\mathbf{YS}_{\mathbf{M}_6}$ extracted from $\mathbf{M}_5$ and $\mathbf{M}_6$, respectively.

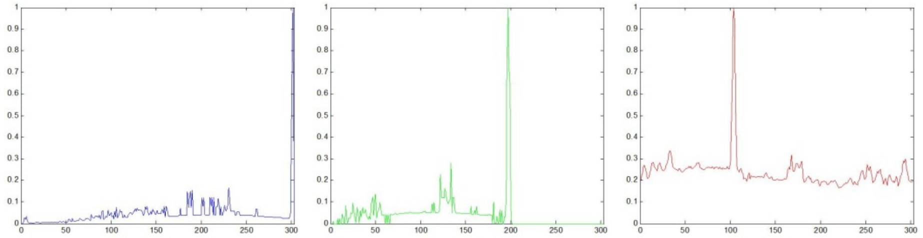
For merging $\mathbf{M}_5$ and $\mathbf{M}_6$, the ESMM extracted their enhanced Hough spectra and computed the cross correlation as shown in Fig. 5.20. The ESMM found the accurate rotation angle between $\mathbf{M}_5$ and $\mathbf{M}_6$ as $\Delta_\theta^{56} = 272°$ by (3.35). Then, the ESMM extracted the enhanced *X*-spectra and computed the cross correlation between the two maps as shown in Fig. 5.21. The ESMM found the accurate the *x*-translation amount between $\mathbf{M}_5$ and $\mathbf{M}_6$ as $\Delta_x^{56} = 309$ by (3.41). Finally, the ESMM extracted the enhanced *Y*-spectra and computed the cross correlation between the two maps as shown in Fig. 5.22. The ESMM found the accurate the *y*-translation amount between $\mathbf{M}_5$ and $\mathbf{M}_6$ as $\Delta_y^{56} = -138$ by (3.42). The map merging error, $\omega_E(\mathbf{M}_5, \mathbf{M}_6)$, was 0.1000, which indicates that the ESMM was performed successfully.

For merging $\mathbf{M}_4$ and $\mathbf{M}_5$, the ESMM extracted their enhanced Hough spectra and computed the cross correlation as shown in Fig. 5.23. The ESMM found the accurate rotation angle between $\mathbf{M}_4$ and $\mathbf{M}_5$ as $\Delta_\theta^{45} = 88°$ by (3.35). Then, the enhanced *X*-spectra were extracted and used to compute the cross correlations between the two maps in *x*-direction as shown in Fig.5.24. The ESMM found the accurate the *x*-translation amount between $\mathbf{M}_4$ and $\mathbf{M}_5$ as $\Delta_x^{45} = 105$ by (3.41). Finally, the ESMM extracted the enhanced *Y*-spectra and computed the cross correlations between maps in *y*-direction as shown in Fig. 5.25. The ESMM found the accurate the *y*-translation amount between $\mathbf{M}_4$ and $\mathbf{M}_5$ as $\Delta_y^{45} = 345$ by (3.42). The map merging error, $\omega_E(\mathbf{M}_4, \mathbf{M}_5)$, was 0.1000 which indicates that the ESMM was performed successfully.



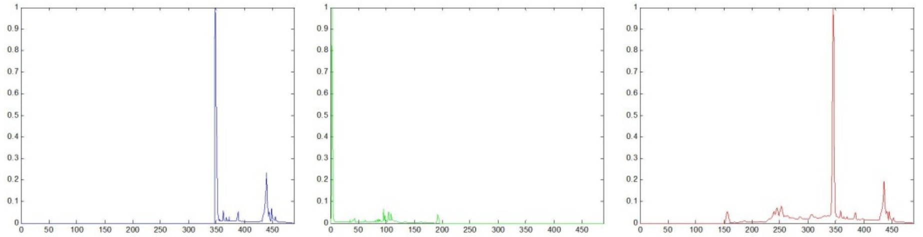
(a) $\mathbf{HS}_{\mathbf{M}_4}$ (b) $\mathbf{HS}_{\mathbf{M}_5}$ (c) $\mathbf{CC}_{\mathbf{M}_4\mathbf{M}_5}$

Figure 5.23 The enhanced Hough spectra and their cross correlation of $\mathbf{M}_4$ and $\mathbf{M}_5$. The cross-correlation function $\mathbf{CC}_{\mathbf{M_4M_5}}$ was derived with the enhanced Hough spectra $\mathbf{HS}_{\mathbf{M_4}}$ and $\mathbf{HS}_{\mathbf{M_5}}$ extracted from $\mathbf{M}_4$ and $\mathbf{M}_5$, respectively.



(a) $\mathbf{XS}_{\mathbf{M}_4}$ (b) $\mathbf{XS}_{\mathbf{M}_5}$ (c) $\mathbf{CCX}_{\mathbf{M}_4\mathbf{M}_5}$

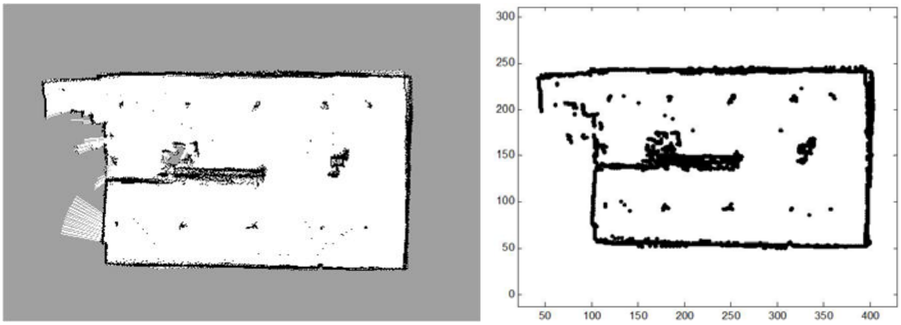
Figure 5.24 The enhanced *X*-spectra and their cross correlation of $\mathbf{M}_4$ and $\mathbf{M}_5$. The cross-correlation function $\mathbf{CCX}_{\mathbf{M_4M_5}}$ was derived with the enhanced *X* spectra $\mathbf{XS}_{\mathbf{M_4}}$ and $\mathbf{XS}_{\mathbf{M_5}}$ extracted from $\mathbf{M}_4$ and $\mathbf{M}_5$, respectively.



(a) $\mathbf{YS}_{\mathbf{M}_4}$ (b) $\mathbf{YS}_{\mathbf{M}_5}$ (c) $\mathbf{CCY}_{\mathbf{M}_4\mathbf{M}_5}$

Figure 5.25 The enhanced *Y*-spectra and their cross correlation of $\mathbf{M}_4$ and $\mathbf{M}_5$. The cross-correlation function $\mathbf{CCY}_{\mathbf{M_4M_5}}$ was derived with the enhanced *Y* spectra $\mathbf{YS}_{\mathbf{M_4}}$ and $\mathbf{YS}_{\mathbf{M_5}}$ extracted from $\mathbf{M}_4$ and $\mathbf{M}_5$, respectively.

The original form and its dual form of the final merged map of $\mathbf{M}_4$, $\mathbf{M}_5$, and $\mathbf{M}_6$ is successfully acquired as shown in Fig. 5.26(a) and Fig. 5.26(b), respectively. The SMM and the ESMM were applied to the different pairs instead of new maps, and the map merging errors defined by (3.29) were calculated. The comparison result with the SMM and the ESMM is shown in Table 5.4. Obviously, the errors in map merging by the ESMM were smaller than the errors in map merging by the SMM. This was because the enhanced spectra in the ESMM improved the accuracy of the cross-correlation.



(a) Merged map by the ESMM (b) Dual form of the merged map

Figure 5.26 The final merged map of $\mathbf{M}_4$, $\mathbf{M}_5$, and $\mathbf{M}_6$. The final merged map and its dual form were successfully acquired by the ESMM.

TABLE 5.4

COMPARISON OF THE MAP MERGING ERRORS IN THE BUILDING ROOF

Type	SMM (conventional.)	ESMM (proposed.)
Merging $\mathbf{M}_5$ and $\mathbf{M}_6$	15.6404	1.0349
Merging $\mathbf{M}_4$ and $\mathbf{M}_5$	18.8676	0.8617

5.3 Experimental Results of the TMM

The TMM has been tested to verify its effectiveness in two types of real environments and a public dataset. The used public dataset was RADISH Fort AP Hill dataset [74]. In each real environment, a global map was built by our SLAM techniques [23] based on mobile robots with laser scan sensors and a wireless router as shown in Fig. 5.27. The used robot was Pioneer3-DX, and the used laser scan sensor was Hokuyo UTM-30LX. Then, various different pairs of partial maps were obtained from the global map to test our map merging algorithm. Because we do not consider communication among robots, this experimental approach is sufficient to verify the performance of our map merging algorithm. The various pairs of partial maps were merged by the SMM and the TMM without any inter-robot measurements. Their acceptance indices were computed to compare the performance of map merging.

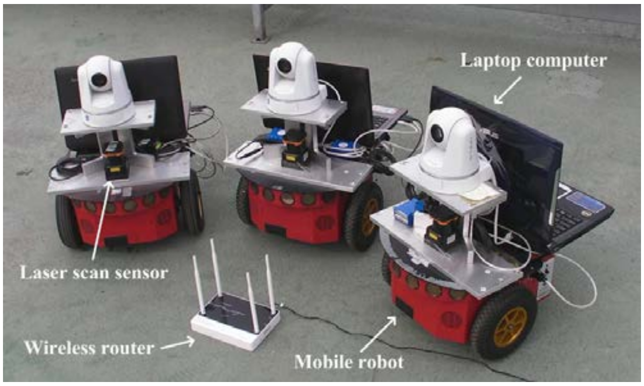


Figure 5.27 The hardware system for the TMM. Each robot performs SLAM using a laser scan sensor. Then, the estimated robot poses and local maps are sent to one of the robots via a wireless router to build a global map.

