



Title	Scan Matching and SLAM for Mobile Robot in Indoor Environment
Author(s)	呂, 吉鑫
Citation	北海道大学. 博士(工学) 甲第12455号
Issue Date	2016-09-26
DOI	10.14943/doctoral.k12455
Doc URL	http://hdl.handle.net/2115/67166
Type	theses (doctoral)
File Information	LV_JIXIN.pdf



[Instructions for use](#)

Scan Matching and SLAM for Mobile Robot in Indoor Environment



Jixin LV

Division of Human Mechanical Systems and Design
Graduate School of Engineering

This dissertation is submitted for the degree of
Doctor of Philosophy in Engineering

Hokkaido University

July 2016

The dissertation titled

Scan Matching and SLAM for Mobile Robot in
Indoor Environment

by

Jixin LV

is reviewed and approved by:

Main referee

Prof. Yukinori Kobayashi

Prof. Itsuro Kajiwara

Prof. Atsushi Konno



July 2016

Acknowledgment

At the beginning of this part, I would like to thank all of people that kindly give me help in the past few years. Five years ago, I came to Japan to join Hokkaido University for my graduation study. That was my first trip to go abroad and I could not speak Japanese at that time. I want to sincerely show my deepest gratitude to my supervisor, Prof. Kobayashi Yukinori, who accepted me as his graduate student. In the past 5 years, he has taught me so many things not only about studies and researches but also the way to behave gentle and confident. Furthermore, Prof. Kobayashi supports my daily life and helps me to get more chances for my future. I will treat these invaluable help as the treasure for my whole life. And I would like to show my appreciation to Associate Prof. Takanori Emaru who is very helpful to my daily life and researchers. He has always ready to give help to me and try everything possible to make things working well. I also want to thank Associate Prof. Yohei Hoshino, who is now in Kitami Institute of Technology. He has instructed me many things on researchers and helped me a lot during my first international conference. And I am thankful to all members of our laboratory especially for Dr. Koichi Imaoka, who helps me a lot of things on research equipment and daily issues.

I would specially like to show my sincere appreciation to Prof. Gamini Dissanayake who gave me the chance to visit University of Technology, Sydney (UTS) as a visiting scholar. He gives me a lot of instruction on researches, helps me to get more chances and also supports my daily cost in Sydney. And I am thankful to Associate Prof. Shoudong Huang who has given me knowledges and experiences on the frontier researches. I want to thank Lakshitha Dantanarayana, Ashok Aravinda, Kanzhi Wu, Teng Zhang, Wenjie Lu, Dao Su, Mahdi Hassan, Raphaël Bergamote and other students for their kind help during my stay in UTS.

I am grateful to China Scholarship Council (CSC) and Japan Student Services Organization (JASSO), which provided me the scholarship to support my living expense and studies. And I am thankful to Center for Engineering Education Development (CEED), which supported me to conduct my oversea internship. I want to show my appreciation to the staffs of graduate school of Engineering, especially for

English Engineering Education (E³) Program staffs for their kind assistance and warm support.

I offer my regards to my parents who always support my decision without reservation. They have tried their best to raise me up and give a warm family to me. I would like to thank all of my friends especially to Hongfei Qiao and Lulu Zhang who can always have a talk with me and support me when I was depressed.

Abstract

Autonomous indoor mobile robots are very promising application of robotics. In order to realize autonomous navigation, a robot that enters an unknown environment needs to reconstruct a consistent map of the environment and estimate its pose with respect to the map, simultaneously. This problem is well known as the Simultaneous Localization And Mapping (SLAM) problem, which has attracted a lot of interest from researchers in past few decades. The most popular approaches towards SLAM problem are usually developed based on the probabilistic methods, such as Extended Kalman Filter (EKF) SLAM, particle filter SLAM, and maximum likelihood SLAM. In recent years, a robust technology named “Scan Matching” plays a very important role in solving the SLAM problem. By matching sensor scans that are taken from different poses, the scan matching method can efficiently estimate the rigid transformation of the robot between two poses. Due to the fact that the exploring sensors are usually very accurate and robust, scan matching is very efficient for mobile robot to localize itself with respect to the given reference scans or maps.

Although vision based approaches are getting more and more popular in SLAM research field, vision sensors are sensitive to the unpredictable variations of environment, such as the change of the lighting condition. Besides, most vision based solutions construct sparse feature points based maps which are not sufficient for robot autonomous navigation. Therefore Laser Range Finder (LRF) based scan matching method and fast indoor SLAM framework are still widely desired in consideration of the robustness of LRF towards environment changes. Another widely adopted sensor is Inertial Measurement Unit (IMU) which provides measurements of accelerations and rotating rates at the same time. In consideration of cost efficiency, Micro Electrical Mechanical Systems (MEMS) technology based IMU is preferable in consuming grade applications as well as robotic researches. However, the measurements of low-cost MEMS-IMUs are usually corrupted by various types of noises. Thus, a calibration work to compress noises is necessary before the usage of MEMS-IMU.

The main contributions of this study are consisted of three main parts:

1. ***Fast and Robust Scan Matching Approach***: Various scan matching methods have been introduced. And the most widely used methods, Iterative Closest Point (ICP) and its variants, have been deeply investigated and modified to obtain better performance. New preprocessors as well as new distance metric for association process have been proposed to enhance the robustness of ICP. A new framework of incremental scan matching has been developed to fulfill the scan matching task in large indoor loops.
2. ***Line-segment based EKF-SLAM with Slope and Edge Detection***: In this contribution, planar line-segment based EKF-SLAM has been well investigated at the very beginning. Then, the slope and edge in structured indoor environment have been modelled to line-segments and merged into the EKF-SLAM framework. In order to overcome the effect of regularly appearing outliers of odometer readings, an robust prediction process of EKF-SLAM, which can automatically choose most suitable operating pattern from odometer based prediction and ICP based prediction, has been proposed.
3. ***Straightforward MEMS-IMU Calibration Framework***: A straightforward and efficient calibration method toward MEMS-IMU has been developed. This method utilizes commonly adopted local gravity as reference of accelerometers calibration. The reference of gyroscope calibration is obtained by taking advantage of the proposed robust scan matching solution.

Table of Contents

Chapter 1	Introduction.....	1
1.1	Simultaneous Localization and Mapping	1
1.1.1	Map Categories.....	2
1.1.2	Main Solutions.....	2
1.1.3	Scan Matching	4
1.2	Sensors and Calibrations	4
1.2.1	Laser Range Finder.....	4
1.2.2	Vision Sensor.....	5
1.2.3	RGBD-Sensor.....	5
1.2.4	Inertial Measurement Unit.....	6
1.3	Motivation and Contributions.....	6
1.4	Dissertation Outlines	7
Chapter 2	Scan Matching	9
2.1	Introduction	9
2.2	Scan Matching Manners	11
2.2.1	Pairwise Scan Matching	11
2.2.2	Incremental Scan Matching	13
2.3	Normal Distribution Transformation.....	15
2.4	Polar Scan Matching.....	17
2.5	Iterative Closest Point.....	18
2.5.1	Procedures of ICP	18
2.5.2	Reject Distance Techniques.....	22
2.5.3	Matching Scans with Large Angular Displacement	23
2.6	ICP Variants	24
2.6.1	Iterative Dual Correspondence	24
2.6.2	Metric-based ICP	25
2.6.3	Point to Line ICP	26
2.6.4	Comparisons between ICP variants	27

2.7	Rotation Estimators	28
2.7.1	Polar-coordinate based Orientation Estimation	28
2.7.2	Iterative Closest Normal	30
2.8	Experiments and Discussion.....	34
2.8.1	Long Corridor	34
2.8.2	Cluttered Office Room	39
2.8.3	Statistical Tests and Comparison.....	44
2.9	Conclusion	49
Chapter 3	Efficient Scan Matching in Large Scale Indoor Environment.....	50
3.1	Introduction	50
3.2	Nearest Neighbor Search	51
3.2.1	Linear Search.....	51
3.2.2	Grid Implementation.....	52
3.2.3	K-d Tree.....	53
3.3	Incident Angle Utilization	55
3.3.1	Analysis on Incident Angle and Transformation.....	55
3.3.2	Radial Distance and Incident Angle based Alignment.....	59
3.3.3	Incident Angle Fused Metric based Association	63
3.4	Points Map based Incremental Scan Matching.....	65
3.4.1	Sparse Points Map based ICP	65
3.4.2	Split SPM-ICP	65
3.5	Experiments and Discussions	76
3.5.1	Verification of RIA.....	76
3.5.2	Incremental Scan Matching in Small Environment.....	80
3.5.3	Incremental Scan Matching in Large Loops.....	82
3.5.4	Statistical Tests and Comparisons of Distance Metrics.....	90
3.6	Conclusion	92
Chapter 4	Line-segment based EKF SLAM with Slope Detection	93
4.1	Introduction	93
4.2	System Architecture	94
4.2.1	Robot Platform and Sensors	94
4.2.2	System Architecture	95

4.3	Line-segments Extraction	96
4.3.1	Points Split.....	96
4.3.2	Line-segments Fitting	100
4.3.3	Line-segment Merge	102
4.4	ICP Assisted Line-segments based EKF-SLAM.....	103
4.4.1	Line-segments based EKF-SLAM.....	103
4.4.2	Dead reckoning based EKF-SLAM Prediction	104
4.4.3	EKF-SLAM Features Association.....	106
4.4.4	EKF-SLAM Correction	107
4.4.5	EKF-SLAM New Feature Addition	107
4.4.6	Dead Reckoning Result Evaluation and ICP Compensation.....	108
4.5	Slope and edge feature in EKF-SLAM.....	111
4.5.1	Vertical Feature Extraction and Integration	111
4.5.2	Vertical Feature Association and Correction	118
4.6	Simulation.....	120
4.7	Experiments and Discussion.....	123
4.7.1	Experimental Raw Data.....	125
4.7.2	EKF-SLAM Results	125
4.7.3	Optional Prediction Results	129
4.7.4	Computational Cost	132
4.8	Conclusion	133
Chapter 5	IMU Calibration with Scan Matching.....	134
5.1	Introduction	134
5.2	IMU and Error	135
5.2.1	Consumer Grade IMU	135
5.2.2	Survey on Sensor Errors	136
5.2.3	Sensor Error Model	138
5.3	IMU Calibration Approach.....	140
5.3.1	Static Raw Readings	141
5.3.2	Accelerometer Calibration.....	144
5.3.3	Gyroscope Calibration.....	150
5.4	Experimental Results	154

5.4.1	Calibration Results	154
5.4.2	Application in IMU based Gait tracking	158
5.5	Conclusion	161
Chapter 6	Conclusion and Future work	162
6.1	Conclusion	162
6.2	Future work.....	163
Reference	165

Chapter 1 Introduction

In early years, mobile robots were mostly controlled by human operators instead of autonomous navigation. Some of them were able to take simple tasks without human operation but following with the predefined procedures, such as vacuum cleaning robot. With the development of sensor technology and following efficient algorithms, robots are now becoming more and more intelligent so that they are expected to be able to autonomously fulfill more challenging tasks such as guidance for people, cargo transportation, explores of dangerous areas, etc. The ability to fulfill Simultaneous Localization and Mapping (SLAM) is the basic requirement for robot in these scenarios. In the following subsections, key technologies are reviewed with references.

1.1 Simultaneous Localization and Mapping

In order to perform autonomous navigation, a robot that enters an unknown environment needs to incrementally reconstruct a consistent map of the environment and estimate its pose with respect to the map simultaneously so that it can keep navigating without collision or falling. In the case of localization, the dead reckoning method is widely applied to estimate the displacement of the robot that is equipped with proprioceptive sensors such as wheel encoders and inertial sensors. The noticeable deficiency is that dead reckoning suffers from unbounded error accumulation. In outdoor scenario, this problem can be easily solved by introducing Global Positioning System (GPS) to eliminate the accumulated error. However, in indoor environments, the GPS signal is blocked and unable to be utilized as a global reference. Therefore, external sensors (typically LRF, ultrasonic sensors, cameras, and RGBD-sensors) are essential not only for mapping the environment but also for the localization correction. Inevitably, the data that is obtained by using external sensors to explore surroundings is also corrupted with noise. In order to realize localization and mapping simultaneously and accurately, the data from various sensors must be precisely fused to get the optimal estimation. This problem is well known as the SLAM problem, which has attracted a lot of interest from researchers in the past decade.

A consistent map is needed for localization while an accurate localization is needed for mapping. This chicken-and-egg problem makes SLAM problem not easy to be handled. A brief history of the SLAM research and some typical solutions to the SLAM problem are presented in [1]. The major issues in the SLAM research, such as computational complexity, data association, and environment representation, are discussed in [2]. A general overview and detailed analysis of SLAM can also be found in [3] and [4].

1.1.1 Map Categories

In SLAM research, occupancy grid maps [5] and feature-based maps [6] are two widely adopted methods to represent the environment. Occupancy grid maps can represent arbitrary forms of the environment by dividing map into grids, where each cell of the grid is either occupied or free. However, it requires a huge amount of memory for the divided grids, and it is computationally expensive during the update process. Feature-based maps are popular in various SLAM studies due to their compactness. Feature points, line-segments, and planes are typical features that are employed in the feature maps. Line-segment based maps, which can represent structured environments adequately, are often employed in indoor environment applications due to their small memory requirement and low computational cost.

1.1.2 Main Solutions

EKF-SLAM

The Kalman filter, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing noise (random variations) and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone. More formally, the Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. The filter is named for Rudolf (Rudy) E. Kálmán, one of the primary developers of its theory [7]. The Extended Kalman filter (EKF) is the nonlinear version of the Kalman filter. It linearizes nonlinear models so that the Kalman filter can be applied. EKF based SLAM algorithm, which utilizes the EKF to SLAM using the maximum likelihood data association, might be the most influential

SLAM algorithm. For the first decade of this century, the EKF-SLAM has been the de facto method for SLAM [8][9].

The primary drawback of the EKF-SLAM is the linearization of inherent nonlinearities of both the robot motion and the observation models introduces inconsistent performance [10]. And the Gaussian assumptions for errors are not always valid in real scenarios. Furthermore, the performance of EKF-SLAM is heavily relies on the accuracy of data association process.

Particle filter based SLAM

Particle filter based SLAM is an alternative solution of EKF-SLAM. In particle filter based SLAM, the uncertainty of the robot pose is modeled by a number of different weighted particles (hypothesis) and each particle maintains its own map. Usually the map is processed by using EKF and the filter becomes Rao–Blackwellized particle filter [11][12]. Each particle is then weighted again according to its likelihood with measurements [13][14]. The particles that capture negligible weights are replaced by new particles in the proximity of the particles with higher weights [15].

The particle filter based SLAM techniques have overcome the linearization error of EKF and do not limit to Gaussian assumptions. Since they conduct the data association for each particle, partial erroneous data association will not lead to disastrous consequence. However, a large amount of particles is required to be maintained in order to acquire a certain level of accuracy in the estimation process.

Optimization based SLAM

Both of EKF-SLAM and particle filter based SLAM fulfill their estimation by using the Bayesian filter. In recent years, efficient sparse least squares optimization techniques such as famous Graph-based SLAM [16][17] is becoming the state-of-art solution for SLAM problem. The constraints between pose-to-pose and pose-to-feature are usually obtained by using odometry and scan matching. The overall goal of these solutions is to find the configuration of state variables that minimize the sum of these constraints by taking the uncertainty of each constraint into consideration [18].

The optimization processes are usually sensitive to the outliers of data association. Therefore they rely heavily on the accuracy of the associated correspondences [19]. And also, these methods are usually sensitive to initial estimation on the state variables [20].

1.1.3 Scan Matching

Given two partially overlapped scans that are observed at different poses, the technique named scan matching is often used to match these two scans on the same coordinate system to find out the rigid body transformation between corresponding poses [21]. Since exploring sensors are usually very accurate and fast, scan matching is very efficient for mobile robot to localize it relative to the given reference scans or maps. In recent years, scan matching methods are playing a very important role in solving the SLAM problem for its accurate and robust performance [22].

Based on the association method between two scans, scan matching algorithms can be classified into point-to-point, point-to-feature and feature-to-feature. Features such as line-segments, curves and corners are extracted from scanned points. Because of the compressed number of elements in the scan, feature-to-feature scan matching is the fastest approach and it works satisfactorily in structured environment. However, it is risky to apply feature-to-feature methods to match scans that are obtained from a noisy environment since the feature extraction is prone to give bad results in this scenario, which leads the data association in the scan matching approach to become an extremely difficult task.

1.2 Sensors and Calibrations

There are a lot of sensors that have been applied in mobile robot for exploring and localization. Some of them are factory-calibrated and no requirement of further calibration before application. On the contrary, some sensors need calibration before usage due to the lack of calibration or its sensitivity to the changes of environment.

1.2.1 Laser Range Finder

Planar Laser Range Finder (LRF) is one of the most popular exploring sensors in the SLAM research due to their high speed and accuracy. Another merit of using LRFs is that they are robust to variations of lighting and temperature conditions. A fixed planar LRF takes measurements in one plane and thus only a 2D map can be built. To acquire a 3D point cloud of the environment, some continuous or reciprocating rotation mechanisms to drive planar LRF have been developed by researchers in [23] and [24]. An alternative method is to fix two planar LRFs orthogonally to realize horizontal and vertical scans

[25]. Compared with rotating mechanisms, the latter solution has some merits such as saving energy, reducing noise, and avoiding extra vibration.

Generally speaking, LRF itself does not need any calibration work to correct its measurements. The calibration related to LRF are the usually the requirement to estimate accurate transformation between LRF frame and actuator's frame [26], other sensor's frame [27] or robot frame.

1.2.2 Vision Sensor

Vision sensor might be the most popular sensor that has been adopted in SLAM research communities for it provides rich information of the explored surroundings and makes features extraction and association more reliable. Furthermore, camera is usually low-cost, lightweight, and low electrical power consuming.

Vision based SLAM has achieved a great success in past decades by using different combination and usage of cameras, such as monocular camera [28][29][30], stereo camera [31][32] and Omnidirectional Cameras [33].

Because of the huge application of vision sensor, the camera calibration work has been deeply investigated [34][35] to estimate intrinsic parameters, extrinsic parameters, and distortion coefficients of camera. And corresponding toolboxes can be found either in Matlab or in OpenCV, which simplify the calibration work.