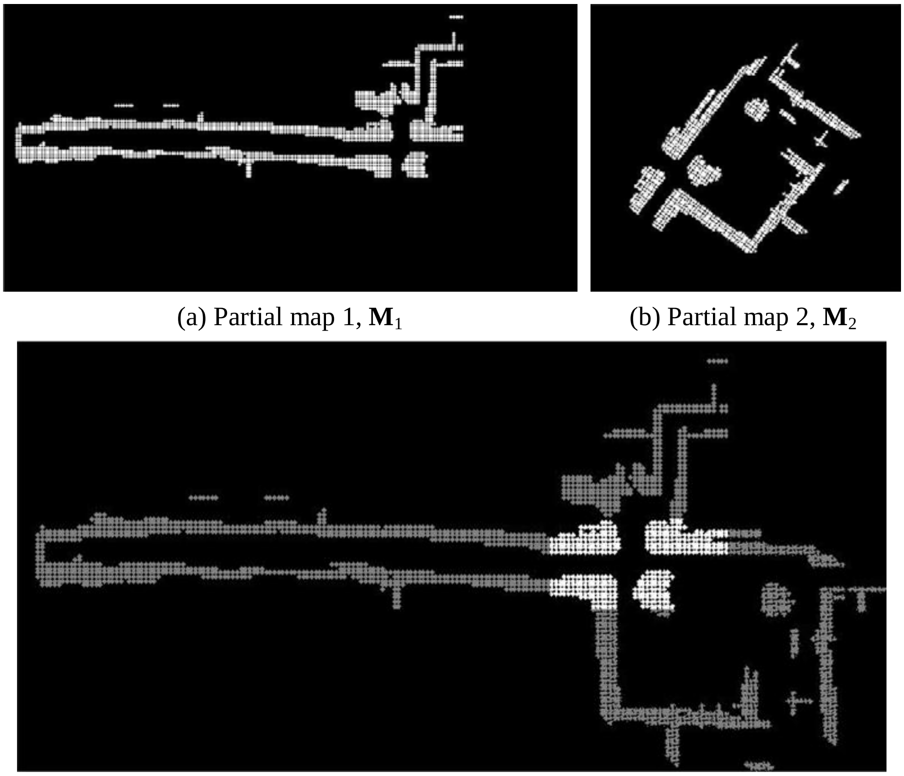
TABLE 5.5

DIFFERENT TYPES OF PARTIAL MAPS FOR INDOOR EXPERIMENTS

Type	Number of Occupied Grids in $\mathbf{M}_1$	Number of Occupied Grids in $\mathbf{M}_2$	Number of Commonly Occupied Grids	Ratio of the Overlapping Region to $\mathbf{M}_1$	Ratio of the Overlapping Region to $\mathbf{M}_2$
1	36924	29122	8673	23%	30%
2	36924	22542	5378	15%	24%
3	36924	25648	6934	19%	27%
4	35926	29138	8174	23%	28%
5	34132	29138	7306	21%	25%
6	34132	25648	5777	17%	23%
7	34132	22542	4221	12%	19%
8	35926	25648	6435	18%	25%
9	35926	22542	4879	14%	22%
10	39830	22542	6831	17%	30%

5.3.1 Results in Indoor Environments

The proposed map merging technique, TMM, has been tested in an indoor environment which is the third floor of Automation and Systems Research Institute at Seoul National University in South Korea. The size of the indoor environment is about 48m×17m, and the grid size of each map was 20cm×20cm. The partial maps and the global map which have already used as an illustrative example in Section 4 were obtained from this environment. More types of partial maps with different sizes, different shapes, and different overlapping regions are obtained as shown in Table 5.5. In each pair, $\mathbf{M}_4$ was rotated and translated by ninety different map transformation matrices. The range of rotation angle was 0º to 90º, and *x*-*y* translations were randomly obtained. Thus, the number of datasets to evaluate the performance of the TMM was nine hundred.



(a) Partial map 1, $\mathbf{M}_1$

(b) Partial map 2, $\mathbf{M}_2$

(c) Merged map of $\mathbf{M}_1$ and $\mathbf{M}_2$

Figure 5.28 The result of merging two partial maps, $\mathbf{M}_1$ and $\mathbf{M}_2$, when the worst acceptance index occurred. Here, $\mathbf{M}_2$ was rotated by Δ_θ and translated by Δ_x and Δ_y. In the worst case, the two partial maps were successfully merged by the TMM. In the merged map, the overlapping regions of the two partial maps are represented by white cells.

The TMM was successfully applied to all the datasets. The worst case index occurred with one of the datasets in Type 7 which has the smallest overlapping region among the ten types. However, the result of map merging was good as shown in Fig. 5.28. Its acceptance index was 0.9628 which was higher than any other acceptance indices obtained by the SMM. To evaluate and compare the performance of the map merging techniques, the average acceptance index over different ninety transformations was computed for each type in Table 5.5.

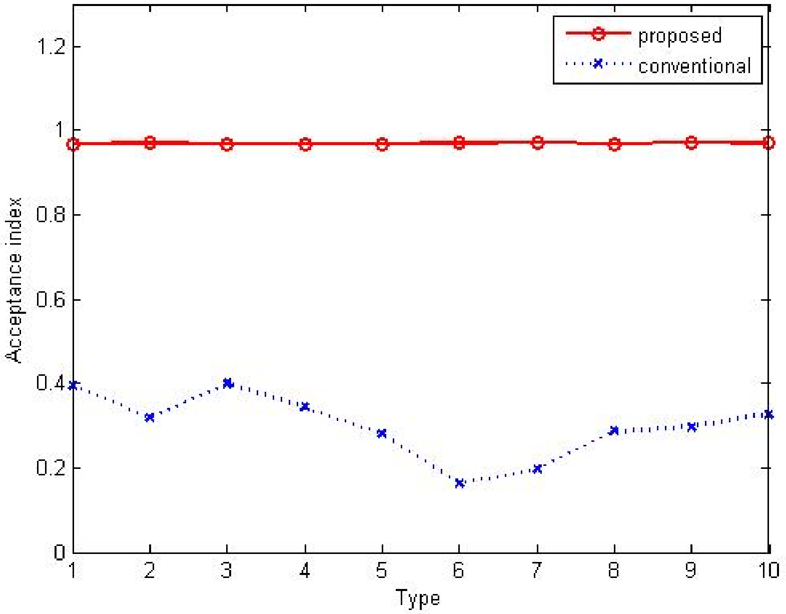


Figure 5.29 Average acceptance indices for different types of the partial maps in Table 5.5. Each value was averaged over different ninety transformations. For all types, the acceptance indices of the proposed technique, TMM, were higher than those of the conventional technique.

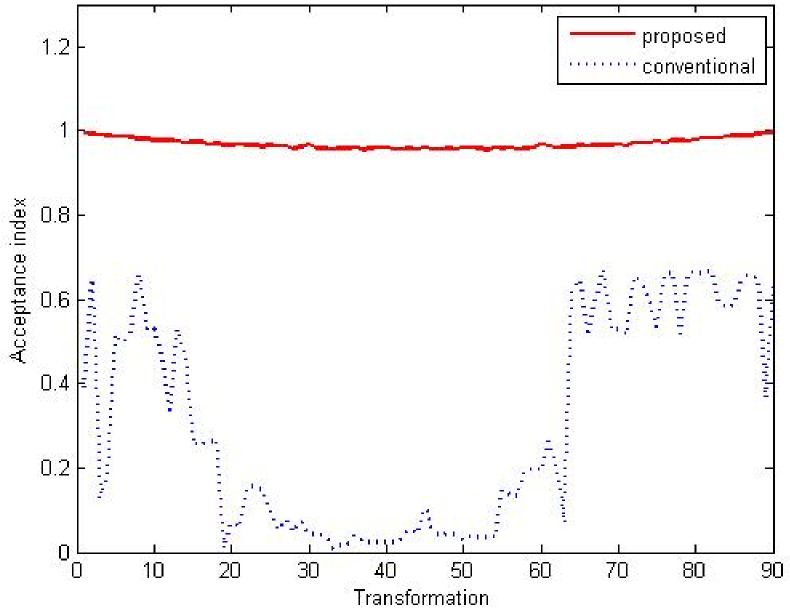
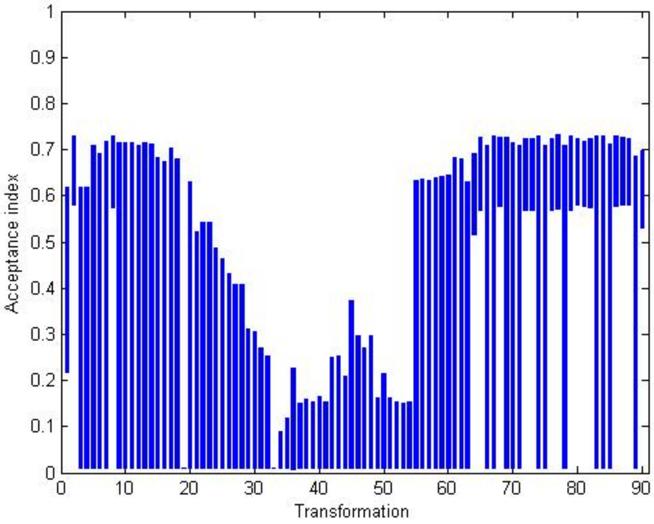
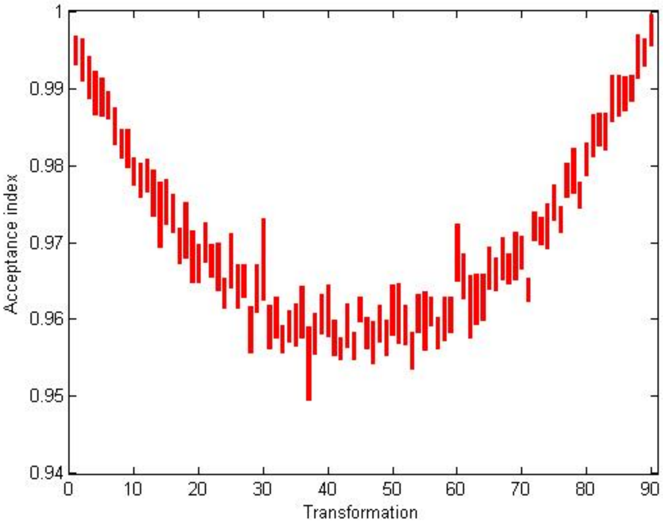


Figure 5.30 Average acceptance indices according to different transformations between the partial maps. Each value was averaged over different ten types of partial maps in Table 5.5. For all transformations, the proposed technique, TMM, showed higher acceptance indices than the conventional technique.

Figure 5.29 shows the comparison result of the average acceptance indices for different types of the partial maps. Obviously, the proposed map merging technique, TMM, was better than the conventional map merging technique, SMM. Especially, the TMM showed consistently high acceptance indices, which was different from the SMM. For the more precise comparison, average acceptance indices according to different transformations were computed as shown in Fig. 5.30. Each value was averaged over different ten types of partial maps in Table 5.5.



(a) Conventional map merging technique, SMM



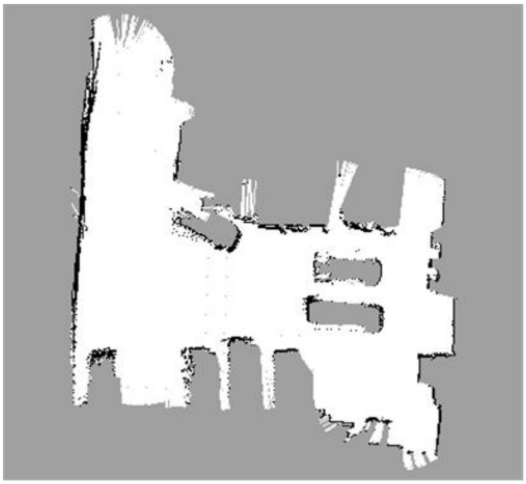
(b) Proposed map merging technique, TMM

Figure 5.31 The ranges of acceptance indices over different ten types in Table 5.5 according to different transformations. The range of acceptance indices with the proposed map merging technique, TMM, was shorter than the range of acceptance indices with the conventional map merging technique, SMM.

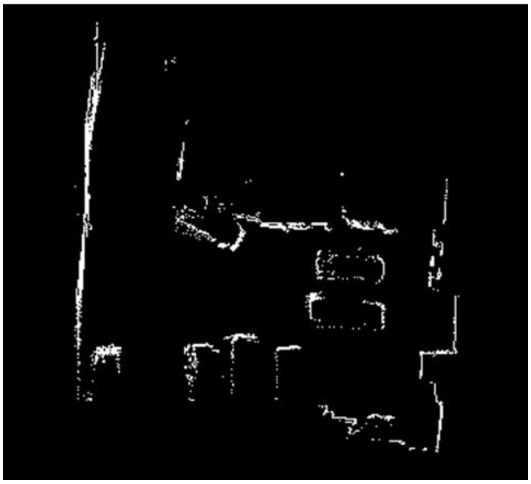
Evidently, the TMM was better than the SMM. Moreover, the TMM showed high acceptance indices regardless of transformation, which was different from the SMM. Figure 5.31 shows the ranges of acceptance indices for each transformation. Note that the scales of acceptance index are different. The range of acceptance indices obtained by the TMM was consistently small, but that of acceptance indices obtained by the SMM fluctuated.

5.3.2 Results in Outdoor Environments

The other experimental environment to verify the performance of the TMM was an outdoor parking lot of Automation and Systems Research Institute at Seoul National University in South Korea. Figure 5.32(a) shows the whole map of the outdoor environment whose size is about 25m×30m. Differently from the grid map of the indoor environment, the grid map of the outdoor environment consists of *occupied* black grids, *unoccupied* white grids, and *unknown* gray grids. The *unknown* portion of the grid map may cause occlusion, which degenerates the performance of map merging. Therefore, this type of grid maps is converted to the type used in the indoor environment, which consists of *occupied* white grids and *unoccupied* black grids as shown in Fig. 5.32(b). Similarly to the case of indoor experiments, more types of partial maps with different sizes, different shapes, and different overlapping regions are obtained as shown in Table 5.6. In each pair, $\mathbf{M}_4$ was rotated and translated by ninety different map transformation matrices. The range of rotation angle was 0º to 90º, and *x*-*y* translations were randomly obtained. Thus, the number of datasets was nine hundred.



(a) Original outdoor map



(b) Converted outdoor map

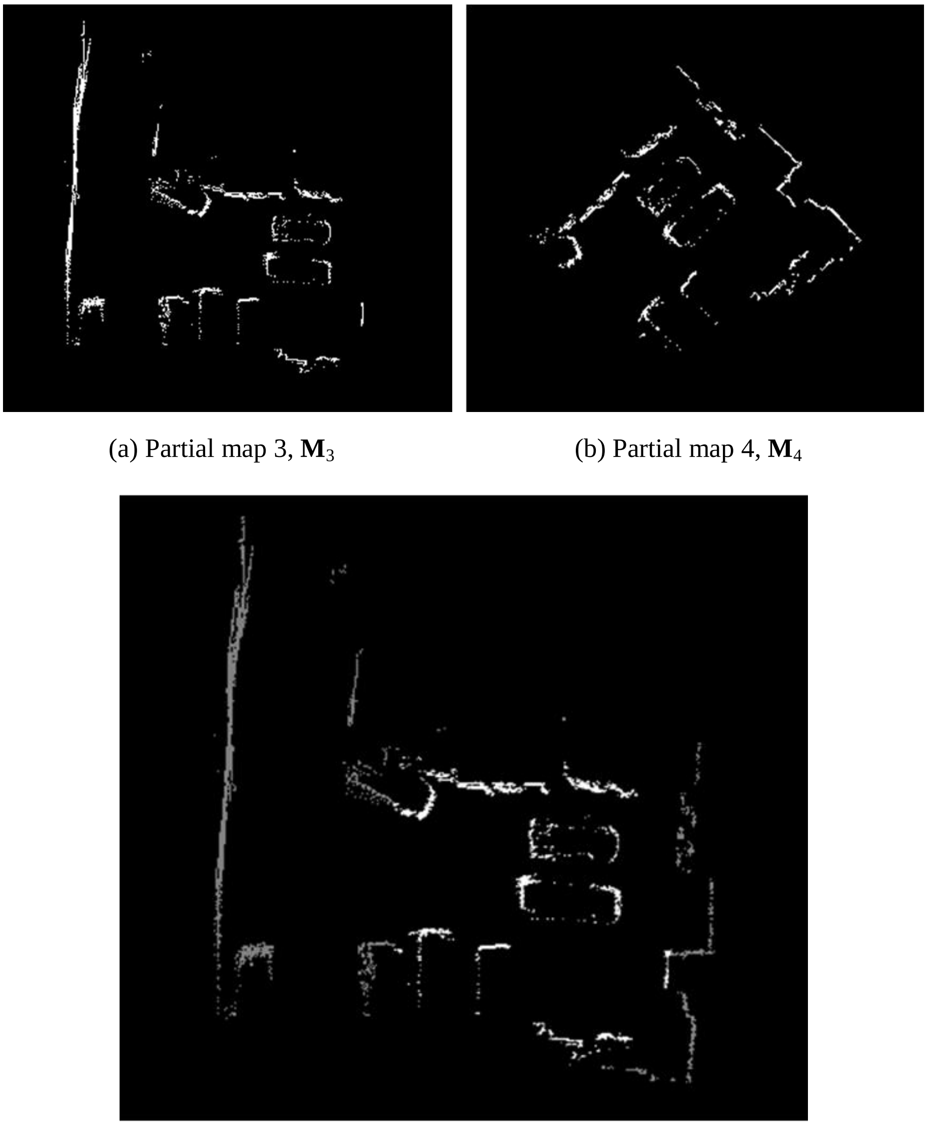
Figure 5.32 The whole grid map of an outdoor environment. (a) The original map of the outdoor environment consists of *occupied* black grids, *unoccupied* white grids, and *unknown* gray grids. (b) The original map is converted to the other type of a grid map which consists of *occupied* black grids, *unoccupied* white grids.