1.2.3 RGBD-Sensor

As a lightweight and low-cost 3D sensor, RGBD Sensor such as Microsoft Kinect which is initially developed for gaming applications, has attracts a lot of attentions from computer vision and robotics research society [36][37]. As the name suggests, RGBD sensor is consisted of a color (RGB) camera and a depth (D) sensor. The color camera provides information on color and texture of the observing area. The depth sensor uses IR projector to emit predefined dotted pattern and captures reflected patterns with an IR camera. Depth is calculated by sensor software on the basis of disparity of reflected patterns with the respect to the reference patterns obtained for a plane placed at a known distance from the sensor. RGBD sensor is powerful in indoor application while its outdoor usage is suppressed due to its short detection range and sensitivity to ambient infrared interference.

For the usage of research activity, various calibration approaches have been proposed [38][39][40] to calibrate RGBD sensor to ensure that the residual systematic errors of the manufacturer's calibration are compressed.

1.2.4 Inertial Measurement Unit

Inertial Measurement Unit (IMU), which is composed of gyroscope sensor and accelerometer sensor, gives measurements on rotational rates and accelerations. It has been extensively used in various applications such as Inertial Navigation Systems (INS) [41], Attitude and Heading Reference System (AHRS) [42], recent popular visual-inertial fusion technology [43][44], and consumer portable devices [45], etc.

High performance IMUs are usually used in military application and commercial airlines. In spite of high accuracy and robustness, their extravagant prices prevent them from their widespread usage in robotics and consumer applications. Micro Electrical Mechanical Systems (MEMS) technology based IMUs are very popular in robotics for its cost-efficient and satisfactory accuracy. However, the measurements of MEMS-IMU are usually corrupted by biases, inaccurate scaling, non-orthogonality of sensor axes, and misalignment between the sensor frame and outer carrying frame. The IMU calibration refers to the process of estimating these quantities.

Factory-based sensor calibration is an expensive and time-consuming process and is typically done for research specific high grade IMUs. For consumer grade inertial sensors, a straightforward and efficiency calibration process which can eliminate governing error is necessary.

1.3 Motivation and Contributions

Mobile robots are very promising application of robotics especially in indoor environment. They can act as service robot for a family, work as Automatic Guided Vehicle (AGV) to convey cargos in industry, take part in Urban Search and Rescue (USAR) to save human beings and properties, etc. Although vision based approaches are getting more and more popular in SLAM research field, vision sensors are sensitive to the unpredictable variations of environment, such as the change of the lighting condition. Besides, most vision based solutions construct sparse feature points based maps which are not sufficient for robot autonomous navigation. Therefore, Laser based scan matching

method and fast indoor SLAM framework are still widely desired in consideration of the robustness of laser towards environment changes. IMU is also widely used in robotics since it can give a good estimation on the attitude of robot. In consideration of reducing cost, MEMS-IMU with easy and efficient calibration work is preferable in consuming grade applications.

The contributions of this study are consisted of three main parts.

- 1) Various scan matching methods have been introduced. And the most widely used methods, Iterative Closest Point (ICP) and its variants, have been investigated and modified to obtain better performance. Furthermore, a scan matching frame that can be utilized in large scale environment has been proposed.
- 2) A line-segment based EKF-SLAM approach is proposed for the application in structured indoor environment. In addition to fulfilling planar SLAM task with line-segment based maps, this method can efficiently detect the slope and edge that exist in structured indoor environment.
- 3) Taking advantage of the proposed robust scan matching solution, a straightforward and efficient calibration method toward MEMS-IMU has been developed.

1.4 Dissertation Outlines

The rest of this dissertation is structured as follows:

- a) **Chapter 2:** Chapter 2 firstly introduces various algorithms of scan matching and then details the ICP variants. Several famous ICP variants have been introduced and compared. The reason why ICP is prone to failure when scans have large angular displacement has been explained. And a new iterative method to eliminate large angular displacement between scans has been proposed.
- b) **Chapter 3:** Chapter 3 firstly demonstrates the methods to speed up the process of ICP variants. And then two fast and robust association approaches for ICP association process have been proposed. The last part of this chapter introduces a framework of an incremental scan matching solution that is able to build sparse point maps and localize robot simultaneously.
- c) **Chapter 4:** Chapter 4 firstly introduces the method that extracts line-segments from LRF scans. Then, the line-segment based EKF-SLAM which adopts optional

prediction process has been detailed. After the line-segment modeling processes of slope and edge have been presented, the vital information of slope and edge has been merged into EKF-SLAM framework.

- d) **Chapter 5:** Chapter 5 presents a straightforward way to calibrate the low-cost MEMS-IMU. A plastic cube that is made by 3D printer is introduced as the carrier of IMU. The proposed approach firstly calibrates accelerometer with multi-position method. By using a reformed swivel chair as an easy turntable, smooth rotation is obtained for gyroscope calibration. The angular displacement is estimated by using proposed scan matching method and the gyroscope is calibrated by comparing raw readings' integrations with angular reference.
- e) **Chapter 6:** Chapter 6 firstly concludes the works that have been explained in the thesis. Based on the summary, several future works that intend to improve current achievements have been proposed.

Chapter 2 Scan Matching

2.1 Introduction

An accurate rigid body transformation estimation of robot between two scans obtained by exploring sensor is one of the key technologies to fulfill the SLAM task successfully. By applying commonly adopted wheel odometer and IMU which consists of accelerometer and gyroscope, dead-reckoning method usually can give good preliminary estimation of the rigid body transformation. However, unpredictable wheel slippage may drag the estimation far away from its correct answer and make the dead reckoning method unreliable. Another possible solution to estimate the rigid body transformation is scan matching technique.

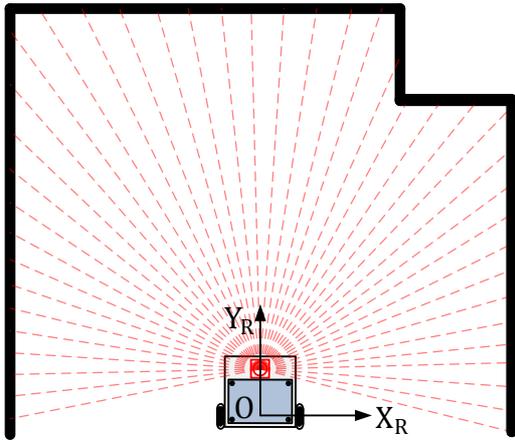
Figure 2-1 (a) and (b) shows a robot with a range sensor observes its surrounding environment at first pose and obtains first scan called reference scan \mathbf{S}_{ref} with respect to the robot frame $O-X_R Y_R$. The scan is consisted of a number of points \mathbf{p}_{ref} :

$$\mathbf{p}_{ref}(i) = \begin{bmatrix} x_{ref}(i) \\ y_{ref}(i) \end{bmatrix} = \begin{bmatrix} r_{ref}(i)\cos\varphi_{ref}(i) \\ r_{ref}(i)\sin\varphi_{ref}(i) \end{bmatrix}, i = 1, 2, \dots, n_{ref}, \quad (2-1)$$

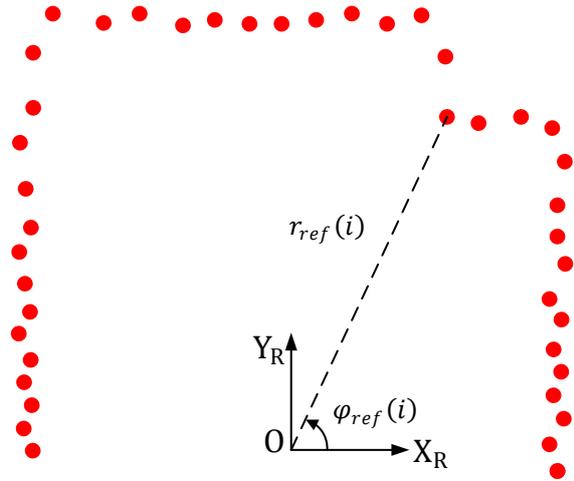
where n_{ref} refers to the points' number of the reference scan, $r_{ref}(i)$ is the radial distance and $\varphi_{ref}(i)$ is the radial angle of i -th point. The robot keeps moving and obtains a new scan \mathbf{S}_{new} with respect to robot frame at second pose, as shown in Figure 2-1 (c) and (d). The relative transformation of robot's second pose with respect to the first pose is often desired in robotics for the purpose of localization and navigation. The relative transformation information is commonly represented by transformation matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{X} \\ 0 & 1 \end{bmatrix}, \quad (2-2)$$

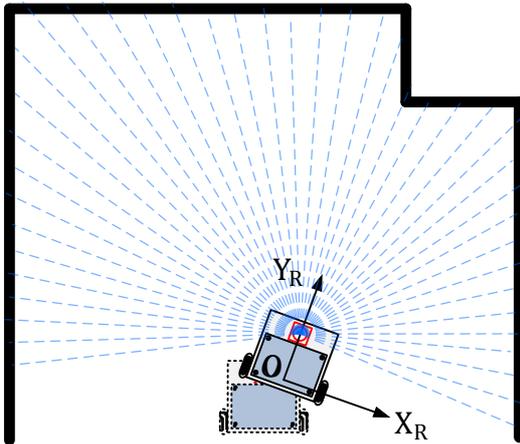
where the translational part $\mathbf{X} = (\Delta x, \Delta y)^T$ and the rotation matrix $\mathbf{R} = \begin{bmatrix} \cos\Delta\theta & -\sin\Delta\theta \\ \sin\Delta\theta & \cos\Delta\theta \end{bmatrix}$. The transformation can also be expressed as $\mathbf{T} \doteq (\Delta x, \Delta y, \Delta\theta)^T$.



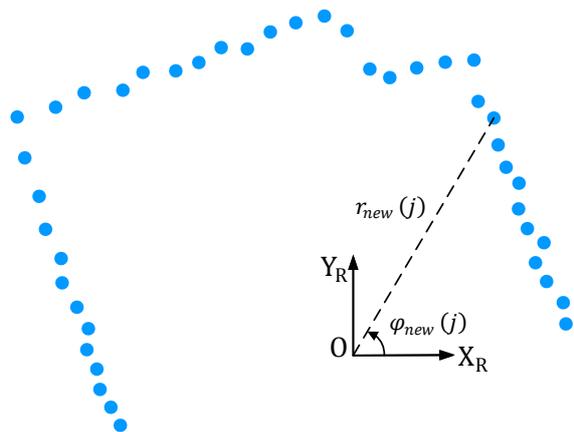
(a) Robot scans at the first position



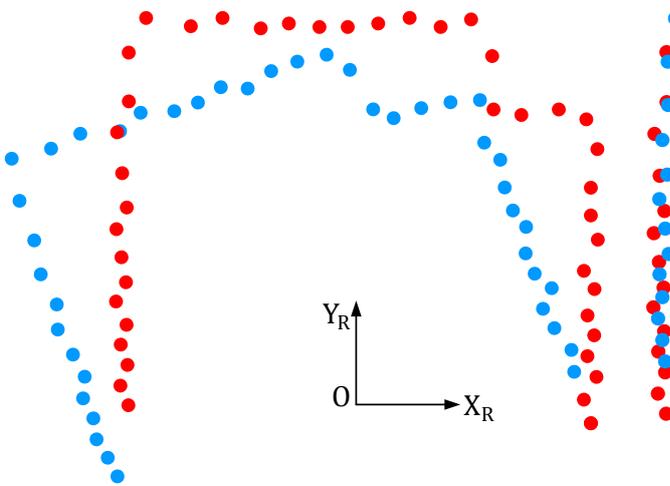
(b) The scans obtained at the first position



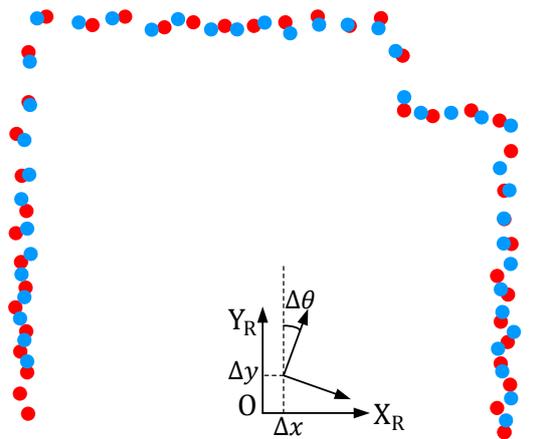
(c) Robot moves and scans at the second position



(d) The scans obtained at the second position



(e) Map scans to the same coordinate



(f) Match scan and estimate transformation

Figure 2-1 Use scan matching to estimate transformation of Robot

To use scan matching method to estimate the relative transformation, all points $p_{new}(j)$ of the scan S_{new} is mapped to the same coordinate frame $O-X_R Y_R$, as shown in Figure 2-1 (e). The transformation that is required to align S_{new} to with S_{ref} in frame $O-X_R Y_R$ is then estimated by using scan matching method.

In this chapter, two scan matching manners to estimate robot trajectory are firstly introduced. And then, some popular scan matching methods are demonstrated. Thirdly, the difficulty of scan matching under large rotation condition is presented. Finally, a new rotation estimator is proposed and its validity is verified by experiments and comparison with traditional rotation estimator.

2.2 Scan Matching Manners

There are at least two ways of matching continuous arriving range scans to estimate robot motion, pairwise scan matching and incremental scan matching.

2.2.1 Pairwise Scan Matching

The most widely used manner, pairwise scan matching, simply matches new arrived scan against the previous scan to find the relative transformation between the poses where the scans have been taken. It is commonly applied as a process of SLAM algorithms to compensate the error of dead-reckoning method since scan matching is usually more accurate than traditional wheel odometer. Pairwise scan matching is straightforward and very efficient for it only deals with two original scans.

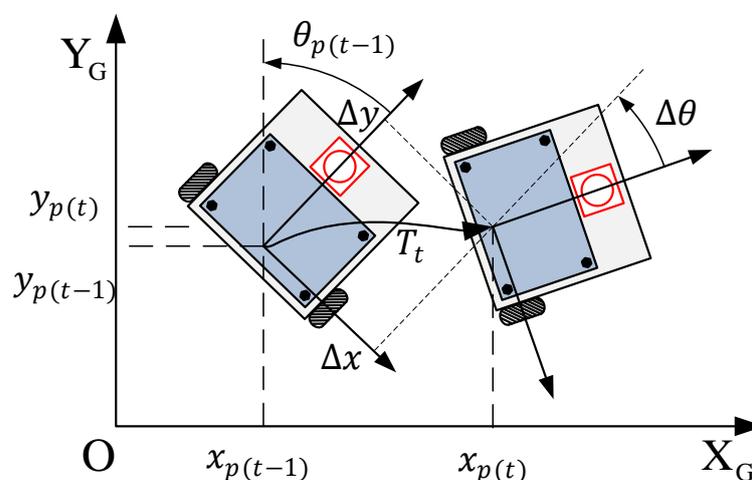


Figure 2-2 Relative transformation and robot poses in global frame

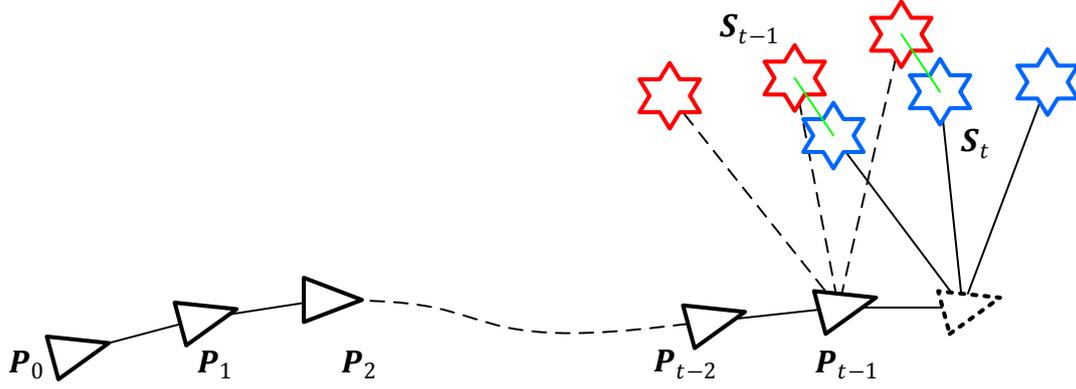


Figure 2-3 Pairwise scan matching

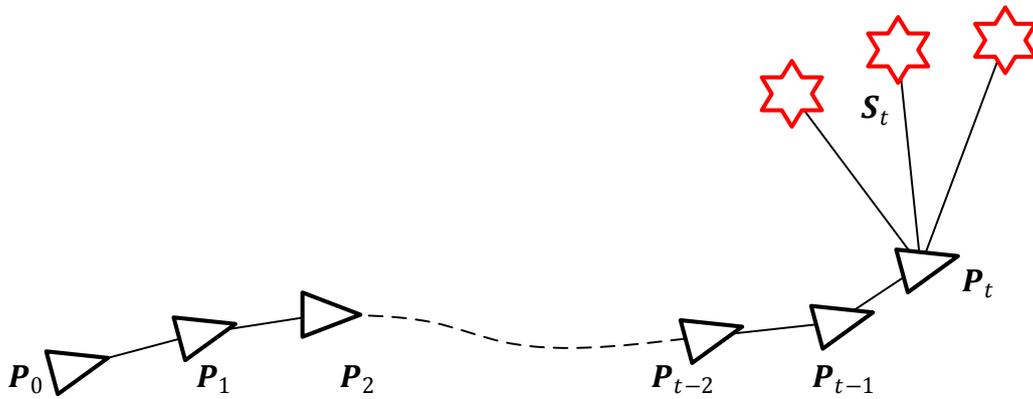


Figure 2-4 Replace old reference scan by using newest scan

Pairwise scan matching can also be used to estimate the trajectory of robot and to map the surroundings. Assume at time $t - 1$, a robot takes a scan \mathbf{S}_{t-1} of its surroundings at known pose $\mathbf{P}_{t-1} = [x_{p(t-1)} \quad y_{p(t-1)} \quad \theta_{p(t-1)}]^T$ with respect to global frame $O-X_G Y_G$, as shown in as shown in Figure 2-2. The robot keeps moving to arrive at an unknown pose \mathbf{P}_t and takes another scan \mathbf{S}_t at time t . The relative transformation of pose \mathbf{P}_t with respect to \mathbf{P}_{t-1} can be estimated by applying scan matching method on two scans, reference scan \mathbf{S}_{t-1} and new scan \mathbf{S}_t , as shown in Figure 2-3. With the relative transformation \mathbf{T}_t estimated by scan matching, the pose \mathbf{P}_t can be obtained by

$$\mathbf{P}_t = \mathbf{P}_{t-1} \oplus \mathbf{T}_t, \quad (2-3)$$

where the operation symbol \oplus makes above equation equal to

$$\mathbf{T}_{P_t} = \mathbf{T}_{P_{t-1}} \mathbf{T}_t, \quad (2-4)$$

$T_{P_{t-1}}$ and T_{P_t} are the transformation matrix of pose P_{t-1} and P_t , respectively. And

$$T_{P_{t-1}} = \begin{bmatrix} R_{P_{t-1}} & X_{P_{t-1}} \\ 0 & 1 \end{bmatrix}, \quad (2-5)$$

where the translational part $X_{P_{t-1}} = (x_{p(t-1)} \ y_{p(t-1)})^T$ and the rotation matrix

$$R_{P_{t-1}} = \begin{bmatrix} \cos\theta_{p(t-1)} & -\sin\theta_{p(t-1)} \\ \sin\theta_{p(t-1)} & \cos\theta_{p(t-1)} \end{bmatrix}.$$

In pairwise scan matching, global map is not going to be maintained. After each scan matching process, the previous reference scan S_{t-1} will be discarded while S_t will be kept as new reference scan for next scan matching, as shown in Figure 2-4. This process can be conducted repeatedly to find the following poses of robot P_{t+1} , P_{t+2} , etc. If all of the scans have been recorded, a global map can be obtained by registering scans on the global coordinate based on corresponding estimated robot poses. This global map, however, includes superfluous data points since successive scans are all partially overlapped.