TABLE 5.6

DIFFERENT TYPES OF PARTIAL MAPS FOR OUTDOOR EXPERIMENTS

Type	Number of Occupied Grids in $\mathbf{M}_3$	Number of Occupied Grids in $\mathbf{M}_4$	Number of Commonly Occupied Grids	Ratio of the Overlapping Region to $\mathbf{M}_3$	Ratio of the Overlapping Region to $\mathbf{M}_4$
11	4512	2850	922	20%	32%
12	4820	2850	1053	22%	37%
13	4586	2850	936	20%	33%
14	4310	2850	892	21%	31%
15	4456	2850	913	21%	32%
16	4556	2992	993	22%	33%
17	4534	2940	970	21%	33%
18	4678	2992	1063	23%	35%
19	4730	2992	1073	23%	36%
20	4964	2992	1190	24%	40%

The TMM was successfully applied to all the outdoor datasets. Although the worst case index occurred with one of the datasets in Type 14 which has the smallest overalpping region among the ten types, the result of map merging was good as shown in Fig. 5.33. Its acceptance index was 0.7347 which was higher than any other acceptance indices of the SMM. To evaluate the performance of the map merging techniques which are the SMM and the TMM, the average acceptance index over different ninety transformations was computed for each type in Table 5.6.



(a) Partial map 3, $\mathbf{M}_3$

(b) Partial map 4, $\mathbf{M}_4$

(c) Merged map of $\mathbf{M}_3$ and $\mathbf{M}_4$

Figure 5.33 The result of merging two partial maps, $\mathbf{M}_3$ and $\mathbf{M}_4$, when the worst acceptance index occurred. Here, $\mathbf{M}_3$ was rotated by Δ_θ and translated by Δ_x and Δ_y. In the worst case, the two partial maps were successfully merged by the sinogram-based map merging algorithm. In the merged map, the overlapping regions of the two partial maps are represented by white cells.

Figure 5.34 shows the comparison result of the average acceptance indices for different types of the partial maps. Obviously, the proposed map merging technique, TMM, was consistently better than the conventional map merging technique, SMM. However, the average acceptance indices in outdoor experiments were lower than those in indoor experiments, which were caused by the map error. In outdoor environments, the mapping error occurs due to uneven terrain and sensing errors more frequently than indoor environments.

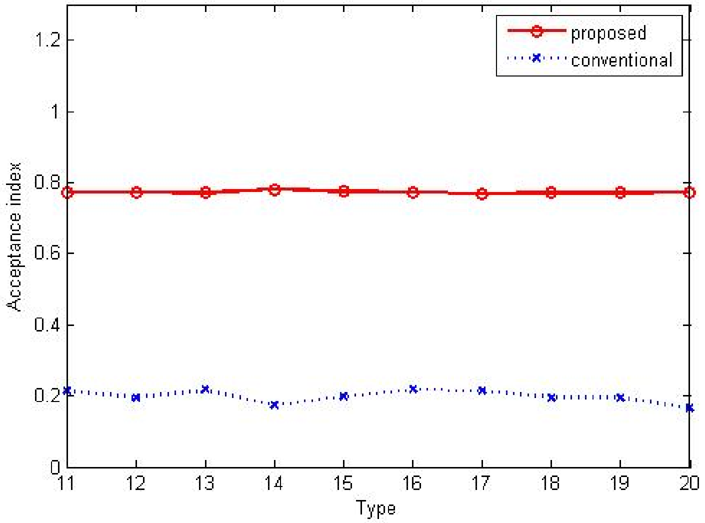


Figure 5.34 Average acceptance indices for different types of the partial maps in Table 5.6. Each value was averaged over different ninety transformations. For all types, the acceptance indices of the proposed map merging technique, TMM, were higher than those of the conventional map merging technique, SMM.

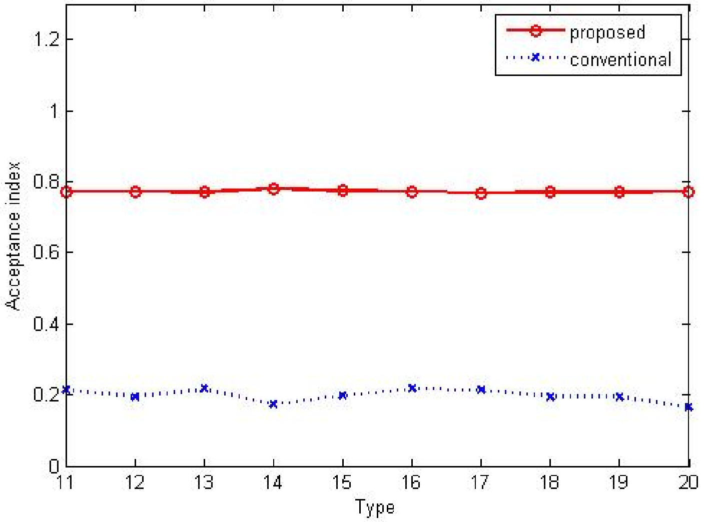
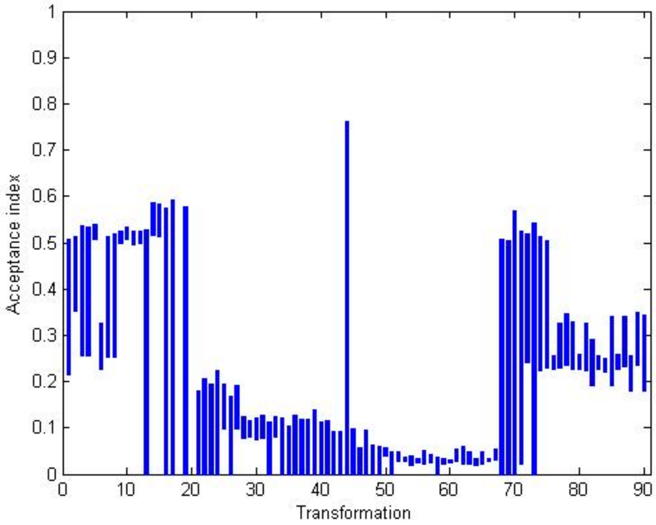
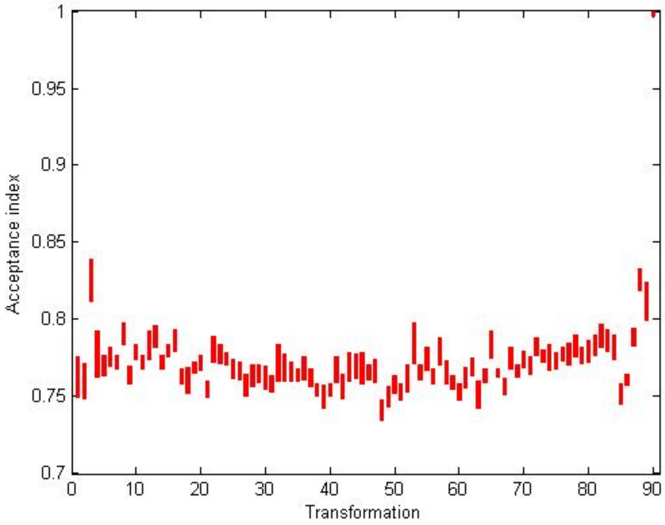


Figure 5.35 Average acceptance indices according to different transformations between the partial maps. Each value was averaged over different ten types of partial maps in Table 5.6. For all transformations, the proposed technique, TMM, showed higher acceptance indices than the conventional technique, SMM.

For the more precise comparison, average acceptance indices according to different transformations were computed as shown in Fig. 5.35. Each value was averaged over different ten types of partial maps in Table 5.6. Obviously, the TMM showed consistently higher acceptance indices regardless of transformation than the SMM. Figure 5.36 shows the ranges of acceptance indices for each transformation in indoor experiments. Note that the scales of acceptance index are different. The range of acceptance indices obtained by the TMM was consistently small, but that of acceptance indices obtained by the SMM fluctuated. Therefore, the TMM is more stable than the SMM.



(a) Conventional map merging technique, SMM



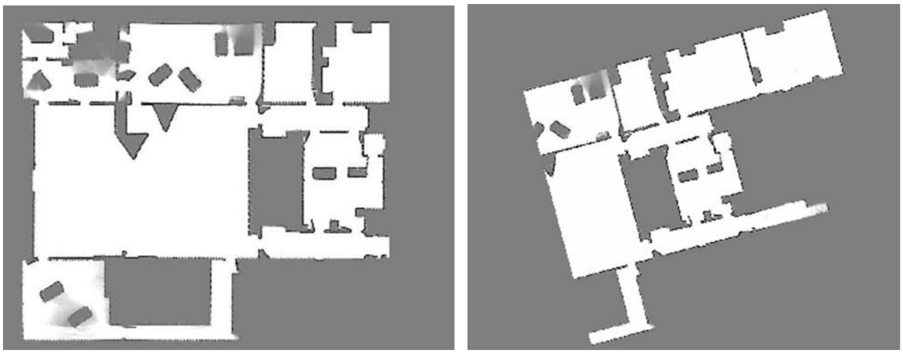
(b) Proposed map merging technique, TMM

Figure 5.36 The ranges of acceptance indices over different ten types in Table 5.6 according to different transformations. The range of acceptance indices with the proposed map merging technique, TMM, was shorter than the range of acceptance indices with the conventional map merging technique, SMM.

TABLE 5.7

THE SIZE OF THE OVERLAPPING AREAS OF DIFFERENT PAIRS OF MAPS OBTAINED BY A PUBLIC DATASET

Type	Ratio of Overlapping Region to $\mathbf{M}_5$	Ratio of Overlapping Region to $\mathbf{M}_6$	Average Ratio of the Overlapping Area
21	85.3%	95.4%	90.3%
22	79.2%	91.8%	85.5%
23	74.2%	88.3%	81.3%
24	66.5%	83.6%	75.0%
25	61.4%	80.0%	70.7%
26	54.4%	76.9%	65.7%
27	49.5%	73.9%	61.7%
28	44.1%	71.6%	57.9%
29	41.4%	62.7%	52.1%
30	39.5%	57.6%	48.6%



(a) Partial map 5, $\mathbf{M}_5$ (b) Partial map 6, $\mathbf{M}_6$

Figure 5.37 An example of two partial maps of the whole map obtained by RADISH AP Hill dataset [74]. There is an overlapped area between the two partial maps.

5.3.3 Results with a Public Dataset

Ten different pairs of maps were manually produced from the whole map obtained by RADISH Fort AP Hill dataset as shown in 5.7. An example of the pair of two partial maps is shown in Fig. 5.37. The size of the overlapping area decreases gradually according to Type 21 through 30. The TMM was compared with the SMM [9], a local feature descriptor matching method, and an image registration method. The local feature descriptor matching was conducted as follows. SURF [19] was applied to maps to find local features and extract descriptors. The MTM was estimated by matching the feature descriptors based on RANSAC (RANdom SAmple Consensus) algorithm [75]. The image registration method was conducted by optimizing the similarity metric based on MSE (Mean Squared Error), which was implemented using MATLAB image processing toolbox. The SURF-based map merging method and the MSE-based map merging method cannot be applied to map merging in real environments due to the excessive distortions caused by sensor errors.

The comparison result of acceptance indices computed by (4.3) along the types shown in Table 5.7 is shown in Fig. 5.38. The TMM shows consistently high acceptance indices regardless of the types. The acceptance index of the SMM was consistently high from Type 21 to 26. But, when the overlapping region became smaller than a certain quantity as shown from Type 27 to 30, the acceptance index became smaller and inconsistent. The acceptance index of the SURF feature matching was consistently high from Type 21 to 25.

However, after that, the acceptance index became smaller. The MSE-based image registration method shows the worst acceptance index except for Type 21. Average acceptance indices and computation times are summarized in Table 5.8. The proposed algorithm showed the highest acceptance index with similar computation time with SMM and SURF feature matching.

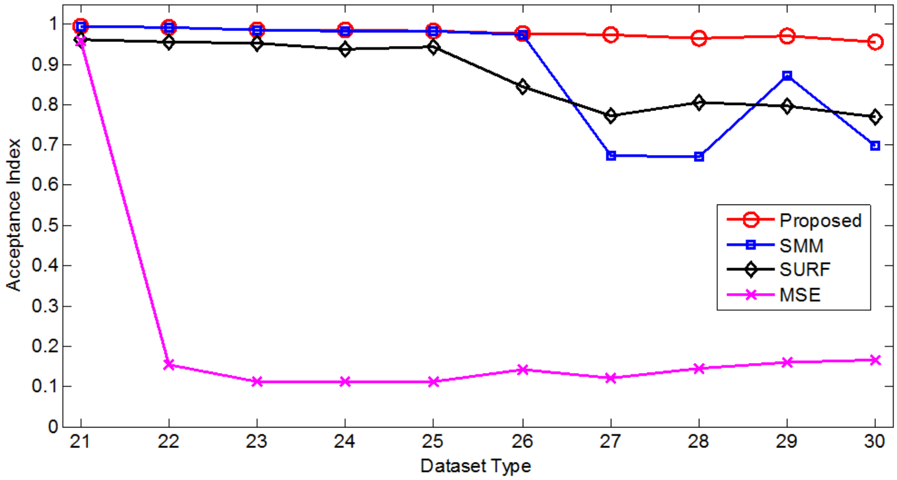
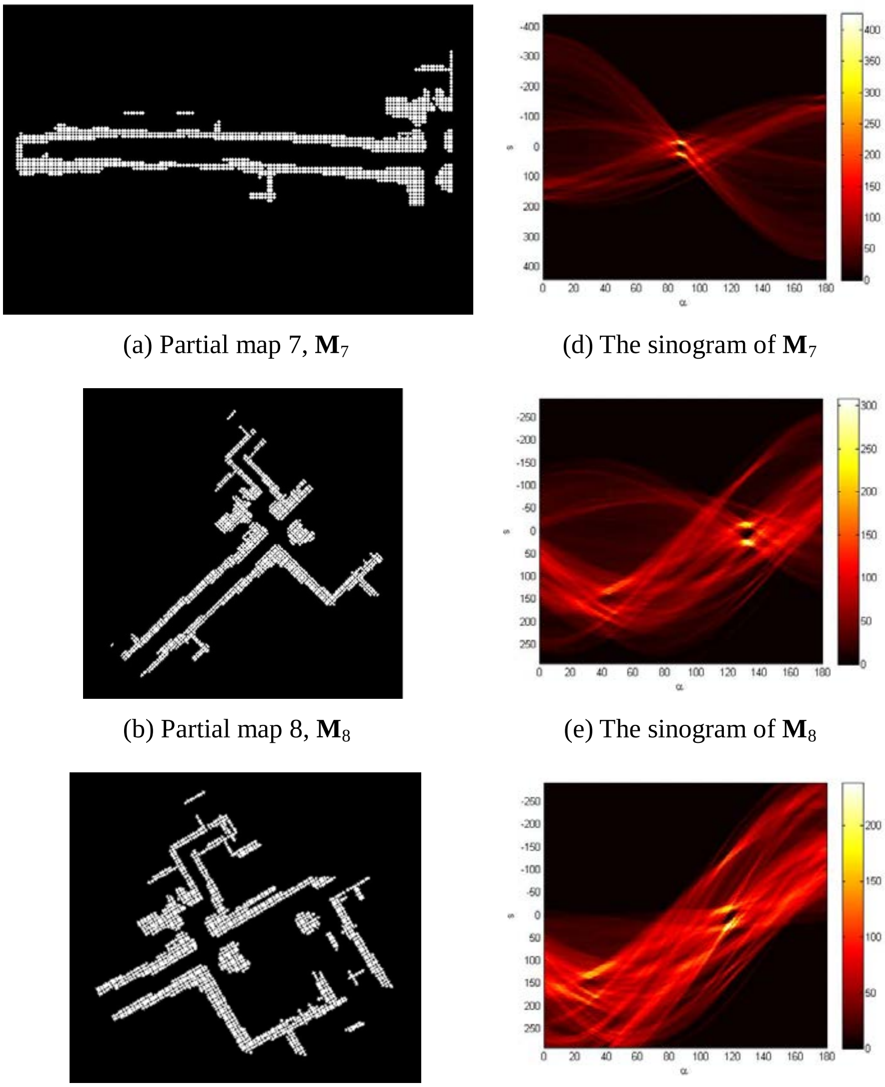


Figure 5.38 The comparison of the acceptance indices in experiments with different pairs of partial maps in Table 5.7. The proposed algorithm shows consistently high acceptance indices even though the size of overlapping areas decreases.

TABLE 5.8

COMPARISON OF AVERAGE ACCEPTANCE INDEX AND COMPUTATION TIME IN EXPERIMENTS WITH A PUBLIC DATASET

Type	Proposed. (TMM)	SMM	SURF	MSE
Acceptance index	0.9783	0.8822	0.8746	0.2187
Computation time (s)	3.5100	3.3228	3.0732	63.2584



(a) Partial map 7, $\mathbf{M}_7$

(d) The sinogram of $\mathbf{M}_7$

(b) Partial map 8, $\mathbf{M}_8$

(e) The sinogram of $\mathbf{M}_8$

(c) Partial map 9, $\mathbf{M}_9$

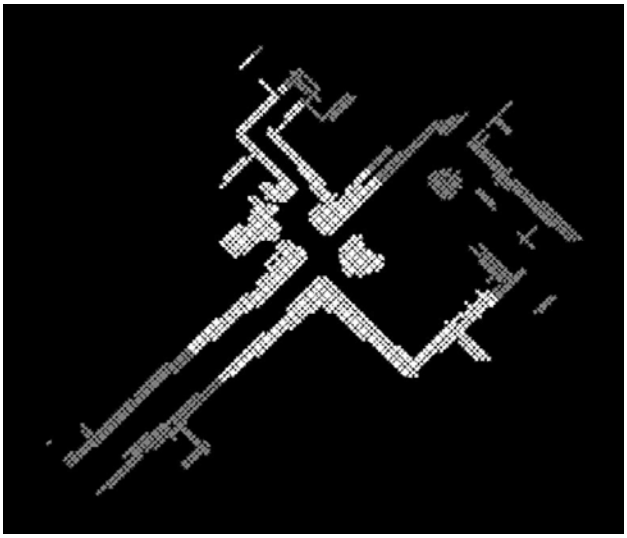
(f) The sinogram of $\mathbf{M}_9$

Figure 5.39 Three partial maps of the indoor environment and their sinograms. Although $\mathbf{M}_7$ and $\mathbf{M}_9$ have insufficient common region, they can be merged through $\mathbf{M}_8$. By the proposed map merging technique, TMM, after $\mathbf{M}_9$ is merged into $\mathbf{M}_8$, $\mathbf{M}_8$ and $\mathbf{M}_9$ are together merged into $\mathbf{M}_7$.