The error will accumulate when pairwise scan matching is used to estimate the trajectory of robot. Therefore pairwise scan matching is treated as scan matching based odometer. To compress the error accumulation, an intuitive improvement towards pairwise scan matching is to introduce key frame method, or a rather suitable name, the key scan method. The key scan method is very straightforward. It initially sets a scan as the key scan and then runs pairwise scan matching as usual. After a certain distance robot has moved, or a certain angle robot has rotated, or some other certain conditions has been satisfied, the new arrived scan is used to matched against previous key scan to compress the accumulated error and then it is going to be saved as the new key scan. Key scan method can slow down the error accumulation with little extra computational cost.

2.2.2 Incremental Scan Matching

Another fashion of scan matching, which is called incremental scan matching, matches new scans to a maintained global map. After the new scan is matched and registered, the global map is updated by merging with registered scans.

The procedures of incremental scan matching can be summarized as follows:

- 1) **Initialization:** In the incremental scan matching process, the first scan S_0 is going to be treated as the initial global map M_0 . Generally, the frame of S_0 is usually regarded as global frame, therefore the initial pose $P_0 = [0 \ 0 \ 0]^T$.

- 2) **Matching:** At time t , the prior estimation of robot pose \bar{P}_t is given by

$$\bar{P}_t = P_{t-1} \oplus \bar{T}_t, \quad (2-6)$$

where \bar{T}_t is the prior estimation of relative transformation from time $t - 1$ to time t and is usually estimated by dead-reckoning method, as shown in Figure 2-5. The new scans S_t is firstly transformed based on the prior pose \bar{P}_t and then matched against M_{t-1} to obtain the accurate estimation of robot pose P_t .

- 3) **Update:** Based on the scan matching result, the new scan S_t is registered on the global frame. In the registered new scan, the data points that have not been associated in the last step of scan matching are regarded as the new features. And map M_{t-1} is commonly augmented with the new features in registered new scan to obtain the new map M_t , as shown in Figure 2-6.

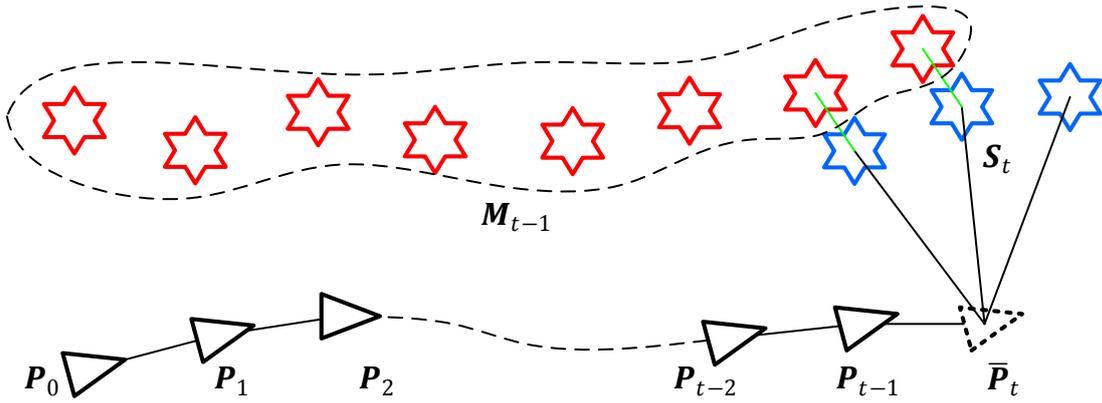


Figure 2-5 Incremental Scan Matching

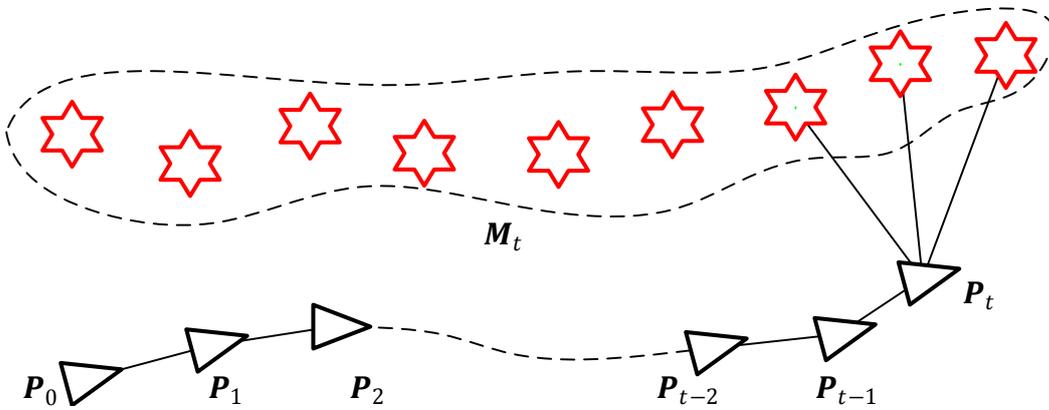


Figure 2-6 Update global map by merging new scan

By matching scans in this fashion, the process of transformation error accumulation is efficiently compressed since every new scan is matched against the maintained global map instead of a single previous scan. Furthermore, comparing with the global map that is obtained by pairwise scan matching, the map maintained by incremental scan matching needs less number of data points to represent the global environment. This is because the overlapping data points in the new scan are treated as re-observed features and are not registered to the global map during the update in incremental scan matching.

One obvious disadvantage of incremental scan matching is that its computational cost is much heavier than pairwise scan matching since it uses augmenting map as the reference scan for scan matching. And the augmenting map captures larger number of data than single scan.

There are several popular scan matching algorithms have been proposed, such as Normal Distribution Transformation (NDT) [46], Polar Scan Matching (PSM) [47] and the most widely applied Iterative Closest Point (ICP) [48] with its variants. These algorithms will be introduced in the following subsections.

2.3 Normal Distribution Transformation

NDT based scan matching is a grid-based technique, which subdivides the plane into cells with constant size. It maps the reference scan \mathbf{S}_{ref} onto a grid space and calculates the Normal Distribution (ND) in each cell, as shown in Figure 2-7.

For k -th cell that captures sufficient points $\mathbf{p}_k(i), i = 1, 2, \dots, n_k$, the mean of the points in the cell is calculated as

$$\bar{\mathbf{p}}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathbf{p}_k(i). \quad (2-7)$$

While the covariance matrix of the points is calculated as

$$\boldsymbol{\Sigma}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} (\mathbf{p}_k(i) - \bar{\mathbf{p}}_k)(\mathbf{p}_k(i) - \bar{\mathbf{p}}_k)^T. \quad (2-8)$$

The points in the k -th cell is then represented by its ND $(\bar{\mathbf{p}}_k, \boldsymbol{\Sigma}_k)$. The new scan \mathbf{S}_{new} is transformed based on the prior estimation of the relative transformation $\bar{\mathbf{T}} = \begin{bmatrix} \bar{\mathbf{R}} & \bar{\mathbf{X}} \\ 0 & 1 \end{bmatrix}$ as

$$\mathbf{S}'_{new} = \bar{\mathbf{T}} \otimes \mathbf{S}_{new} \quad (2-9)$$

where the operation symbol \otimes makes above equation equal to

$$\mathbf{p}'_{new}(j) = \bar{\mathbf{R}}\mathbf{p}_{new}(j) + \bar{\mathbf{X}}, \quad j = 1, 2, \dots, n_{new}. \quad (2-10)$$

The transformed new scan \mathbf{S}'_{new} is then mapped in to the grid space of reference scan. For a point $\mathbf{p}'_{new}(j)$ and its corresponding ND $(\bar{\mathbf{p}}_k, \Sigma_k)$, the evaluation score is defined as the probability when robot execute another scan on the pose where the reference scan obtained. Therefore the total score of scan matching is calculated as follows

$$\mathbf{s} = \sum_{j=1}^{j=n_c} \exp \left[-\frac{1}{2} (\mathbf{R}\mathbf{p}_{new}(j) + \mathbf{X} - \bar{\mathbf{p}}_k)^T \Sigma_k^{-1} (\mathbf{R}\mathbf{p}_{new}(j) + \mathbf{X} - \bar{\mathbf{p}}_k) \right], \quad (2-11)$$

where n_c is the number of corresponding points. The relative transformation $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{X} \\ 0 & 1 \end{bmatrix}$ is iteratively updated by maximizing total score \mathbf{s} using Newton's algorithm.

The NDT does not require point-to-point associations and can match scans accurately. However, NDT algorithm may not work properly when the prior estimation of the relative transformation between two scans is not close to the ground truth. Furthermore, the scan matching performance of NDT method relies on the size of the grid and outliers detection.

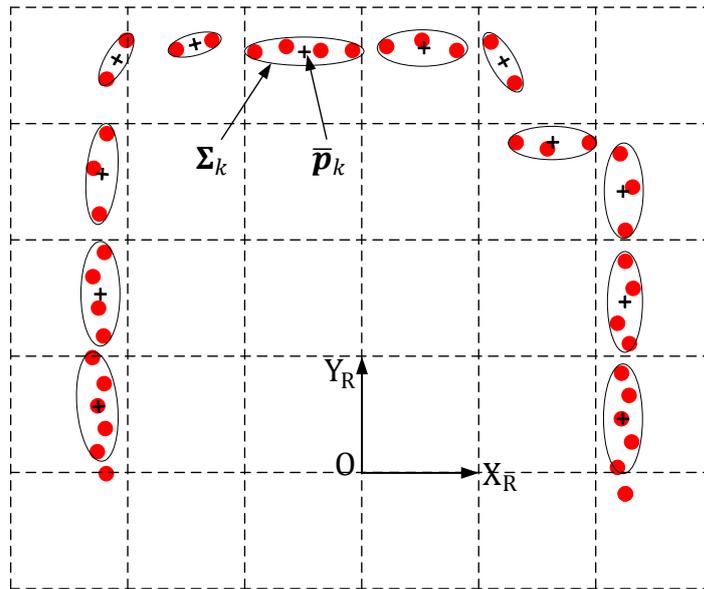


Figure 2-7 Normal Distributions created by reference scans

2.4 Polar Scan Matching

Unlike NDT and ICP that match scans in Cartesian coordinate frame, PSM handles scan matching processing in polar coordinate system, which takes advantage of the native polar coordinate system of a laser scan.

The procedure of PSM can be summarized as follows:

- a) **Preprocess:** By using median filter, PSM removes the points that are suspected to be scanned from small dynamic objects such as human foot, legs of chairs and tables. Afterwards, the points that capture distances longer than preset threshold are tagged and will not be utilized in scan matching process. In consideration of the accuracy of following interpolation process, the measurements that are suspected to be scanned from different objects are segmented.
- b) **Projection and Interpolation:** Before matching process, the new scan \mathbf{S}_{new} is transformed to \mathbf{S}'_{new} based on the prior estimation on the relative transformation $\bar{\mathbf{T}}$. To make the radial angles in agreement with reference scan \mathbf{S}_{ref} , the transformed new scan \mathbf{S}'_{new} is then resampled to \mathbf{S}''_{new} . The corresponding radial distance of \mathbf{S}''_{new} is calculated by applying linear interpolation on \mathbf{S}'_{new} . The resampled points that have distance longer than preset threshold or have been occluded by other measurements are tagged and will not be used in scan matching.
- c) **Scan Matching:** The points in \mathbf{S}_{ref} and \mathbf{S}''_{new} that capture same radial angle are associated as a correspondence. The translation is firstly estimated by minimizing the sum of the difference of correspondences' radial distances. The rotation estimation is conducted by shift the scan \mathbf{S}''_{new} along the radial angular axis to find the minimal sum error of radial distances with \mathbf{S}_{ref} . The translation estimation and rotation estimation is executed interactively and iteratively. The estimation process will be terminated when iterations reach the maximal limitation or the change of estimation is small.

In the original work of PSM, it has claimed that PSM outperforms ICP in both accuracy and converging speed. However, the first two procedures of PSM are not very straightforward, which prevent PSM from wide application. Furthermore, the performance of PSM heavily relies on the quality of initial guess on the transformation between two scans.

2.5 Iterative Closest Point

Among various existing scan matching algorithms, ICP is the most widely used one for the virtue of its superior simplicity and high efficiency. Without any prefix or declaration, ICP refers to the point-to-point based iterative closest point algorithm. It is a direct-decent technique denominated method that adopts point-to-point Euclidean distance as its distance metric. It was originally proposed in [48] for 3D shape registration. In general, ICP algorithm establishes correspondences between two given scans based on the closest-point rule, and it calculates a transformation that minimizes the distances between corresponding points. This process will be iteratively conducted until some predefined termination conditions has been satisfied.

2.5.1 Procedures of ICP

Given a reference scan \mathcal{S}_{ref} and a new scan \mathcal{S}_{new} that are supposed to be partially overlapped, the procedure of ICP to estimate the relative transformation \mathbf{T} between two scans is introduced in the following parts.

A. Initialization

Based on wheel odometer or constant velocity assumption, dead reckoning method can gives ICP prior estimation of the relative transformation $\mathbf{T}^0 = \begin{bmatrix} \mathbf{R}^0 & \mathbf{X}^0 \\ 0 & 1 \end{bmatrix} = \bar{\mathbf{T}} = \begin{bmatrix} \bar{\mathbf{R}} & \bar{\mathbf{X}} \\ 0 & 1 \end{bmatrix}$. Then the points of scan \mathcal{S}_{new} are firstly transformed by \mathbf{T}^0 as follows

$$\mathcal{S}_{new}^0 = \mathbf{T}^0 \otimes \mathcal{S}_{new}, \quad (2-12)$$

where the operation symbol \otimes makes above equation equal to

$$\mathbf{p}_{new}^0(j) = \mathbf{R}^0 \mathbf{p}_{new}(j) + \mathbf{X}^0, \quad j = 1, 2, \dots, n_{new}. \quad (2-13)$$

The superscript of scan \mathcal{S}_{new}^0 and \mathbf{T}^0 indicate the times of ICP iteration that have been conducted.

The initialization work is finished after \mathcal{S}_{new}^0 has been mapped into the same coordinate frame with \mathcal{S}_{ref} . Then, the 1st iteration of ICP will be started.

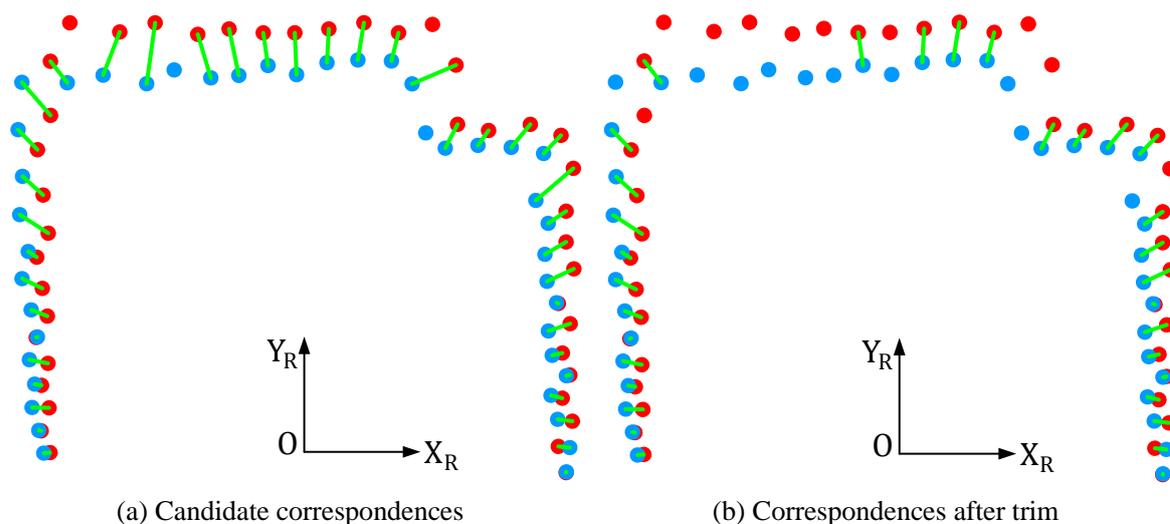


Figure 2-8 Association based on the closest point rule

B. Association

At the beginning of $(k+1)^{\text{th}}$ iteration, based on the closest point rule, the points of \mathcal{S}_{ref} are associated with the points of \mathcal{S}_{new}^k that has been transformed from \mathcal{S}_{new}^0 by previous k iterations of ICP. The associated points pairs are regarded as candidate correspondences and then be trimmed with distance based filter which eliminates the pairs that capture longer distance than the reject distance threshold D_{icp}^{k+1} , as shown in Figure 2-8. The superscript of D_{icp}^{k+1} indicates that this reject distance threshold should be updated after each iteration. The methods about how to determine an appropriate value of reject distance threshold will be introduced later. With $m-1$ valid pairs have been stored, the new valid pair $(\mathbf{p}_{ref}(i), \mathbf{p}_{new}^k(j))$ is added to the collection as m^{th} correspondence $\mathcal{C}(m) = (\mathbf{p}_{ref}(c_{ref}^m), \mathbf{p}_{new}^k(c_{new}^m))$, where the sequences $c_{ref}^m = i$ and $c_{new}^m = j$.