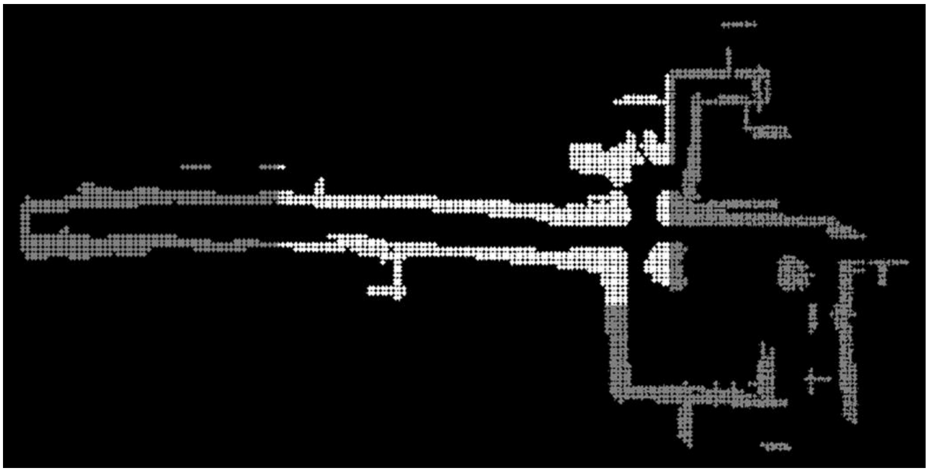
5.3.4 Results of Merging More Maps

The proposed map merging technique, TMM, can be applied to merge not only two partial maps but also more than three partial maps in successive rounds. Three different partial maps of the same environment are shown in Fig. 5.39(a)-(c). The sinograms of the partial maps are shown in Fig. 5.39(d)-(f). Because $\mathbf{M}_7$ and $\mathbf{M}_9$ have insufficient common region, they cannot be merged directly. Instead, $\mathbf{M}_7$ and $\mathbf{M}_9$ can be successfully merged through $\mathbf{M}_8$. The detailed process of applying the TMM to merging three partial maps is as follows.

Firstly, $\mathbf{M}_9$ is merged into $\mathbf{M}_8$ as shown in Fig. 5.40(a). The overlapping region between $\mathbf{M}_8$ and $\mathbf{M}_9$ is represented by white grids. The information on their map merging is summarized in Table 5.9. The acceptance index between $\mathbf{M}_8$ and $\mathbf{M}_9$ was sufficiently high. Then, the map resulting from the first map merging is successively merged into $\mathbf{M}_7$ as shown in Fig. 5.40(b). The information on their map merging is summarized in Table 5.10. The acceptance index between $\mathbf{M}_7$ and $\mathbf{M}_8$ was also sufficiently high. Consequently, the three partial maps were successfully merged by the proposed map merging algorithm in successive rounds. In other words, the lack of the overlapping area between $\mathbf{M}_7$ and $\mathbf{M}_9$ can be compensated by the successive rounds through $\mathbf{M}_8$. Obviously, the coordinate system used for the last map merging becomes equivalent to the coordinate system used for the final merged map.



(a) The result of merging **M**$_9$ into **M**$_8$



(b) The result of merging **M**$_8$ and **M**$_9$ into **M**$_7$

Figure 5.40 The result of merging the three partial maps of the indoor environment. (a) The overlapping area between **M**$_8$ and **M**$_9$ is represented with white grids. (b) The overlapping area between **M**$_7$ and **M**$_8$ is represented with white grids. The three partial maps are merged sequentially by the TMM.

TABLE 5.9

DIFFERENT PARTIAL MAPS, $\mathbf{M}_8$ AND $\mathbf{M}_9$

Number of Occupied Grids in $\mathbf{M}_8$	43106
Number of Occupied Grids in $\mathbf{M}_9$	34398
Number of Commonly Occupied Grids	12128
Ratio of the Common Region to $\mathbf{M}_8$	28%
Ratio of the Common Region to $\mathbf{M}_9$	35%
Acceptance Index between $\mathbf{M}_8$ and $\mathbf{M}_9$	0.9053

TABLE 5.10

DIFFERENT PARTIAL MAPS, $\mathbf{M}_7$ AND $\mathbf{M}_8$

Number of Occupied Grids in $\mathbf{M}_7$	29264
Number of Occupied Grids in $\mathbf{M}_8$	43106
Number of Commonly Occupied Grids	9832
Ratio of the Common Region to $\mathbf{M}_7$	34%
Ratio of the Common Region to $\mathbf{M}_8$	23%
Acceptance Index between $\mathbf{M}_7$ and $\mathbf{M}_8$	0.9764

5.4 Comparison among the Proposed Techniques

Since the SFMM, the ESMM and the TMM have their own advantages over one another, the absolute comparison among them is not appropriate. Because the SFMM is conducted with sparsely distributed feature points, the computation cost is low. But, it cannot be conducted in the environments where infrared features are rarely extracted such as outdoor environments. Because the ESMM is conducted with the estimated locations of the visual objects, the risk of being caught on local maxima decreases. But, it cannot be conducted in the environments without visual object or their substitutes. The TMM can be conducted without any help from infrared features or visual objects, but it needs a certain degree of the overlapping area between maps.

Nevertheless, the coarse comparison of the accuracy of them is possible in the environment where the three map merging techniques can be commonly applied with several modifications. The indoor environment used for the experiments of the SFMM is selected because of the existence of the infrared features. Two individual maps, $\mathbf{M}_{10}$ and $\mathbf{M}_{11}$, were built by two robots. The visual objects are substituted by the other robots, and the map enhancement in the ESMM is performed when one of the robots perceives another robot. The perception process in each robot was conducted using histograms of oriented gradients (HoG) features and support vector machine (SVM). The formulation to measure the map merging error defined in (3.29) was commonly used as a performance index because it can be more generally applied than the performance indices such as (3.8) and (4.3). The comparison result of the map merging errors of the SFMM, the ESMM and the TMM is summarized in Table 5.11. Obviously, the proposed techniques were successfully conducted, and the difference among their map merging errors was not significant.

TABLE 5.11

COMPARISON OF MAP MERGING ERRORS OF THE PROPOSED TECHNIQUES

Type	SFMM	ESMM	TMM
Merging $\mathbf{M}_{10}$ and $\mathbf{M}_{11}$	0.0500	0.1097	0.1000

5.5 Discussion

Under unknown initial correspondences among robots, correlation-based map merging techniques presented in this dissertation can reduce the amount of the overlapping regions for map matching. Especially, the TMM was successfully performed with relatively small overlapping regions, compared with the conventional map merging technique. Nevertheless, the size of the overlapping regions needs to be discussed because it is highly crucial for map merging with unknown initial correspondences among robots.

The size of the overlapping regions required for the TMM depends on the accuracy of individual maps. In indoor experiments described in Section 5.3.1, the average ratio of the overlapping regions to each map was about 21.55%. However, in outdoor experiments described in Section 5.3.2, the average ratio of the overlapping regions to each map was about 27.96%. This increase indicates that the more overlapping regions were needed to perform the proposed algorithm successfully. Generally, the accuracy of building maps in outdoor environments is lower than indoor environments due to uneven terrain and measurement errors. Because the map errors disturb the extraction of proper sinograms, the map merging with the proposed algorithm needs more overlapping regions than indoor environments. Moreover, because of the inevitable errors, the acceptance indices in outdoor experiments were lower than the acceptance indices in indoor experiments even though more overlapping regions were given. In Section 5.3.4, the average ratios of the overlapping regions to each map in the first and the second map merging as shown in Fig. 5.40 were about 31.70% and 28.20%, respectively. Although the average ratios were similar, the acceptance index between $\mathbf{M}_7$ and $\mathbf{M}_8$ was higher than the acceptance index between $\mathbf{M}_8$ and $\mathbf{M}_9$ in spite of the similar amount of overlapping regions. This result indicates that the performance of map merging is affected by not only the amount of overlapping regions but also the shape of overlapping regions.

Correlation-based map merging can be applied to merging the other types of maps such as appearance maps [63], RGB-D maps [65], and 3D maps [66]-[68]. It can be extended to stochastic map merging by applying probabilistic matching algorithms [43, 44] and fingerprint matching algorithms [38, 69]. Also, it can be extended to cooperative active map merging by applying local motion planning techniques [32, 33, 36, 37].

Chapter 6

Conclusions

This dissertation presents novel map merging techniques for multi-robot SLAM, which do not need both the knowledge of the relative initial poses among robots and the direct robot-to-robot measurements. Firstly, this dissertation presents two extensions of the spectrum-based map merging (SMM) which is a recently developed grid map merging technique. Secondly, this dissertation presents a new grid map merging technique, which does not require any information on the initial poses among robots or any robot-to-robot measurements.

Feature map merging without the robot-to-robot measurements is performed to find the MTM by matching the multiple feature maps of different robots. Generally, the process of the feature map matching has required too much computation time due to its iterative property. This dissertation proposed the SFMM, which is based on the spectral information. The non-iterative process of finding the map transformation matrix made the SFMM performed faster, which benefits the real-time implementation of feature map merging. Therefore, the SFMM contributes to multi-robot SLAM which use feature maps to represent the surrounding environments, especially for vision-based map building systems. The accuracy of the SFMM was verified by measuring the acceptance index according to various map rotation angles sampled randomly in experiments. Moreover, the robustness of the SFMM was verified by consistently finding the best MTM in spite of the existence of multiple candidates. The SFMM can be extended by analyzing the appropriate radius of the circular geometric kernel according to the geometric relation among features for future work.

The conventional spectrum-based map merging method can solve the problem but requires a sufficient amount of common regions among maps, which degenerates the efficiency of multi-robot SLAM. The naive use of visual landmarks to reduce the common regions is likely to cause inaccurate map merging due to the inevitable errors in the recognition of them. This paper proposed the ESMM, which is a novel extension of the conventional map merging technique, when visual landmarks are used to reduce the common regions. The ESMM was applied to merging three individual maps built by different robots in two different outdoor environments. The results showed that the ESMM was successfully performed. Also, the map merging errors of the ESMM were much smaller than the SMM.

The grid map merging without any information on initial robot poses or inter-robot measurements is one of the challenging problems in multi-robot systems. This paper proposed the TMM to solve the problem. The spatial feature of each map was analyzed based on the sinogram extracted from the map, and the map transformation matrix between maps was computed by utilizing the difference and correlation between the sinograms. The TMM was applied to various pairs of partial maps obtained from both indoor and outdoor environments. The results showed that the map merging was successfully performed by the TMM, and its performance was better than the conventional algorithm in the context of accuracy and consistency. Besides, our algorithm was applicable to not only merging two maps but also more maps in successive rounds. In future work, the TMM will be extended to three-dimensional map merging.

Bibliography

[1] H. C. Lee, S. H. Lee, M. H. Choi, and B. H. Lee, “Probabilistic map merging for multi-robot RBPF-SLAM with unknown initial poses,” *Robotica*, vol. 30, no. 2, pp. 205-220, Mar. 2012.

[2] M. Montemerlo and S. Thrun, “Simultaneous localization and mapping with unknown data association using FastSLAM” in *Proc. of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sep. 14-19, 2003, pp. 1985-1991.

[3] A. Howard, “Multi-robot simultaneous localization and mapping using particle filters,” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243-1256, Dec. 2006.

[4] A. I. Mourikis and S. I. Roumeliotis, “Predicting the performance of cooperative simultaneous localization and mapping,” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1273-1286, Dec. 2006.

[5] L. E. Parker, G. Bekey, and J. Barhen, “Current state of the art in distributed autonomous mobile robots,” *Distributed Autonomous Robotic*

Systems, vol. 4, pp. 3-12, 2000.

[6] L. E. Parker, “Distributed intelligence: Overview of the field and its application in multi-robot systems,” *Journal of Physical Agents*, vol. 2, no. 1, pp. 5-14, Mar. 2008.

[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics,* Cambridge: MIT, 2006, Chapter 10.

[8] H. C. Lee and B. H. Lee, “Improved feature map merging using virtual supporting lines for multi-robot systems,” *Advanced Robotics*, vol. 25, no. 13-14, pp. 1675-1696, Aug. 2011.

[9] S. Carpin, “Fast and accurate map merging for multi-robot systems,” *Autonomous Robots*, vol. 25, no. 3, pp. 305-316, Oct. 2008.

[10] H. C. Lee, N. Kwak, J. H. Lee, and B. H. Lee, “Probabilistic feature matching for map merging in the multi-robot FastSLAM with unknown initial correspondence,” in *Proc. of International Conference on Ubiquitous Robots and Ambient Intelligence*, Seoul, Korea, Nov. 20-22, 2008, pp. 693-698.

[11] K. Konolige, D. Fox, B. Limketkai, et al., “Map merging for distributed robot navigation,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems,* Las Vegas, Nevada, USA, Oct. 27-31, 2003, pp. 212-217.

[12] X. S. Zhou and S. I. Roumeliotis, “Multi-robot SLAM with unknown initial correspondence: the robot rendezvous case,” in *Proc. of IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 9-15, 2006, pp. 1785-1792.

[13] A. Birk and S. Carpin, “Merging occupancy grid maps from multiple robots,” *Proceedings of the IEEE*, vol. 94, pp. 1384-1397, Jul. 2006.

[14] W. H. Huang and K. R. Beevers, “Topological map merging,” *The International Journal of Robotics Research*, vol. 24, no. 8, pp. 601-613, Aug. 2005.

[15] A. Howard, “Multi-robot mapping using manifold representations,” in *Proc. of IEEE International Conference on Robotics and Automation*, New Orleans, Louisiana, USA, Apr. 26-May 1, 2004, pp. 4198-4203.

[16] H. J. Chang, C. S. G. Lee, Y. C. Hu, et al., “Multi-robot SLAM with topological/metric maps,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, Oct. 29-Nov. 2, 2007, pp. 1467-1472.

[17] S. Thrun and Y. Liu, “Multi-robot SLAM with sparse extended information filters,” in *Proc. of International Symposium of Robotics Research*, Sienna, Italy, Oct. 19-22, 2003, pp. 254-266.

[18] D. G. Lowe, “Object recognition from local scale-invariant features,” in

Proc. of IEEE International Conference on Computer Vision, Kerkyra, Greece, Sep. 20-27, 1999, vol. 2, pp. 1150-1157.

[19] H. Bay, A. Ess, T. Tuytelaars, et al., “SURF: Speeded up robust features,” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, Jun. 2008.

[20] A. J. Davison, I. D. Reid, N. D. Molton, et al., “MonoSLAM: Real-time single camera SLAM,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052-1067, Jun. 2007.

[21] R. Duda and P. Hart, “Use of the Hough transform to detect lines and curves in the pictures,” *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[22] L. A. Fernandes and M. M. Oliveira, “Real-time line detection through an improved Hough transform voting scheme,” *Pattern Recognition*, vol. 41, no. 1, pp. 299-314, Jan. 2008.

[23] I. K. Kim, N. Kwak, H. C. Lee, and B. H. Lee, “Improved particle filter using geometric relation between particles in FastSLAM,” *Robotica*, vol. 27, no. 6, pp. 853-859, 959, Oct. 2009.

[24] N. Kwak, K. Yokoi, and B. H. Lee, “Analysis of rank-based resampling based on particle diversity in the Rao–Blackwellized particle filter for simultaneous localization and mapping,” *Advanced Robotics*, vol. 24, no.

4, pp. 585-604, Mar. 2010.

[25] H. C. Lee, S. K. Park, J. S. Choi, and B. H. Lee, “PSO-FastSLAM: An improved FastSLAM framework using particle swarm optimization,” in *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, Texas, USA, Oct. 11-14, 2009, pp. 2763-2768.

[26] 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, 205-217, Mar. 2008.

[27] S. N. Sinha, J. M. Frahm, M. Pollefeys, et al., “GPU-based video feature tracking and matching,” in *Proc. of Workshop on Edge Computing using New Commodity Architectures*, Chapel Hill, NC, USA, May, 2006.

[28] F. Tungadi, W. L. D. Lui, L. Kleeman, et al., “Robust online map merging system using laser scan matching and omnidirectional vision,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 18-22, 2010, pp. 7-14.

[29] T. Y. Zhang and C. Y. Suen, “A Fast Parallel Algorithm for Thinning Digital Patterns,” *Communications of the ACM*, vol. 27, no. 3, pp. 236-239, Mar. 1984.

[30] J. Radon, “On the determination of functions from their integral values along certain manifolds,” *IEEE Transactions on Medical Imaging*, vol. 5,

no. 4, pp. 170-176, Dec. 1986.

[31] S. H. Lee, H. C. Lee, and B. H. Lee, “A scan restoration method for robust polar scan matching in dynamic environments,” *Advanced Robotics*, 2013. (online, DOI:10.1080/01691864.2013.791741)

[32] H. C. Lee, T. Yaniss, and B. H. Lee, “Grafting: A path replanning technique for rapidly-exploring random trees in dynamic environments,” vol. 26, no. 18, pp. 2145-2168, Sep. 2012.

[33] S. H. Lee, H. C. Lee, and B. H. Lee, “Visually-extended vector polar histogram applied to robot route navigation,” *International Journal of Control, Automation, and Systems*, vol. 9, no. 4, pp. 726-736, Aug. 2011.

[34] H. C. Lee, S. H. Lee, J. M. Kim, and B. H. Lee, “Robust scan matching with curvature-based matching region selection,” in *Proc. of IEEE/SICE International Symposium on System Integration*, Kyoto, Japan, Dec. 20-22, 2011, pp. 1257-1262.

[35] H. C. Lee, S. H. Lee, S. H. Lee, et al., “Comparison and analysis of scan matching techniques for cooperative-SLAM,” in *Proc. of International Conference on Ubiquitous Robots and Ambient Intelligence*, Incheon, South Korea, Nov. 23-26, 2011, pp. 165-168.

[36] H. C. Lee, S. H. Lee, D. J. Kim, and B. H. Lee, “Visual route navigation using an adaptive extension of rapidly-exploring random trees," in *Proc.*

of IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, Oct. 18-22, 2010, pp. 1396-1401.

[37] H. C. Lee, T. S. Lee, S. H. Lee, et al., “Visual path following and obstacle avoidance using multiple cameras for outdoor environments,” in *Proc. of International Conference on Ubiquitous Robots and Ambient Intelligence*, Gwangju, South Korea, Oct. 29-31, 2009, pp. 709-711.

[38] H. C. Lee, N. Kwak, I. K. Kim, and B. H. Lee, “A map merging technique using the fingerprint matching method in multi-robot FastSLAM,” in *Proc. of International Conference on Ubiquitous Robots and Ambient Intelligence*, Pohang, South Korea, Nov. 2007, pp.452-457.

[39] N. Kwak, I. K. Kim, H. C. Lee, and B. H. Lee, “Adaptive prior boosting technique for the efficient sample size in FastSLAM,” in *Proc. of International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, Oct. 29-Nov. 2, 2007, pp.630-635.

[40] N. Kwak, I. K. Kim, H. C. Lee, and B. H. Lee, “Analysis of resampling process for the particle depletion problem in FastSLAM,” in *Proc. of IEEE International Symposium on Robot and Human Interactive Communication*, Jeju, South Korea, Aug. 26-29, 2007, pp.200-205.

[41] X. S. Zhou and S. I. Roumeliotis, “Robot-to-robot relative rose estimation from range measurements,” *IEEE Transactions on Robotics*,

vol. 24, no. 6, pp. 1379-1393, Dec. 2008.