C. Minimization

Given a set of correspondences $\mathcal{C}(m) = (\mathbf{p}_{ref}(c_{ref}^m), \mathbf{p}_{new}^k(c_{new}^m))$, $m = 1, 2, \dots, n_c$, the summed errors between correspondences are defined as follows:

$$E_{icp}^{k+1} = \sum_{m=1}^{n_c} \|\mathbf{p}_{ref}(c_{ref}^m) - \mathbf{R}_{icp}^{k+1} \mathbf{p}_{new}^k(c_{new}^m) - \mathbf{X}_{icp}^{k+1}\|^2 \quad (2-14)$$

where \mathbf{R}_{icp}^{k+1} and \mathbf{X}_{icp}^{k+1} are the rotation matrix and translation of transformation \mathbf{T}_{icp}^{k+1} that need to be estimated at $(k+1)^{th}$ iteration of ICP to minimize the sum of distance errors of correspondences between the scans \mathbf{S}_{ref} and \mathbf{S}_{new}^k .

The transformation is calculated by minimizing the sum of errors as follows

$$\mathbf{T}_{icp}^{k+1} = \arg \min_{\mathbf{R}_{icp}^{k+1}, \mathbf{X}_{icp}^{k+1}} \sum_{m=1}^{n_c} \|\mathbf{p}_{ref}(c_{ref}^m) - \mathbf{R}_{icp}^{k+1} \mathbf{p}_{new}^k(c_{new}^m) - \mathbf{X}_{icp}^{k+1}\|^2. \quad (2-15)$$

This optimization problem has been solved analytically in [49] by setting

$$\frac{\partial E_{icp}^{k+1}}{\partial \Delta \theta_{icp}^{k+1}} = 0, \quad \frac{\partial E_{icp}^{k+1}}{\partial \Delta x_{icp}^{k+1}} = 0, \quad \frac{\partial E_{icp}^{k+1}}{\partial \Delta y_{icp}^{k+1}} = 0, \quad (2-16)$$

which gives solution as follows

$$\Delta \theta_{icp}^{k+1} = \arctan \left(\frac{A_{x_{ref}} A_{y_{new}} + n_c A_{y_{ref}} x_{new} - n_c A_{x_{ref}} y_{new} - A_{x_{new}} A_{y_{ref}}}{n_c A_{x_{ref}} x_{new} + n_c A_{y_{ref}} y_{new} - A_{x_{ref}} A_{x_{new}} - A_{y_{ref}} A_{y_{new}}} \right), \quad (2-17)$$

$$\Delta x_{icp}^{k+1} = \frac{A_{x_{ref}} - \cos(\Delta \theta_{icp}^{k+1}) A_{x_{new}} + \sin(\Delta \theta_{icp}^{k+1}) A_{y_{new}}}{n_c}, \quad (2-18)$$

$$\Delta y_{icp}^{k+1} = \frac{A_{y_{ref}} - \sin(\Delta \theta_{icp}^{k+1}) A_{x_{new}} - \cos(\Delta \theta_{icp}^{k+1}) A_{y_{new}}}{n_c}, \quad (2-19)$$

where A stands for the accumulated terms:

$$A_{x_{ref}} = \sum_{m=1}^{n_c} x_{ref}(c_{ref}^m), \quad A_{y_{ref}} = \sum_{m=1}^{n_c} y_{ref}(c_{ref}^m),$$

$$A_{x_{new}} = \sum_{m=1}^{n_c} x_{new}^k(c_{new}^m), \quad A_{y_{new}} = \sum_{m=1}^{n_c} y_{new}^k(c_{new}^m),$$

$$A_{x_{ref} x_{new}} = \sum_{m=1}^{n_c} x_{ref}(c_{ref}^m) x_{new}^k(c_{new}^m), \quad A_{y_{ref} y_{new}} = \sum_{m=1}^{n_c} y_{ref}(c_{ref}^m) y_{new}^k(c_{new}^m),$$

$$A_{x_{ref} y_{new}} = \sum_{m=1}^{n_c} x_{ref}(c_{ref}^m) y_{new}^k(c_{new}^m), \quad A_{y_{ref} x_{new}} = \sum_{m=1}^{n_c} y_{ref}(c_{ref}^m) x_{new}^k(c_{new}^m).$$

D. Update

With the calculated transformation \mathbf{T}_{icp}^{k+1} , the \mathbf{S}_{new}^k is transformed to \mathbf{S}_{new}^{k+1}

$$\mathbf{S}_{new}^{k+1} = \mathbf{T}_{icp}^{k+1} \otimes \mathbf{S}_{new}^k, \quad (2-20)$$

and the estimation of relative transformation \mathbf{T} between \mathbf{S}_{ref} and \mathbf{S}_{new} is updated from \mathbf{T}^k to \mathbf{T}^{k+1} in the form of

$$\mathbf{T}^{k+1} = \mathbf{T}_{icp}^{k+1} \mathbf{T}^k, \quad (2-21)$$

which also can be described in follows form:

$$\mathbf{X}^{k+1} = \mathbf{R}_{icp}^{k+1} \mathbf{X}^k + \mathbf{X}_{icp}^{k+1}, \quad \mathbf{R}^{k+1} = \mathbf{R}_{icp}^{k+1} \mathbf{R}^k. \quad (2-22)$$

The update of reject distance D_{icp}^{k+1} is the critical task during ICP iteration and it will be detailed in the next sub-section.

E. Termination

Typically, iteration of ICP will be terminated if one of the following conditions is satisfied:

- a) Iteration has researched maximal times limitation that was predefined.
- b) The updating translation and rotation are small than predefined threshold, which indicates the convergence.

ICP is straightforward and very efficient. However, ICP only guarantees the estimation of the relative transformation \mathbf{T} converges to a local minimum that is close to the prior estimation $\bar{\mathbf{T}}$ which is not necessarily the global optimal one. And notice that the most expensive computation of ICP is the association process which finds the closest points in two scans in every iteration.

2.5.2 Reject Distance Techniques

In order to get the satisfactory estimation, reject threshold D_{icp} is the most important parameter which excludes the outliers from candidate correspondences. A small value of D_{icp} may lead to inadequate correspondences so that the estimation is prone to fall into the local optimum. On the contrary, a large value introduces incorrect correspondences so that the accuracy of the estimation will be deteriorated and the converging speed will be slowed.

Several methods to determine and update the reject threshold are listed as follows:

1. Constant: The most straightforward way to define reject threshold is to manually set a certain fixed value. In every iteration, all candidate pairs with a distance longer than this value are rejected to be a valid correspondence. The concept of this method is easy but it is difficult to pick an appropriate value since the transformation error between two scans is decreasing with the iterations going.
2. Median: In this method, D_{icp} equals to the sum of mean and standard deviation of the distances between all candidate pairs [50]. This method is also straightforward and it does not require manually setting parameter. However it assumes that the distribution of distances is Gaussian, which is far from truth in some the real case.
3. Trim: This approach firstly sorts all candidate pairs based on their distances. And then it excludes the candidate pairs whose distance is among a certain percentage of all pairs' distances. This method is less dependent on the shape of the distribution [51]. However, it needs to sort all pairs at every iteration, which increases computational cost.

In this study, the reject distance D_{icp} is initialized as

$$D_{icp}^{max} = 3\sigma + D_s, \quad (2-23)$$

where σ corresponds to range standard deviation of the scanned point and D_s stands for possible maximal translation, which is determined by taking robot translational speed and time interval into consideration. The rejection threshold is updated by the algorithm named relative motion threshold (RMT), proposed in [52]. At $(k+1)^{th}$ iteration, the reject distance is set as

$$D_{icp}^{k+1} = \begin{cases} D_{icp}^{max}, & \text{if } k = 0 \\ D_{icp}^{max} - |\mathbf{X}_{icp}^k|, & \text{if } k = 1 \\ \frac{|\mathbf{X}_{icp}^k|}{|\mathbf{X}_{icp}^{k-1}|} D_{icp}^k, & \text{if } |\mathbf{X}_{icp}^k| < |\mathbf{X}_{icp}^{k-1}| \text{ and } k \geq 2 \\ D_{icp}^k, & \text{if } |\mathbf{X}_{icp}^k| \geq |\mathbf{X}_{icp}^{k-1}| \text{ and } k \geq 2 \\ D_{icp}^{min}, & \text{if } D_{icp}^{k+1} < D_{icp}^{min} \end{cases} \quad (2-24)$$

where

$$D_{icp}^{min} = 3\sigma. \quad (2-25)$$

This method does not need much computational cost while it shows satisfactory performance in outlier rejection process.

2.5.3 Matching Scans with Large Angular Displacement

Mobile robot is widely used in indoor applications. With the consideration of security, in most cases the movement speeds of the mobile robots are limited, which means the translation of the robot should keep small and it can be well estimated by using ICP algorithm. However, the rotational changes can be relatively larger, for instance, when the robot rotates around itself or when some unexpected slippage happens, which leads to the consequence that the angular displacement between two scans becomes excessive large, as shown in Figure 2-9. Unfortunately, ICP are prone to fall into the local optimum when the rotational change is large. This reason can be explained by using Figure 2-9 (b), with large angular displacement, huge Euclidean distance will be introduced to the scanned data point that is far from LRF with its correct corresponding point. In this situation, the association procedure of ICP, which uses closest point rule, is unable to find the correct correspondence. Therefore large rotation between two scans can leads ICP to failure in searching global optimum.

To solve this difficulty, two famous ICP variants, Iterative Dual Correspondence (IDC) and Metric-based ICP (MbICP) have been proposed in [53] and [54], respectively. And they will be introduced in next section.

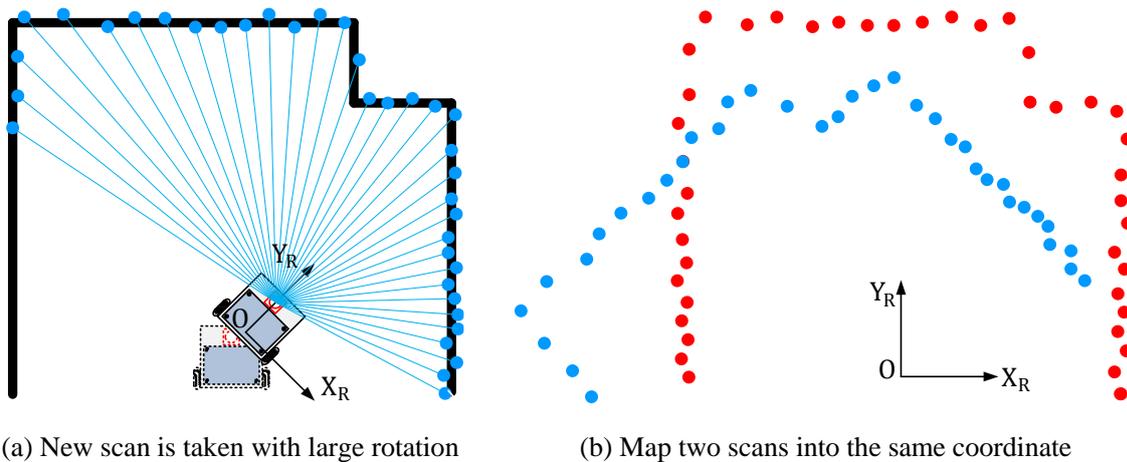


Figure 2-9 Scans with large angular displacement brought by robot rotation

2.6 ICP Variants

2.6.1 Iterative Dual Correspondence

Differing from ICP that adopts closet point rule for association, Iterative Dual Correspondence (IDC) algorithm adopts two different correspondence establishing rules to estimate the transformation between two scans.

The first rule is the closest-point rule which is similar to the correspondence establishing rule of ICP.

The second rule is named matching-range point rule which associates the points that have closest radial distance within certain angular difference.

In every iteration, IDC executes association process two times and establishes two correspondences collections based on different association rules. Then it calculates two transformation results by using least-squares method on two correspondences collections. The final transformation is obtained by combining the translation component of closest-point solution and the rotation component of the matching-range solution.

It has been reported in [53] that IDC converges significantly faster than original algorithm in consideration of iteration times. Especially in the case of large angular displacement existence, IDC shows superior robustness.

2.6.2 Metric-based ICP

Metric-based ICP (MbICP) adopts geometric distance that takes translation and rotation into account at the same time and it is robust to the rotational change.

For two points $\mathbf{p}(i) = \begin{bmatrix} x(i) \\ y(i) \end{bmatrix}$ and $\mathbf{p}(j) = \begin{bmatrix} x(j) \\ y(j) \end{bmatrix}$, the most widely applied Euclidean distance between them is given as

$$d_{Euc}(i, j) = \sqrt{\Delta x(i, j)^2 + \Delta y(i, j)^2}, \quad (2-26)$$

where $\Delta x(i, j) = x(j) - x(i)$ and $\Delta y(i, j) = y(j) - y(i)$.

There are countless combinations that can transform point $\mathbf{p}(i)$ to point $\mathbf{p}(j)$ by rotation and translation. In MbICP, the norm of a transformation $\mathbf{T}_{(i, j)} \doteq (\Delta x, \Delta y, \Delta \theta)^T$ between these two points is defined as:

$$\|\mathbf{T}_{(i, j)}\| = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (L\Delta \theta)^2}, \quad (2-27)$$

where L is an artificial parameter to project rotational angle from radian to length. Based on this definition, a unique distance of two points are proposed as follows

$$d_{MbICP}(i, j) = \min\|\mathbf{T}_{(i, j)}\|. \quad (2-28)$$

With some deduction, this new distance is finally approximated as

$$d_{MbICP}(i, j) = \sqrt{\Delta x(i, j)^2 + \Delta y(i, j)^2 - \frac{(\Delta x(i, j)y(i) - \Delta y(i, j)x(i))^2}{x(i)^2 + y(i)^2 + L^2}}. \quad (2-29)$$

Take advantage of this unique distance for association and followed with least square minimization, MbICP turns out to be much more robust than ICP to solve scan matching problem with the existence of large angular displacement. This improvement is foreseeable since the part of Euclidean distance introduced by angular displacement now could be easily compensated by cost of $L\Delta \theta$ which should be much smaller than the cost be compensated by $\sqrt{(\Delta x)^2 + (\Delta y)^2}$ when L is reasonable. Comparing with IDC, MbICP only needs single association process in each iteration, which makes it about 2 times faster.

2.6.3 Point to Line ICP

Point-to-line Iterative Closest Point (PLICP) executes same procedures as what point-to-point ICP does. The difference between them is that PLICP tries to utilize the information of objects' surface, more specifically, the normal vectors which is estimated by fitting line-segment with neighbor points, as shown in Figure 2-10 (a). In terms of a single correspondence, ICP tries to minimize the straight Euclidean distance between two points while PLICP tries to minimize the distance component along the normal vector of reference point, as shown in Figure 2-10 (b).

Therefore, in $(k+1)^{\text{th}}$ iteration of PLICP algorithm, given correspondences collection $\mathcal{C}(m) = (\mathbf{p}_{ref}(c_{ref}^m), \mathbf{p}_{new}^k(c_{new}^m))$, $m = 1, 2, \dots, n_c$, the errors are defined as follows:

$$E_{plicp}^{k+1} = \sum_{m=1}^{n_c} \|\mathbf{v}_{ref}(c_{ref}^m)(\mathbf{p}_{ref}(c_{ref}^m) - \mathbf{R}_{plicp}^{k+1}\mathbf{p}_{new}^k(c_{new}^m) - \mathbf{x}_{plicp}^{k+1})\|^2 \quad (2-30)$$

where $\mathbf{v}_{ref}(c_{ref}^m)$ is the normal vector of point $\mathbf{p}_{ref}(c_{ref}^m)$. And $\mathbf{v}_{ref}(c_{ref}^m) =$

$$\begin{bmatrix} x_{ref}^{normal}(c_{ref}^m) \\ y_{ref}^{normal}(c_{ref}^m) \end{bmatrix} = \begin{bmatrix} \cos\vartheta_{ref}(c_{ref}^m) \\ \sin\vartheta_{ref}(c_{ref}^m) \end{bmatrix} \text{ where } \vartheta_{ref}(c_{ref}^m) \text{ is the polar angle of the normal}$$

vector.

Therefore, the transformation $\mathbf{T}_{plicp}^{k+1} = \begin{bmatrix} \mathbf{R}_{plicp}^{k+1} & \mathbf{x}_{plicp}^{k+1} \\ 0 & 1 \end{bmatrix}$ is estimated by minimizing the sum of error vectors' norm as follows

$$\mathbf{T}_{plicp}^{k+1} = \arg \min_{\mathbf{R}_{plicp}^{k+1}, \mathbf{x}_{plicp}^{k+1}} E_{plicp}^{k+1}. \quad (2-31)$$

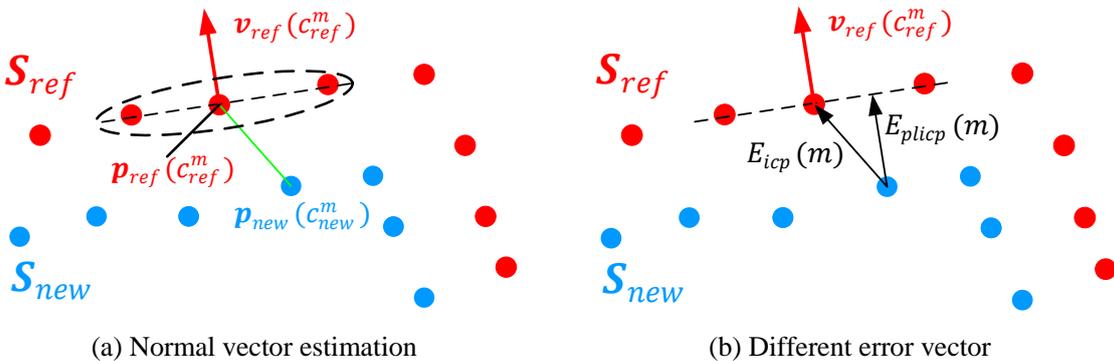


Figure 2-10 Normal estimation and difference between ICP and PLICP

Usually, the point-to-line distance based PLICP is more in line with the actual situation in comparison with point-to-point distance based ICP. In common indoors, most surface area of objects capture low curvature. And the points scanned by sensors are the discrete sampling of the objects' surface, which means the constraint of the scanned point should be along the normal vector. This is the reason why PLICP converges faster than ICP and it is much more robust and accurate.