[42] T. Bailey, J. Nieto, and E. Nebot, “Consistency of the FastSLAM algorithm,” in *Proc. of IEEE International Conference on Robotics and Automation*, Orlando, FL, USA, May 15-19, 2006, pp. 424-429.

[43] L. Montesano, J. Minguez, and L. Montano, “Probabilistic scan matching for motion estimation in unstructured environments,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, Aug. 2-6, 2005, pp. 3499-3504.

[44] A. Burguera, Y. Gonzalez, and G. Oliver, “Probabilistic sonar scan matching for robust localization,” in *Proc. of IEEE International Conference on Robotics and Automation*, Roma, Italy, Apr. 10-14, 2007, pp. 3154-3160.

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

[46] W. Y. Jeong, and K. M. Lee, “CV-SLAM: A new ceiling vision-based SLAM technique,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug. 2-6, 2005, pp. 3195-3200.

[47] H. Zhang, H. C. Lee, and B. H. Lee, “A practical FastSLAM implementation method using an infrared camera for indoor

environments," Journal of Korea Robotics Society, vol. 4, no. 4, pp. 305-311, Nov. 2009.

[48] F. Lu and E. Milios, "Robot pose estimation in unknown environment by matching 2D range scans," *Journal of Intelligent and Robotic Systems*, vol. 18, no. 3, pp. 249-275, Mar. 1994.

[49] J. S. Gutmann and C. Schlegel, "AMOS: Comparison of scan matching approaches for self-localization in indoor environments," in *Proc. of Euromicro Workshop on Advanced Mobile Robots*, Kaiserslautern, Germany, Oct. 9-11, 1996, pp. 61-67.

[50] I. J. Cox, "Blanche: An experiment in guidance and navigation of an autonomous robot vehicle," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 193-203, Apr. 1991.

[51] 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, Feb. 1992.

[52] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. of International Conference on 3-D Digital Imaging and Modeling*, Quebec City, Canada, May 28-June 1, 2001, pp. 145-152.

[53] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. of Robotics: Science and Systems*, Seattle, WA, USA, Jun. 28-Jul. 1, 2009.

[54] A. Censi, L. Iocchi, and G. Grisetti, “Scan matching in the Hough domain,” in *Proc. of IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 18-22, 2005, pp. 2739-2744.

[55] A. Censi and S. Carpin, “HSM3D: Feature-less global 6DOF scan-matching in the Hough/Radon domain,” in *Proc. of IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12-17, 2009, pp. 3899-3906.

[56] 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.

[57] P. Nunez, R. Vazquez-Martin, A. Bandera, et al., “Fast laser scan matching approach based on adaptive curvature estimation for mobile robots,” *Robotica*, vol. 27, no. 3, pp. 469-479, May 2009.

[58] H. S. Lee and K. M. Lee, “Multi-robot SLAM using ceiling vision,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, Oct. 10-15, 2009, pp. 912-917.

[59] A. León, R. Barea, L. M. Bergasa, et al., “SLAM and map merging,” *Journal of Physical Agents*, vol. 3, no. 1, pp. 13-23, Jan. 2009.

[60] K. Wang, S. Jia, Y. Li, et al., “Research on map merging for multi-robotic system based on RTM,” in *Proc. of IEEE International Conference on*

Information and Automation, Shenyang, China, Jun. 2012, pp. 156-161.

[61] 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, Aug. 2008.

[62] M. Cugliari and F. Martinelli, “A FastSLAM algorithm based on the unscented filtering with adaptive selective resampling,” *Field and Service Robotics*, vol. 42, pp. 359-368, Jun. 2008.

[63] M. Cummins and P. Newman, “FAB-MAP: Probabilistic localization and mapping in the space of appearance,” *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647-665, Jun. 2008.

[64] D. Fox, J. Ko, K. Konolige, et al., “Distributed multirobot exploration and mapping,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1325-1339, Jul. 2006.

[65] P. Henry, M. Krainin, E. Herbst, et al., “RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647-663, Apr. 2012.

[66] A. Nüchter, K. Lingemann, and J. Hertzberg, “6D SLAM: 3D mapping outdoor environments,” *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 699-722, Aug. 2007.

[67] D. Borrmann, J. Elseberg, K. Lingemann, et al., "Globally consistent 3D mapping with scan matching," *Robotics and Autonomous Systems*, vol. 56, no. 2, pp. 130-142, Feb. 2008.

[68] J. Ryde and H. Hu, "3D mapping with multi-resolution occupied voxel lists," *Autonomous Robots*, vol. 28, no. 2, pp. 169-185, Feb. 2010.

[69] J. Feng, "Combining minutiae descriptors for fingerprint matching," *Pattern Recognition*, vol. 41, no. 1, pp. 342 – 352, Jan. 2008.

[70] M. E. Jankowska, *A Hough Transform Based Approach to Map Stitching*, Ph. D. Dissertation, University of Oxford, UK, Sep. 2009.

[71] D. F. Wolf and G. S. Sukhatme, "Semantic mapping using mobile robots," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 245-258, Apr. 2008.

[72] A. Nuchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, pp. 915-926, Nov. 2008.

[73] H. C. Lee and B. H. Lee, "Enhanced-spectrum-based map merging for multi-robot systems," *Advanced Robotics*, 2013. (Online published, DOI: 10.1080/01691864.2013.819609)

[74] [Online]. Available: http://cres.usc.edu/radishrepository

[75] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, no. 6, pp.

381–395, 1981.

[76] S. Belongie and J. Malik, “Matching with shape contexts,” in *Proc. of IEEE Workshop on Content-based Access of Image and Video Libraries*, Hilton Head Island, SC, USA, Jun. 2000, pp. 20-26.

[77] S. Belongie and J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 24, Apr. 2002.

초 록

다중로봇 SLAM(Simultaneous Localization And Mapping)은 미지의(unknown) 환경에서 여러 대의 로봇들이 장착된 센서들만 이용하여 스스로 위치를 인식함과 동시에 집합적(collective) 지도를 작성하는 기술이다. 집합적 지도는 각 로봇들에 의해 작성된 개별지도들을 지도 간 변환행렬(Map Transformation Matrix)을 이용하여 병합함으로써 얻게 된다. 하지만, 로봇들이 서로 초기위치를 공유할 수 없다면 로봇들은 각자 고유의 좌표계를 기준으로 개별지도를 작성하기 때문에 지도병합의 난이도는 높아진다. 심지어 주행 중에 로봇 간 상대위치측정 정보를 사용할 수 없다면, 지도병합의 난이도는 더욱 높아지게 된다. 본 논문은 로봇 간 초기위치 정보나 상대위치측정 정보가 불필요한, 개별지도 간 상관관계 분석 기반의 새로운 지도병합 기술들을 제안한다. 개별지도로부터 추출한 고유의 스펙트럼(spectrum)정보 혹은 토모그래피(tomography)정보 간 교차상관관계(cross-correlation) 함수를 생성한다. 교차상관관계 함수의 최대치에 대응하는 각도 및 변위를 기반으로 지도 간 변환행렬을 구성하여 지도병합을 수행한다. 본 논문에서 제안하는 스펙트

럼 정보 간 상관관계 기반 지도병합 기술들은 기존기술에서 확장된 것들이다. 첫 번째 확장 기술은 격자지도병합(grid map merging)에만 적용되는 기존기술을 특징지도병합(feature map merging)에도 적용할 수 있도록 확장한 SFMM(Spectrum-based Feature Map Merging)이다. SFMM은 특징지도의 특징점 위치와 함께, 가상지지선(Virtual Supporting Lines)을 이용하여 특징점들 간 기하학적 관계로부터 스펙트럼 정보를 추출한다. 이렇게 추출된 스펙트럼 정보 간 교차상관관계를 이용하여 특징지도 간 변환행렬을 계산하여 지도병합을 수행함으로써 기존기술의 한계점을 극복한다. 두 번째 확장 기술은 기존기술에서 필요한 지도 간 공통영역(overlapping area)의 크기를 줄일 수 있는 ESMM(Enhanced-Spectrum-based Map Merging)이다. 필요한 공통영역의 크기를 줄이면 다중로봇의 운용의 융통성이 높일 수 있다. ESMM은 환경으로부터 로봇 간 공유 가능한 객체를 인식하고 추정한 후, 그 위치를 기반으로 가우시안 분포를 이용하여 격자지도를 증강(enhancement)시키고, 그로부터 증강 스펙트럼을 추출한다. 이렇게 추출된 스펙트럼 정보 간 교차상관관계를 이용하여 지도 간 변환행렬을 계산함으로써 보다 작은 공통영역으로도 지도병합을 수행할 수 있다. 마지막으로, 본 논문에서는 토모그래피 정보 간 상관관계를 이용하는 새로운 지도병합 기술인 TMM(Tomography-based Map Merging)을 제안한

다. 토모그래피 정보는 스펙트럼 정보보다 개별지도의 회전각도 및 변위량에 대해 더 자세한 정보를 제공할 수 있기 때문에, 기존기술보다 더 정확한 지도병합을 수행할 수 있다. 지도에서 토모그래피 정보를 추출하기 위해 Radon 변환을 변형하여 적용한다. 추출된 토모그래피 정보 간 교차상관관계를 이용하여 지도 간 변환행렬을 계산하여 지도병합을 수행한다. 본 논문에서 제안된 지도병합 기술들의 향상된 성능은 기존기술보다 높은 지도 간 유사성(similarity) 또는 낮은 지도병합 오차를 나타냄으로써 검증되었다.

주요어: 지도병합, 다중 로봇 SLAM, 상관관계, 분광분석법, 토모그래피(Tomography)

학 번: 2008-30882