On the contrary, if the points is all scanned from pillar-like object such as the thin leg of furniture, in which case the single point can roughly represent the whole object or feature, then point-to-point distance based ICP is a better choice. For detail deduction and comprehensive introduction of PLICP, please refer to [55].

2.6.4 Comparisons between ICP variants

A comparison among ICP, IDC and MbICP are conducted in [54], which suggests that both IDC and MbICP are more robust than ICP when large angular displacement between scans is introduced. Both MbICP and IDC need less iteration times to converge than ICP. However, IDC needs two association processes in every iteration, which makes computational cost of IDC larger than two others.

Another comparison between ICP, MbICP and PLICP is explicated in [56]. It shows that MbICP is a good option if large angular displacement occurs. On the contrary, it does not always converge for small rotational errors while the standard ICP doses. And point-to-point based least squares minimization is not good and fast enough comparing with a point-to-line minimization.

The comparison presented in [55] shows that PLICP performs more accurate and converges much faster than other ICP variants.

In the existence of large angular displacement, PLICP performs better than ICP [57]. Nevertheless, it is still difficult to claim that PLICP is robust to large angular displacement since PLICP usually adopts similar association method as ICP.

Taking these comparisons into consideration, using a preprocessor that can efficiently diminish rotational angle between two scans and then applying PLICP to find the residual transformation would be a preferable solution for scan matching.

2.7 Rotation Estimators

This section firstly demonstrates one polar-coordinate based method that is commonly utilized to estimate large rotational angle between two scans. The strengths and weaknesses of this method are analyzed. Then, an iterative method that can efficiently estimate rotational angle of two scans is proposed, which is more accurate than first method especially when the translation between two scans becomes large.

2.7.1 Polar-coordinate based Orientation Estimation

For a point \mathbf{p}_{ref} in reference scan \mathcal{S}_{ref} scanned by robot at first pose, its exact corresponding point in new scan \mathcal{S}_{new} at second pose is marked as \mathbf{p}_{new} . The unknown relative transformation matrix $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{X} \\ 0 & 1 \end{bmatrix}$ should satisfy following equation:

$$\mathbf{p}_{ref} = \mathbf{R}\mathbf{p}_{new} + \mathbf{X}. \quad (2-32)$$

Thus the radial distance of point \mathbf{p}_{ref} can be represent as

$$\|\mathbf{p}_{ref}\| = \|\mathbf{R}\mathbf{p}_{new} + \mathbf{X}\|. \quad (2-33)$$

The translation term \mathbf{X} can be neglected if it is very small comparing with the radial distance of point, then this equation could be simplified to

$$\|\mathbf{p}_{ref}\| \approx \|\mathbf{R}\mathbf{p}_{new}\| = \|\mathbf{p}_{new}\|. \quad (2-34)$$

Therefore the simplest method to estimate rotational angle between two scans is to shift the radial distances along angular axis in polar coordinate to find the angular displacement that makes the radial distances reach maximal cross-correlation [58].

The scans \mathcal{S}_{ref} and \mathcal{S}_{new} plotted in Figure 2-9 have been expressed in polar coordinate from of frame $O-X_R Y_R$, as shown in Figure 2-11 (a). The cross-correlation of range between scans with respect to shift in angular axis is calculated as

$$s(k) = \sum_{i=1}^{n_{ref}} r_{ref}(i) r_{new}(i+k), k = -n_s, -(n_s - 1), \dots, -1, 0, 1, \dots, n_s \quad (2-35)$$

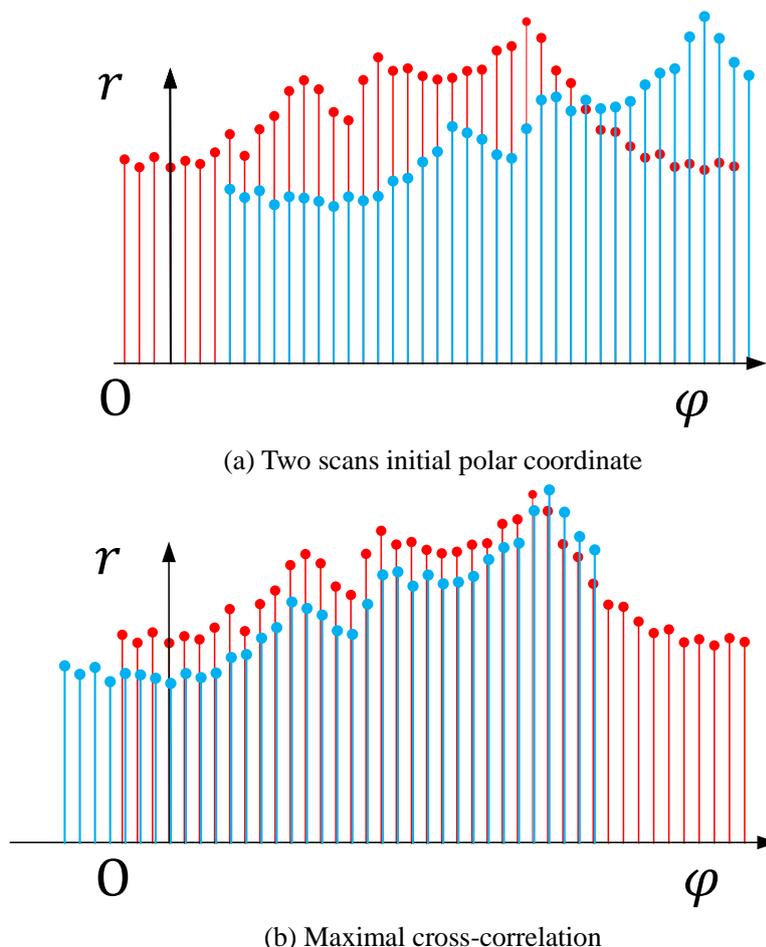


Figure 2-11 Polar coordinate of the scans with large angular displacement

where r_{ref} and r_{new} is the range parameter of the points in \mathcal{S}_{ref} and \mathcal{S}_{new} , n_s is the artificially defined shift interval, and the range $r_{new}(i+k)$ is set to equal zero if $(i+k) < 1$.

Then the angular displacement is roughly estimated as follows

$$\Delta\theta \approx \Delta\varphi = k_m \varphi_{res} \quad (2-36)$$

where φ_{res} is the angular resolution of scans and k_m is the shift sequence that maximizes $s(k)$, as shown in Figure 2-11(b).

Based on this concept, a practical solution named Polar-coordinate based Orientation Estimation (POE) has been introduced in PSM [47]. It firstly resamples the scans and then shifts the scans along angular axis in polar coordinate with a preset interval to find the minimal sum of radial distance differences within a predefined search boundary.

The method is straightforward and easy to apply with the assumption of small translation. The calculation cost is about $O(n_s n_{ref})$ which is mainly paid in the

calculation of Equation (2-35). A small value of n_s makes the process faster but the process will fail to find a good estimation result when the rotational angle is larger than $n_s\varphi_{res}$. On the contrary, a large value of n_s can guarantee process to find the solution around the global minimum but it takes high computation cost. Furthermore, the accuracy of the method is affected by the translation since the Equation (2-34) is valid only when translation is close to zero.

2.7.2 Iterative Closest Normal

To reduce the effect brought by translation during the rotational angle estimation, a new angular displacement estimator called Iterative Closest Normal (ICN) is proposed in this study.

For the correct corresponding points \mathbf{p}_{ref} and \mathbf{p}_{new} , their normal vectors $\mathbf{v}_{ref} = \begin{bmatrix} x_{ref}^{normal} \\ y_{ref}^{normal} \end{bmatrix} = \begin{bmatrix} \cos\vartheta_{ref} \\ \sin\vartheta_{ref} \end{bmatrix}$ and $\mathbf{v}_{new} = \begin{bmatrix} x_{new}^{normal} \\ y_{new}^{normal} \end{bmatrix} = \begin{bmatrix} \cos\vartheta_{new} \\ \sin\vartheta_{new} \end{bmatrix}$ are estimated by fitting tangent lines with their neighboring points. The unknown relative transformation matrix $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{X} \\ 0 & 1 \end{bmatrix}$ should satisfy following equation:

$$\mathbf{v}_{ref} = \mathbf{R}\mathbf{v}_{new}. \quad (2-37)$$

which can be expressed in simplified form:

$$\vartheta_{ref} = \vartheta_{new} + \Delta\theta. \quad (2-38)$$

This means, the rotational angle can be accurately estimated by comparing the normal vectors' polar angles of corresponding points. Therefore, the rotational angle estimation problem becomes an association problem that is required to associate points which are scanned from the same linear surface.

The proposed method ICN iteratively associates the points in two scans that have similar radial distance and similar normal vector with the smallest compensating angular distance. Then the rotational angle is estimated by comparing the normal vectors of associated points. This procedure is iteratively executed until the termination condition is satisfied.

A. Initialization

Similar to the first step of ICP, scan \mathbf{S}_{ref} and \mathbf{S}_{new} are mapped to the same coordinate frame. \mathbf{S}_{new} is transformed to \mathbf{S}_{new}^0 based on the initial estimation of transformation \mathbf{T}^0 . With this initialization work finished, the 1st iteration of ICN starts.

B. Association

At the beginning of $(k+1)^{th}$ iteration, the points of \mathbf{S}_{ref} are associated with the points of \mathbf{S}_{new}^k that has been rotated from \mathbf{S}_{new}^0 by previous iterations of ICN.

For a point $\mathbf{p}_{ref}(i)=[r_{ref}(i) \ \varphi_{ref}(i)]^T$ in scan \mathbf{S}_{ref} , the points that have similar radial distance in scan \mathbf{S}_{new}^k are firstly picked out as possible correspondences with following judgment:

$$|r_{new}^k(j) - r_{ref}(i)| < C_{range}, j = 1, 2, \dots, n_{new} \quad (2-39)$$

where C_{range} is a constant displacement criterion which is decided by taking the robot potential translation and error of scan data into consideration.

Secondly, for each possible corresponding pair that consists of $\mathbf{p}_{ref}(i)$ and $\mathbf{p}_{new}^k(j)$, the angular parameter ϑ of their normal vector $\mathbf{n}_{ref}(i)$ and $\mathbf{n}_{new}^k(j)$ will be compared after compensating angular distance $\Delta\varphi_{ij} = (\varphi_{ref}(i) - \varphi_{new}^k(j))$ as follows

$$|\vartheta_{ref}(i) - \vartheta_{new}^k(j) - \Delta\varphi_{ij}| < C_{normal} \quad (2-40)$$

where C_{normal} is the constant angular criterion that is designed to compensate the estimation error of normal vector. From possible corresponding points, all of the points that satisfies judgment (2-40) will be picked out and treated as the candidate corresponding points of $\mathbf{p}_{ref}(i)$. From the candidate corresponding points, the point $\mathbf{p}_{new}^k(j)$ that has smallest angular distance is found out. If $\Delta\varphi_{ij}$ is smaller than the angular threshold $C_{\Delta\varphi}^{k+1}$, $\mathbf{p}_{new}^k(j)$ will be associated with $\mathbf{p}_{ref}(i)$. The brief process of correspondence establishment has been plotted in Figure 2-12.

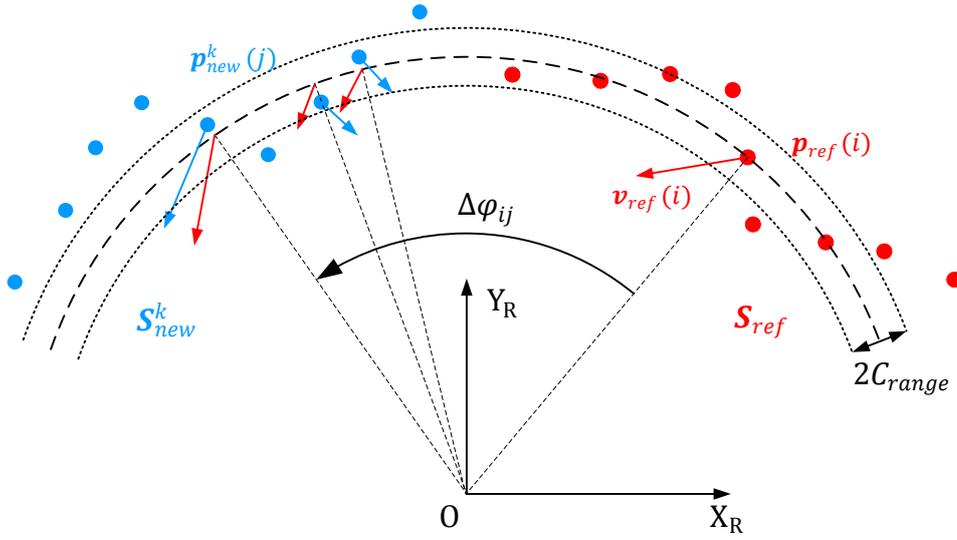


Figure 2-12 Association process

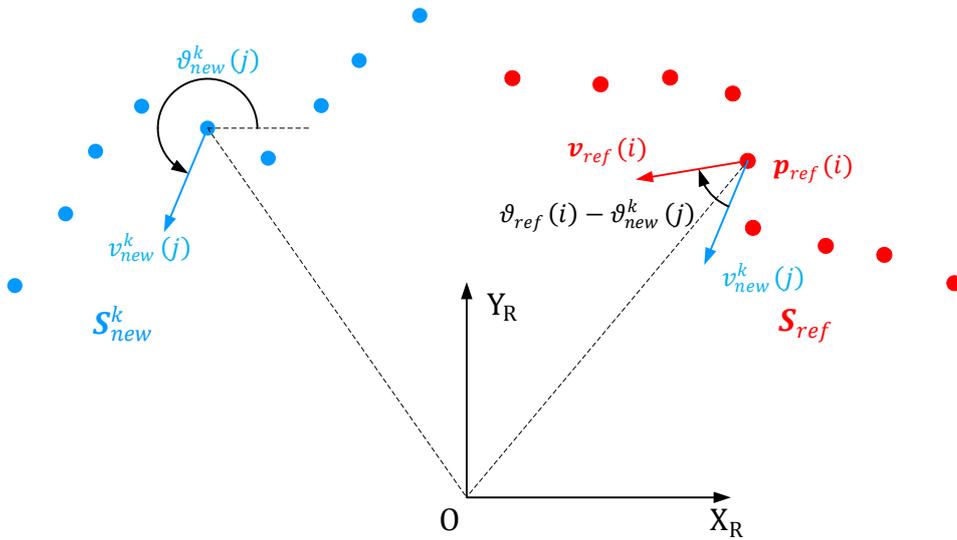


Figure 2-13 Rotational angle estimation from correspondences

C. Rotation estimation

After the association process, the rotational angle between two scans can be easily estimated. As shown in Figure 2-13, from each valid correspondence, a rotational angle estimation $\Delta\theta$ can be calculated by comparing the angles of normal vectors as

$$\Delta\theta = \Delta\vartheta + \sigma_\vartheta = \vartheta_{ref} - \vartheta_{new} + \sigma_\vartheta, \quad (2-41)$$

where σ_ϑ stands for the error of normal vectors.

In order to improve the accuracy of rotational angle estimation, the effect of normal vector's error needs to be compressed. The normal vector of the point that captures long radial distance is usually more accurate because of the long distances between neighbors along the tangent line of the point. Another factor that should be taken into consideration is that the intensity of the scan data varies inversely to the radial distance, which means fewer points will be scanned from the far object than the near one. Since the long range scan data gives better estimation of the rotation, the density factor should be compensated. Finally, for a set of correspondences $\mathbf{C}(m) = (\mathbf{p}_{ref}(c_{ref}^m), \mathbf{p}_{new}^k(c_{new}^m))$, $m = 1, 2, \dots, n_c$, the rotation estimation adopts square of radial distance as its weighting and it is going to be calculated as

$$\Delta\theta_{icn}^{k+1} = \frac{\sum_{m=0}^{n_c} r_{ref}(c_{ref}^m)^2 \Delta\vartheta(m)}{\sum_{m=0}^{n_c} r_{ref}(c_{ref}^m)^2} \quad (2-42)$$

where n_c is the valid correspondence number and $\Delta\vartheta(m)$ is the angular difference of the normal vectors of m^{th} correspondence.

D. Update

The new scan \mathbf{S}_{new}^k is rotated by $\Delta\theta_{icn}^{k+1}$ to obtain \mathbf{S}_{new}^{k+1} . The displacement criterion C_{range} keeps constant since translation has not been compensated. C_{normal} also keeps constant for the estimation error of normal vector will not change. In the $(k+1)^{th}$ iteration, the angular threshold $C_{\Delta\varphi}$ is determined by using angular RMT as follows:

$$C_{\Delta\varphi}^{k+1} = \begin{cases} C_{\Delta\varphi}^{max}, & \text{if } k = 0 \\ C_{\Delta\varphi}^{max} - |\Delta\theta_{icn}^k|, & \text{if } k = 1 \\ \frac{|\Delta\theta_{icn}^k|}{|\Delta\theta_{icn}^{k-1}|} C_{\Delta\varphi}^k, & \text{if } |\Delta\theta_{icn}^k| < |\Delta\theta_{icn}^{k-1}| \text{ and } k \geq 2 \\ C_{\Delta\varphi}^k, & \text{if } |\Delta\theta_{icn}^k| \geq |\Delta\theta_{icn}^{k-1}| \text{ and } k \geq 2 \\ C_{\Delta\varphi}^{min}, & \text{if } C_{\Delta\varphi}^{k+1} < C_{\Delta\varphi}^{min} \end{cases} \quad (2-43)$$

where $C_{\Delta\varphi}^{min}$ and $C_{\Delta\varphi}^{max}$ are the predefined minimal and maximal threshold, respectively.

E. Terminate condition.

The iteration will be terminated when the newest estimated rotation $\Delta\theta_{icn}^{k+1}$ is smaller than a preset threshold or the iteration has researched predefined maximal times.

2.8 Experiments and Discussion

Two experiments have been conducted in different types of environments to execute the comparison between ICP variants and verify the validity of proposed ICN algorithm.

2.8.1 Long Corridor

The robot employed in this experiment is developed by assembling two LRFs on the platform (Plat-F1, Japan Systems Design Co. Ltd), as shown in Figure 2-14 (a). A LRF (UHG-08LX) is attached on the front part of the bottom base to fulfill horizontal scanning. The second layer is placed above the base to support the PC. An aluminum frame is fixed on the forefront of the second layer to vertically sustain vertical LRF (UBG-04LX-F01) whose scanning plane is orthogonal to the forward direction of robot. This design makes vertical LRF be able to obtain spatial scans.

The experiment is executed in a real environment where there is a long narrow corridor, as shown in Figure 2-14(b). The corridor consists of two parts. The left part is a clean passageway with no objects while the right part is filled with unstructured objects such as chairs, tables, glass cabinet, etc. During the experiment, the robot is controlled to move forward with sharp turns, thus the rotation between two successive scans is prone to be large. The robot moves not only on the clean passageway but also moves through the noisy part.

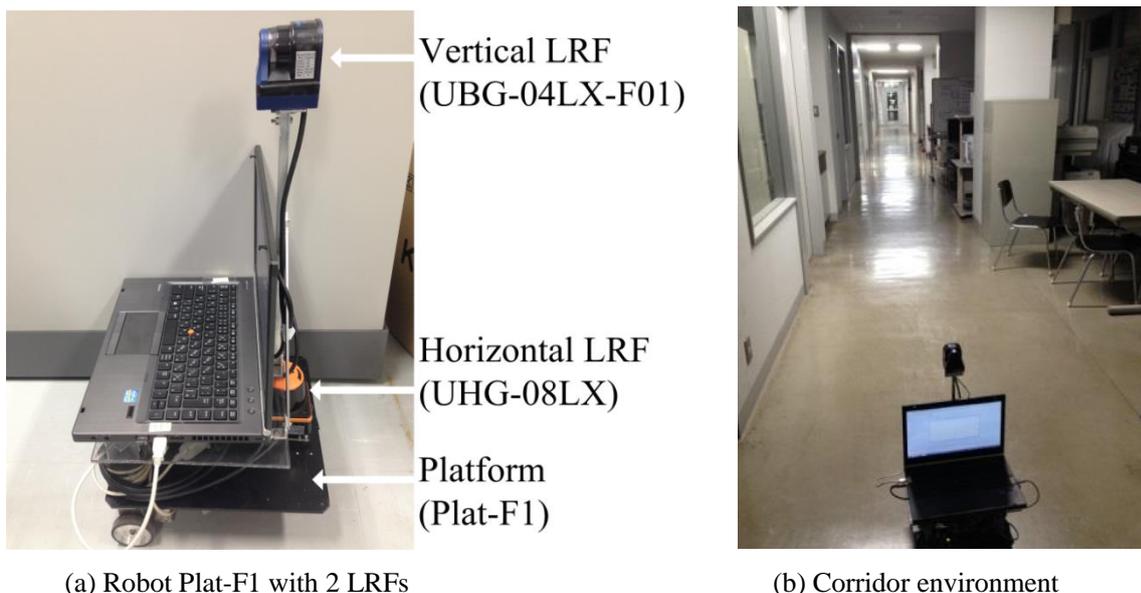
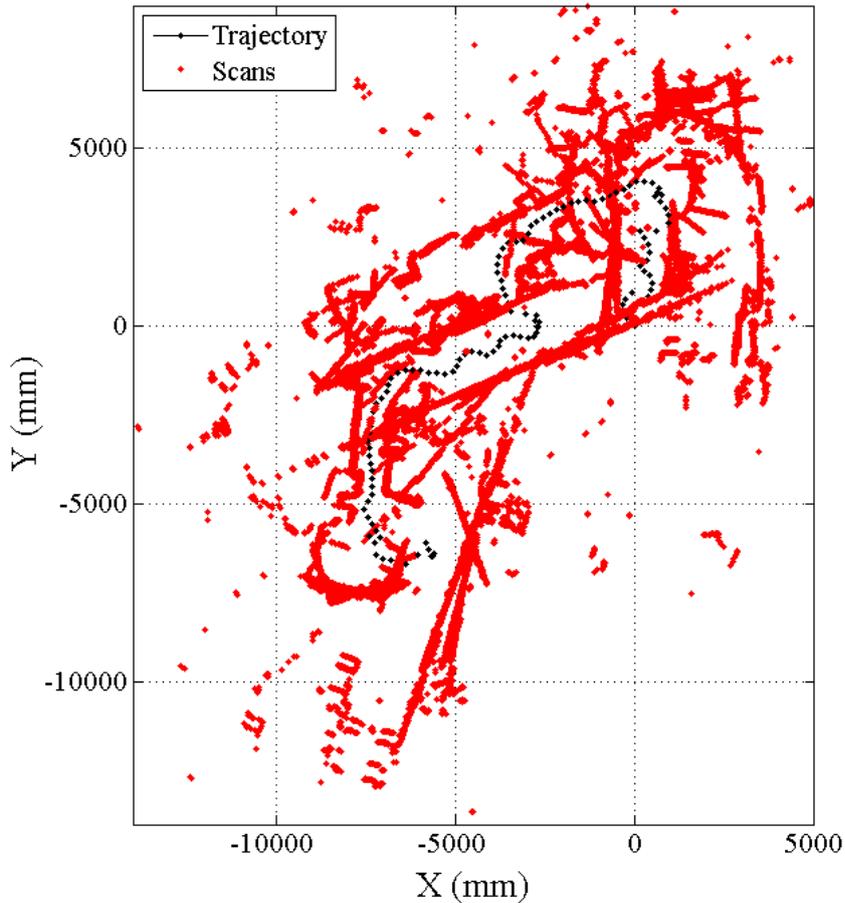
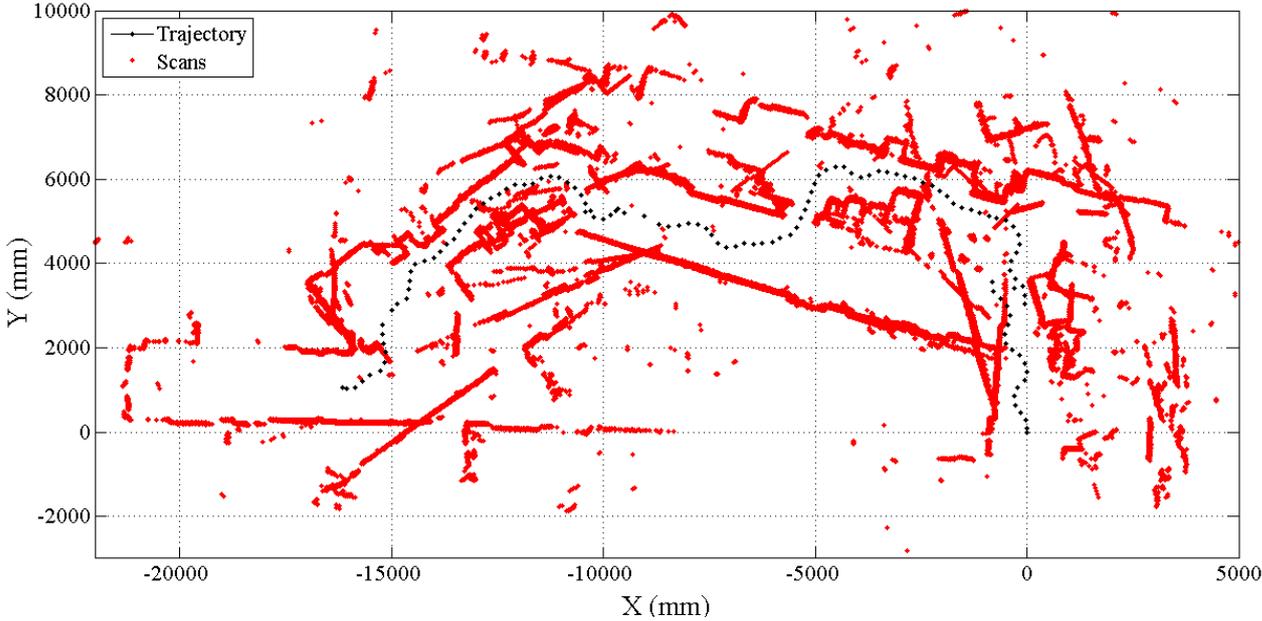


Figure 2-14 The mobile robot and sensors used for the experiment in corridor



(a) ICP matching result



(b) PLICP matching result

Figure 2-15 Successive scans iteratively matched by ICP and PLICP

The first trial in this experiment is to estimate robot trajectory by using ICP and PLICP method in the manner of pairwise scan matching, respectively. In the application of ICP and PLICP, all the scans are registered on the global map based on their corresponding pose in the estimated trajectories, respectively. The matching results of ICP, even with a large reject distance setting, are still poor. The trajectory estimated by ICP is severely twisted and the registered scans cannot outline the corridor, as shown in Figure 2-15 (a). Although PLICP shows a better matching result, several mismatching prevent it to find an accurate trajectory, which leads to the misalignment of scans on the map, as shown in Figure 2-15 (b). These results clearly demonstrate the potential risk brought by large rotational angle when scans are matched by using ICP and PLICP.

The iteration times that have been executed during the scan matching processes of ICP and PLICP are plotted in Figure 2-16. Because of massive outlier correspondences established by using closest point rule, ICP has to run a lot of iterations to converge to a local minimum. Though PLICP failed to provide robust scan matching results, it converges much faster than ICP with less iterations have been conducted. Since the robot is controlled to move forward with turning left and right, the angular displacement between two scans is varying, which leads to the consequence that the iteration times of ICP and PLICP keeps periodically changing.

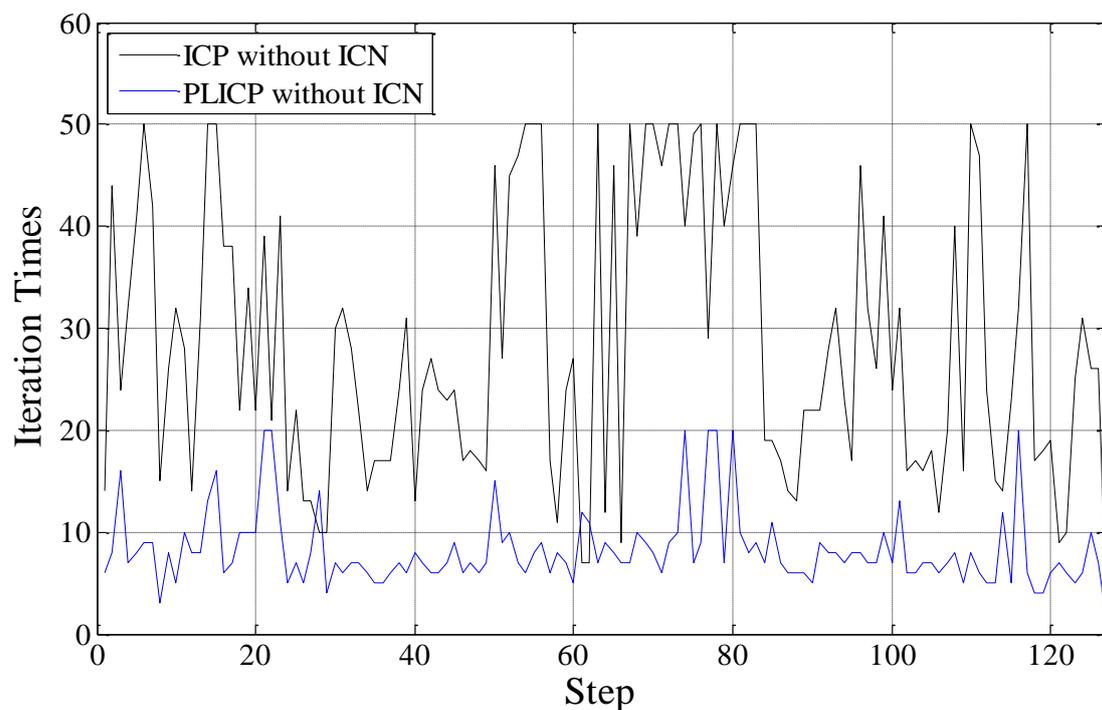
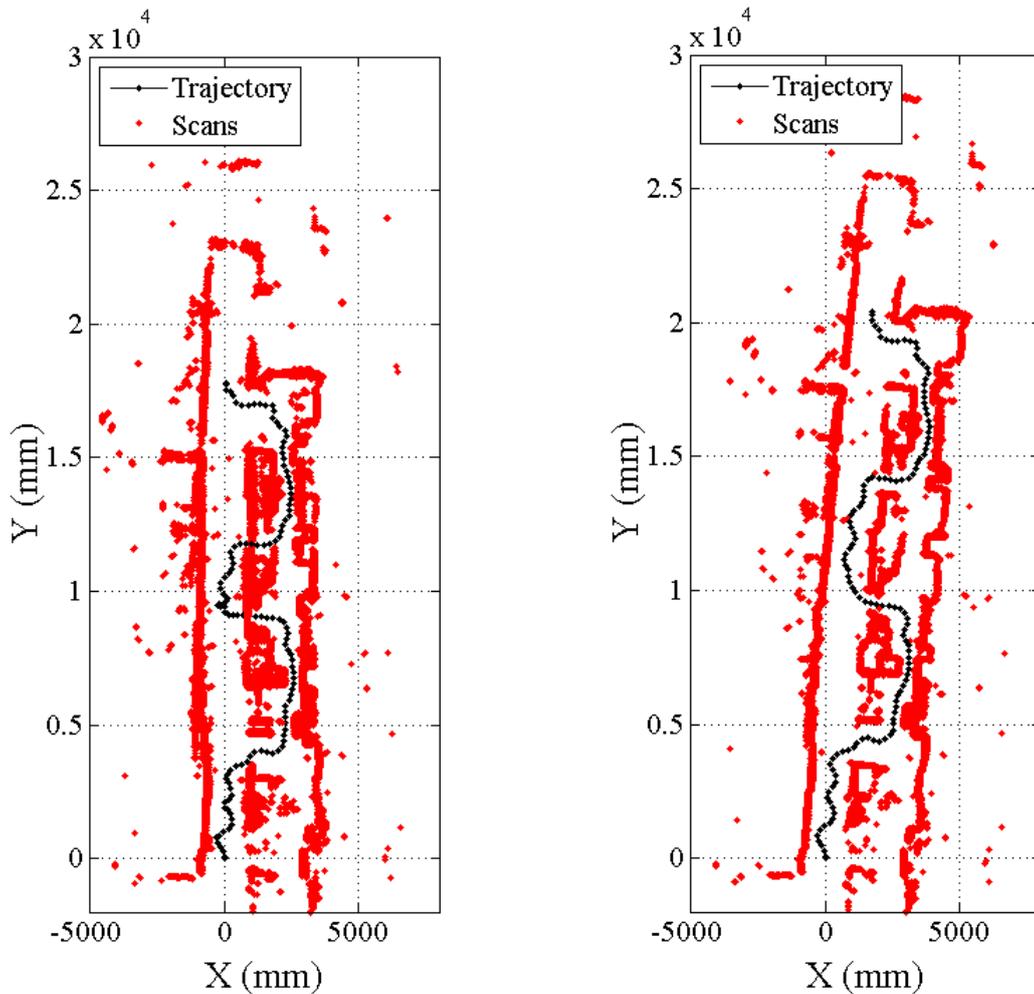


Figure 2-16 The iteration times that have been executed in the scan matching processes



(a) ICP matching result after applying ICN

(b) PLICP matching result after applying ICN

Figure 2-17 Scans matched by ICP and PLICP after applying ICN

In the second trial, the proposed algorithm, ICN is firstly applied to the successive scans to diminish the large angular displacement. The full transformation is then estimated by utilizing ICP and PLICP, respectively. In comparison with the previous results, the new scan matching results are much improved, as shown in Figure 2-17. More or less, PLICP gives out a better result than ICP for there is no collision between robot trajectory and aligned scans. Figure 2-18 plots out the rotation corrected by ICN, and its followers, ICP and PLICP, respectively. There are few differences that can be found, and ICN corrects most of the rotation. The errors of ICN are estimated by comparing ICN corrected angles with PLICP results. The numbers of executed iteration have been plotted in Figure 2-19. Overall, comparing with the iteration time plotted in Figure 2-16, both ICP and PLICP require fewer iteration times after ICN has been introduced. ICN commonly converges in a few iterations and corrects most of the angular displacements between scans.

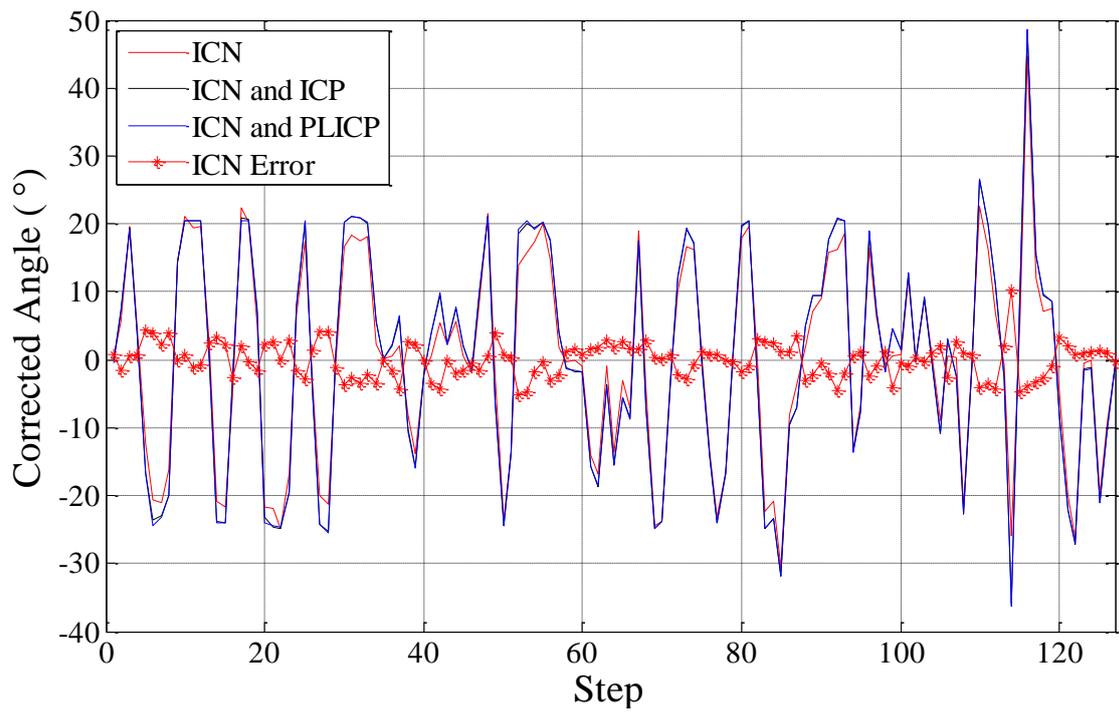


Figure 2-18 The rotations corrected by three algorithms at each step

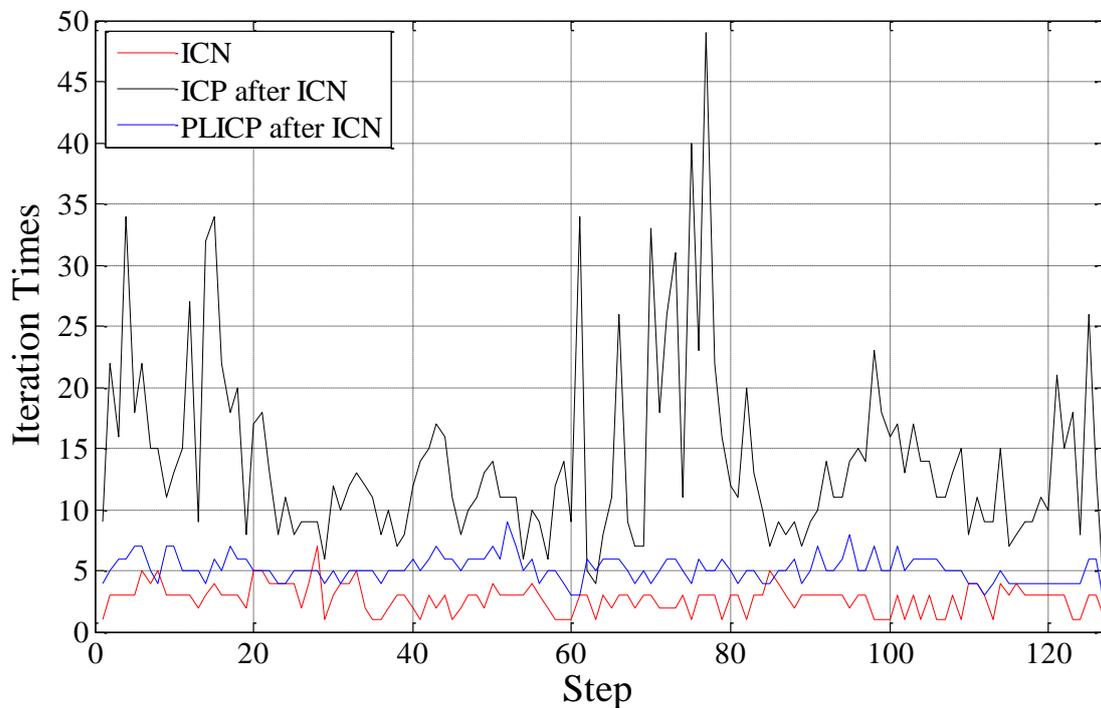


Figure 2-19 The iteration times that have been executed after applying ICN

2.8.2 Cluttered Office Room



Figure 2-20 Robot moves in cluttered office room

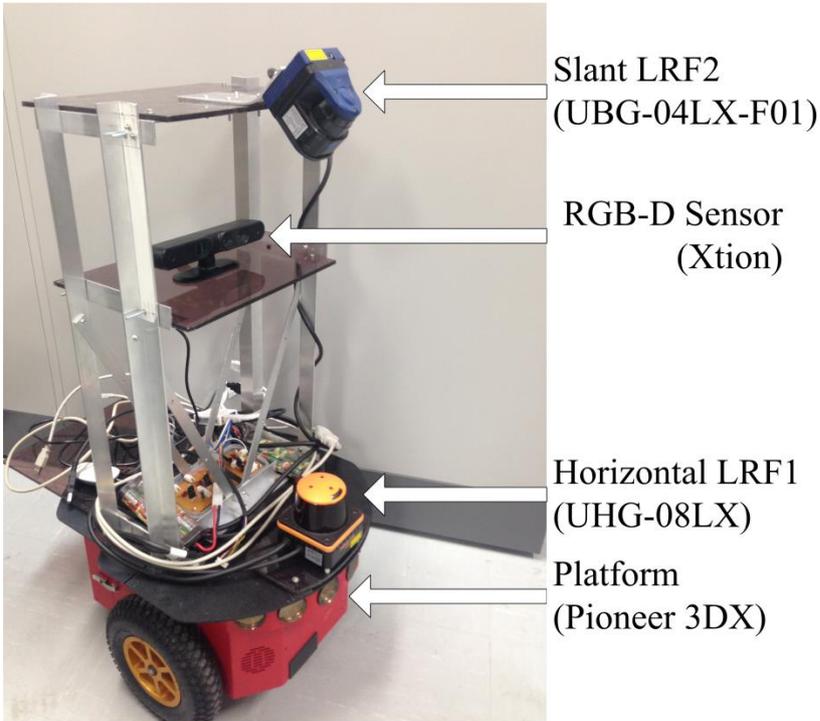


Figure 2-21 Robot platform Pioneer 3DX with sensors

The second experiment is conducted in a cluttered office room, as shown in Figure 2-20. This office room is about 4.5 meter by 7 meter. And it is filled with daily objects such as chairs, boxes, desks, printer, etc. The robot employed in this experiment is developed from famous robot platform Pioneer 3DX. The LRF UHG-08LX is placed on the forefront of the Pioneer 3DX for horizontal scanning. A three layers aluminum frame is fixed behind for robotic system extension. The bottom layer is filled with battery and some extension circuit boards. A RGB-D sensor (Xtion) is temporarily fixed on the middle layer. The second LRF UBG-04LX-F01 is slantwise placed on the top layer to keep scanning on the front ground.

Similar to the motion fashion in the previous experiment, the robot is controlled to move in the narrow office with sharp turns. The robot moves around the middle cluster of objects and finally returns to the original position as shown in Figure 2-20. And again, scans obtained by horizontally placed LRF are used for scan matching by using different methods. As expected, ICP and PLICP have to run a lot of iterations to match scans with large angular displacement, as shown in Figure 2-22. And neither ICP nor PLICP give satisfactory scan matching results, as shown in Figure 2-25. Although the trajectory estimated by ICP looks closer to the ground truth, it is difficult to judge which scan matching method performs better since both of them mismatch scans at some steps.

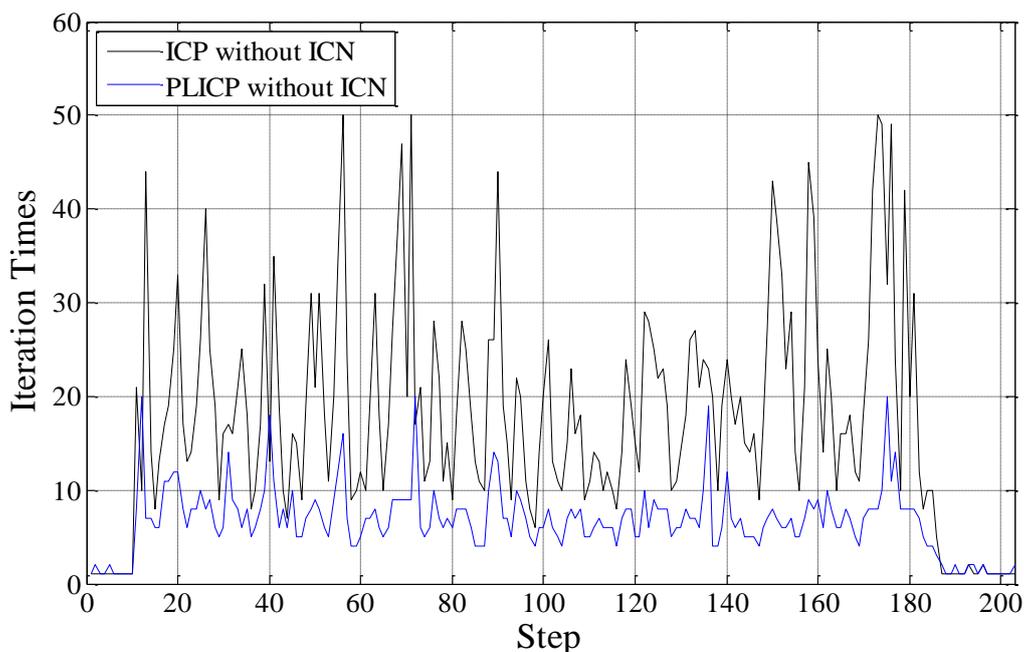
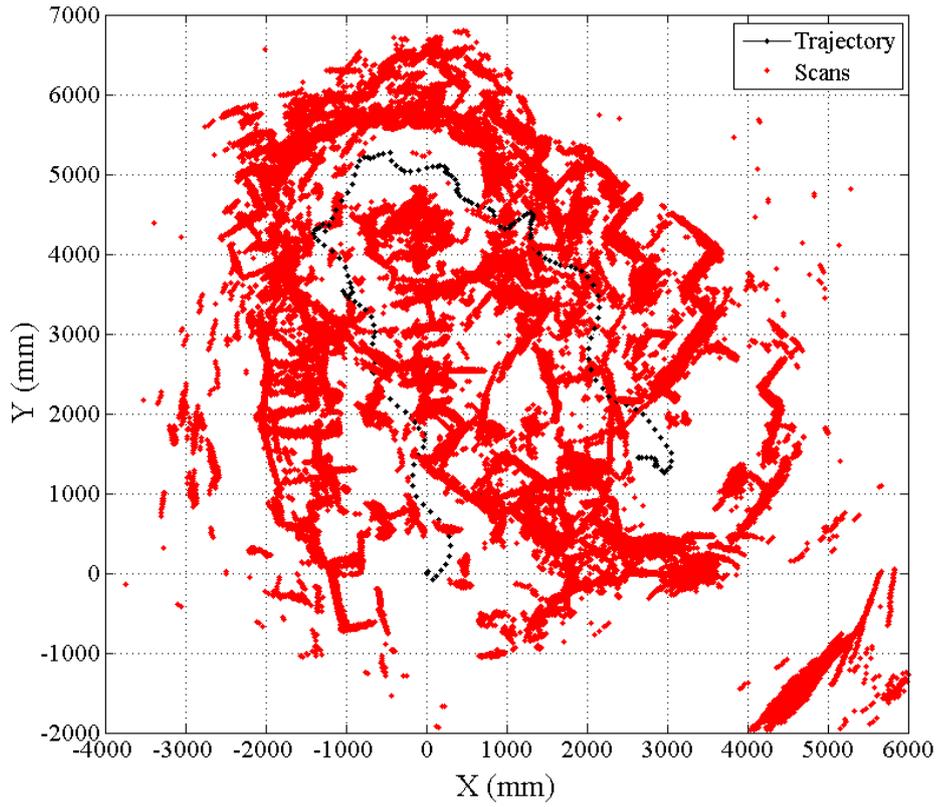
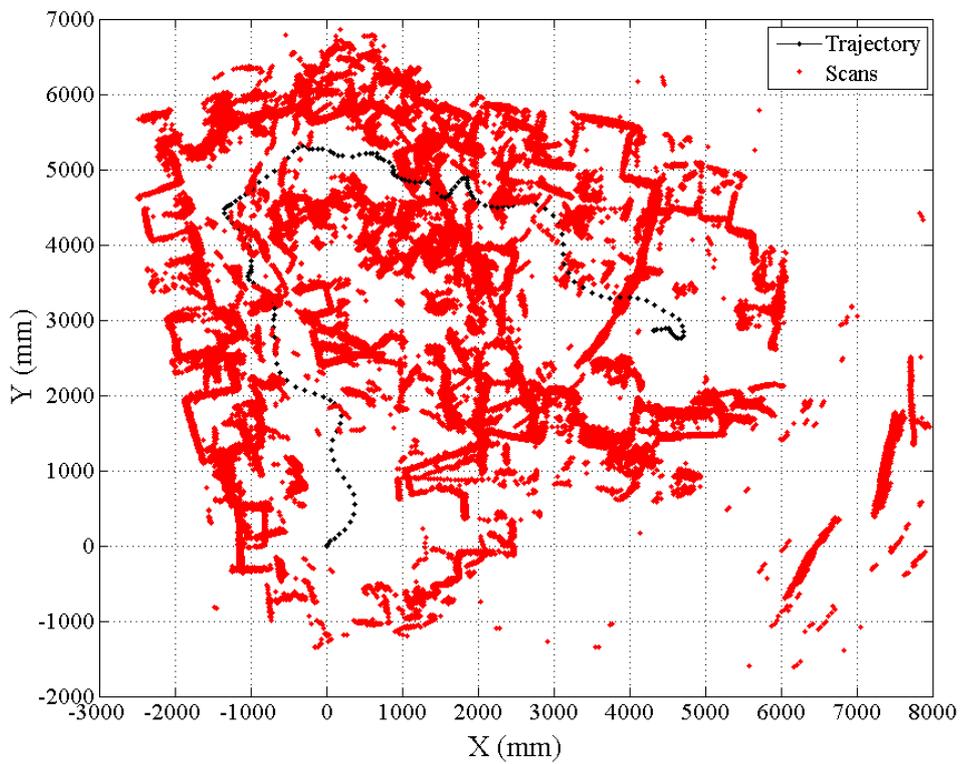


Figure 2-22 The iteration times that have been executed in the scan matching processes



(a) ICP matching result



(b) PLICP matching result

Figure 2-23 Successive scans iteratively matched by ICP and PLICP

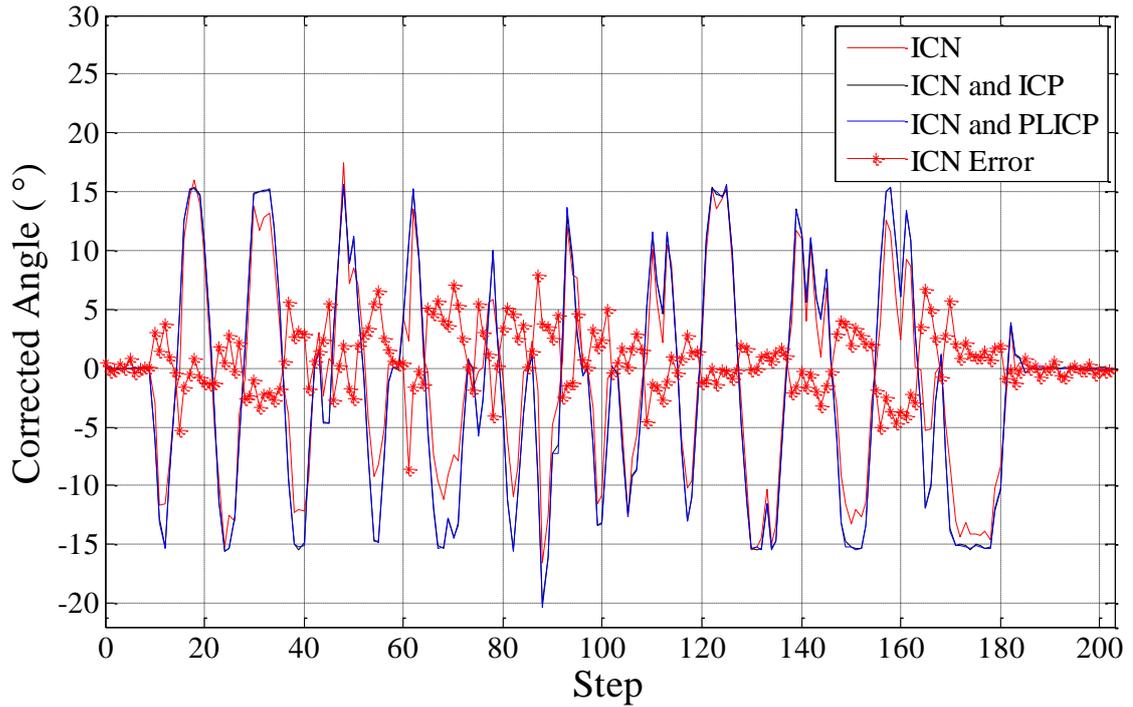


Figure 2-25 The rotations corrected by three algorithms at each step

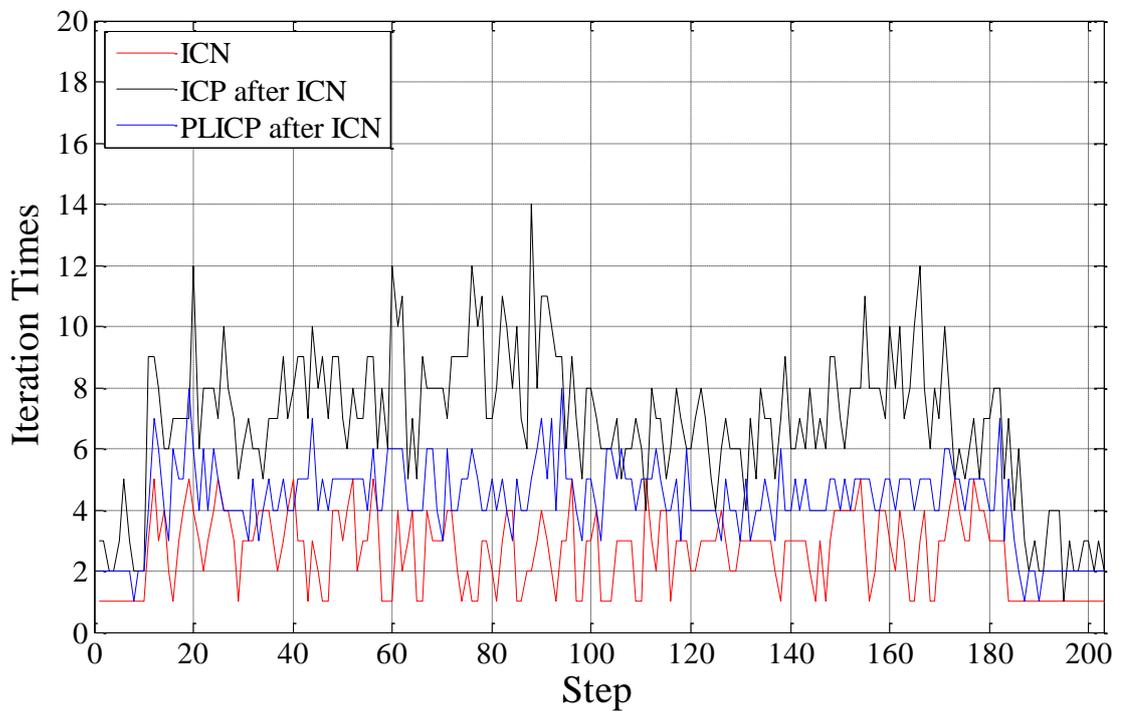


Figure 2-26 The iteration times that have been executed in the scan matching processes

2.8.3 Statistical Tests and Comparison

It is difficult to get the exact transformation reference at each step during the experiment, thus the errors analysis of experimental results cannot be presented. In order to estimate the performance of each algorithm, several simulation tests have been executed to all scans obtained in experiments. In each test, every experimental scan is copied and transformed with given trial transformation. This copy is then matched with the original scan by applying different algorithms. After the matching, the residual error can be easily found out and the statistical analysis has been done. The trial transformations can be divided into 4 levels according to the different translations, which increase from (100mm, 100mm) in first level to (400mm, 400mm) in forth level. In each level, the rotation is varying from 10° to 40°. Therefore, there are 16 tests in this simulation in total, as shown in Table 2-1.

The Mean Absolute Error (MAE) and Strandard Deviation (SD) of rotations estimated by using different algorithms have been plotted in Figure 2-27 and Figure 2-28, respectiely. The MAE and SD of ICP and PLICP keeps low when the test rotation is 10° or 20°, which means the scan matching results are robust under these conditions. However, the MAE and SD of ICP and PLICP increase dramatically after test rotation arrive 30° no matter whether translation is big or small. The accuracy of ICN is not sensitive to the variation of rotaions. On the contraty, it is slightly deteriorated with the increasement of translation. With regard to the rotation estimation, the performances of ICP and PLICP are both satisfactory after ICN has been applied for their MAE and SD keeps close to zero.

Table 2-1 Initial transformation of each test

	x (mm)	y (mm)	θ (°)		x (mm)	y (mm)	θ (°)
Test 1	100	100	10	Test 9	300	300	10
Test 2	100	100	20	Test 10	300	300	20
Test 3	100	100	30	Test 11	300	300	30
Test 4	100	100	40	Test 12	300	300	40
Test 5	200	200	10	Test 13	400	400	10
Test 6	200	200	20	Test 14	400	400	20
Test 7	200	200	30	Test 15	400	400	30
Test 8	200	200	40	Test 16	400	400	40

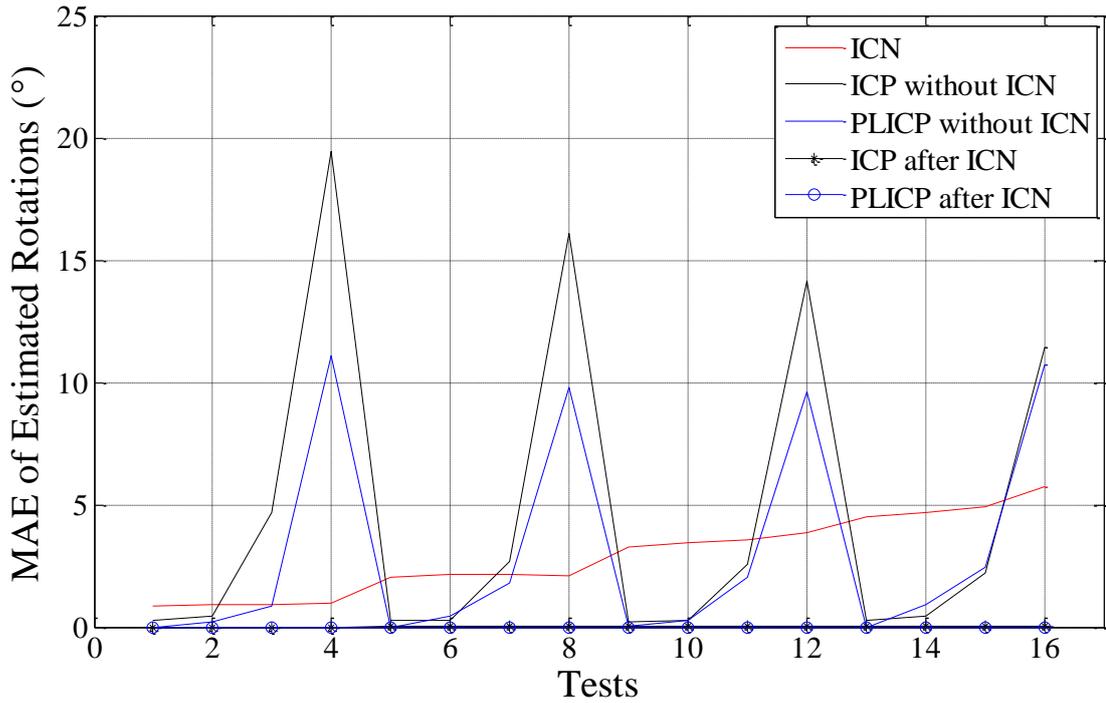


Figure 2-27 Mean Absolute Error of rotations estimated by different algorithms

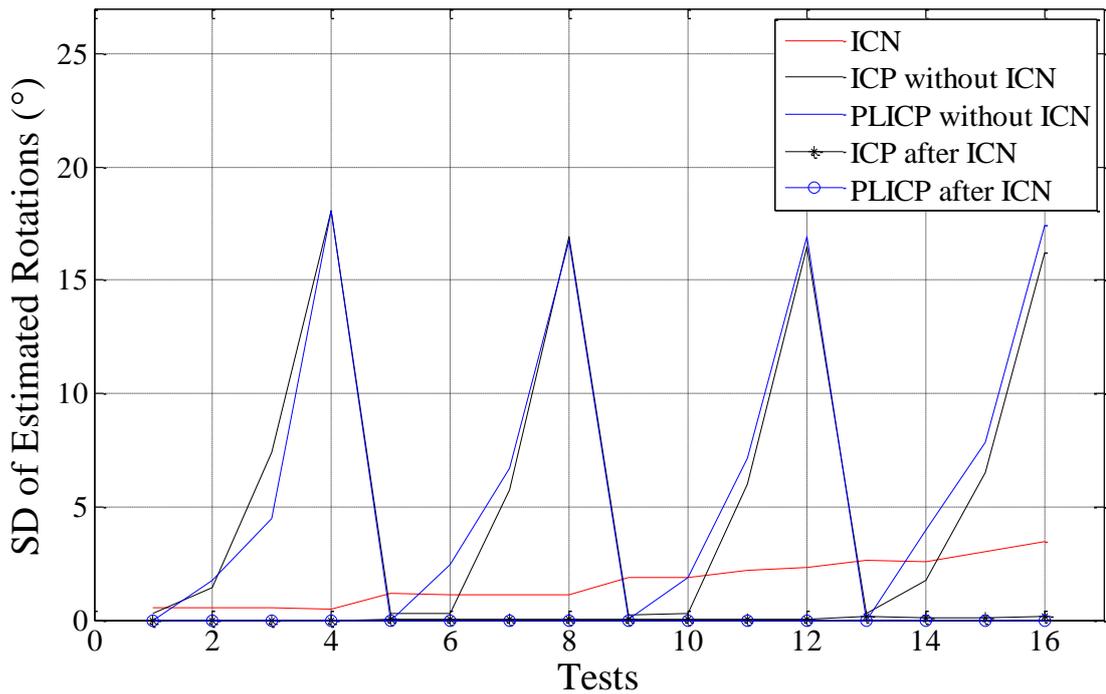


Figure 2-28 Standard Deviation of rotations estimated by applying different algorithms

Figure 2-29 and Figure 2-30 demonstrate the performances of different algorithms on translation estimation in different tests. The accuracy and stability of ICP and PLICP are prone to be deteriorate by the excessive angular displacement between scans. After ICN has been introduced and angular displacement has been bounded, PLICP becomes very

robust towards the variation of translation. On the contrary, ICP can not efficiently eliminate the translation when the initial translational component is oversized.

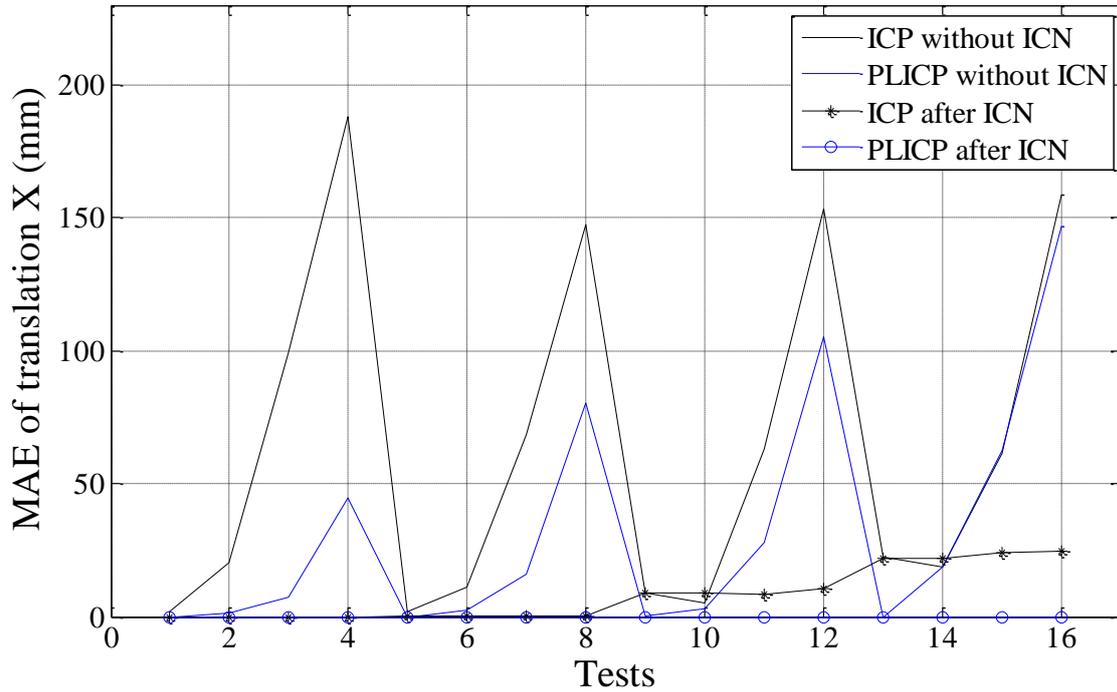


Figure 2-29 Mean Absolute Error of estimated translation along X after applying different algorithms

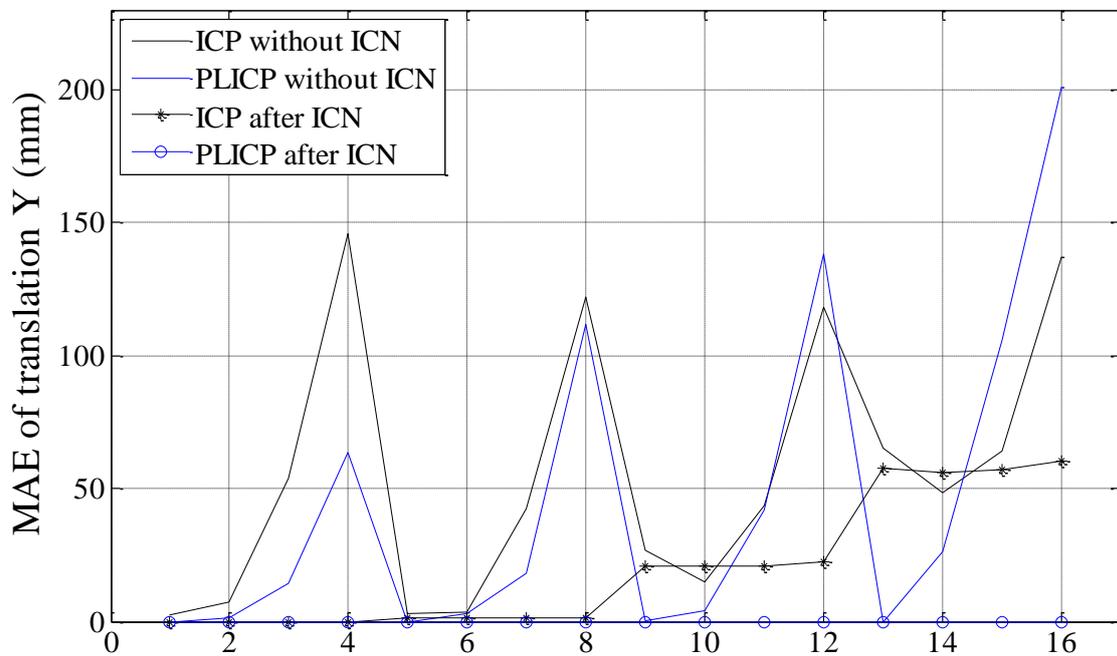


Figure 2-30 Mean Absolute Error of estimated translation along Y after applying different algorithms

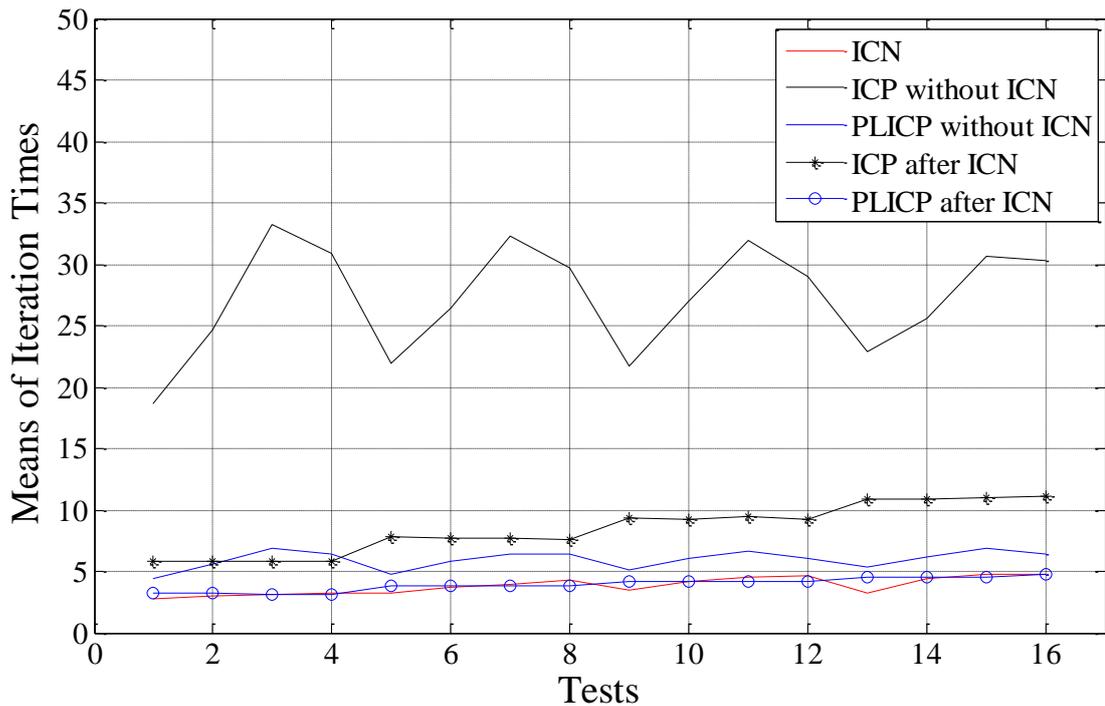


Figure 2-31 Means of executed iteration times of different algorithms

The means of executed iterations of different algorithm are plotted in Figure 2-31. Comparing with PLICP, ICP requires massive iterations to match scans. And the executed iterations of both ICP and PLICP increase with the size of rotational component. The iterations of ICN keep almost constant, varying from 3 times to 5 times no matter how much the transformation changes.

Comparison between POE and ICN

Another work has been conducted in this statistical test is the comparison between POE and ICN. Both of these two methods handle rotation estimation only and act as preprocessor of full scan matching.

The rotation estimated by ICN and POE have been compared with simulated rotation listed in Table 2-1. Means of Absolute Errors (MAE) and Standard Deviation (SD) have been obtained and plotted in Figure 2-32 and Figure 2-33, respectively. It is not difficult to figure out that the ICN is more accurate and more robust in rotation estimation since it captures lower MAE and SD in comparison with POE. Although the performance of ICN is also affected by the increment of translation, the effect is much slighter.

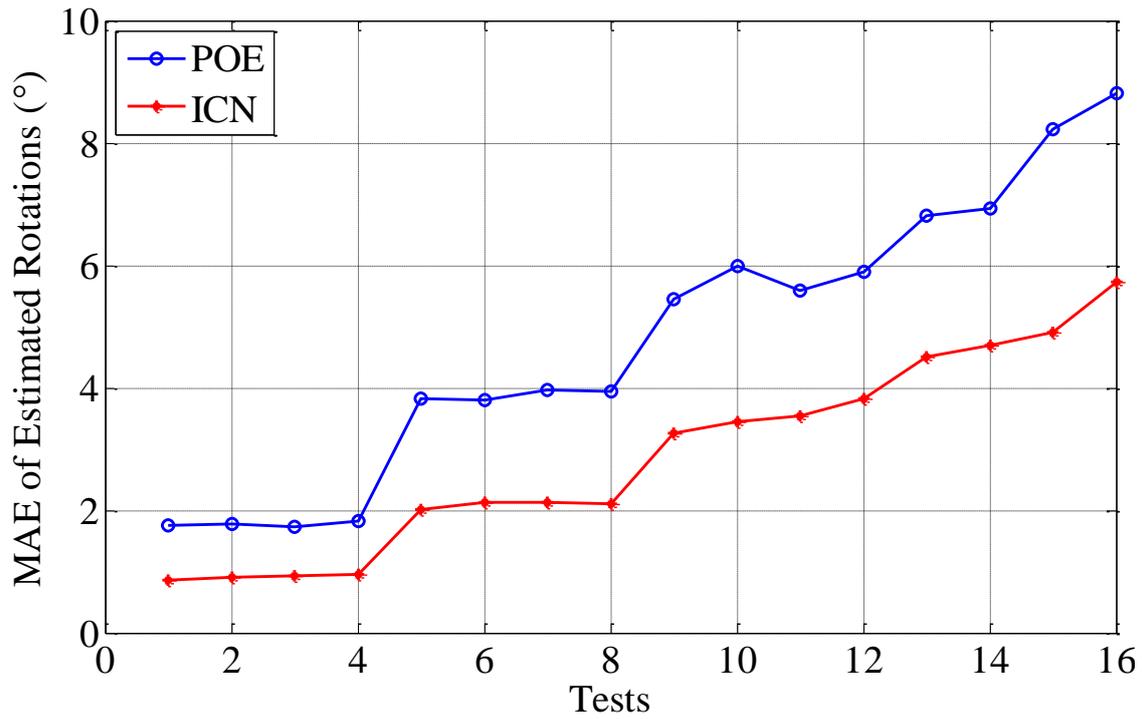


Figure 2-32 Means of absolute errors of rotations estimated by POE and ICN

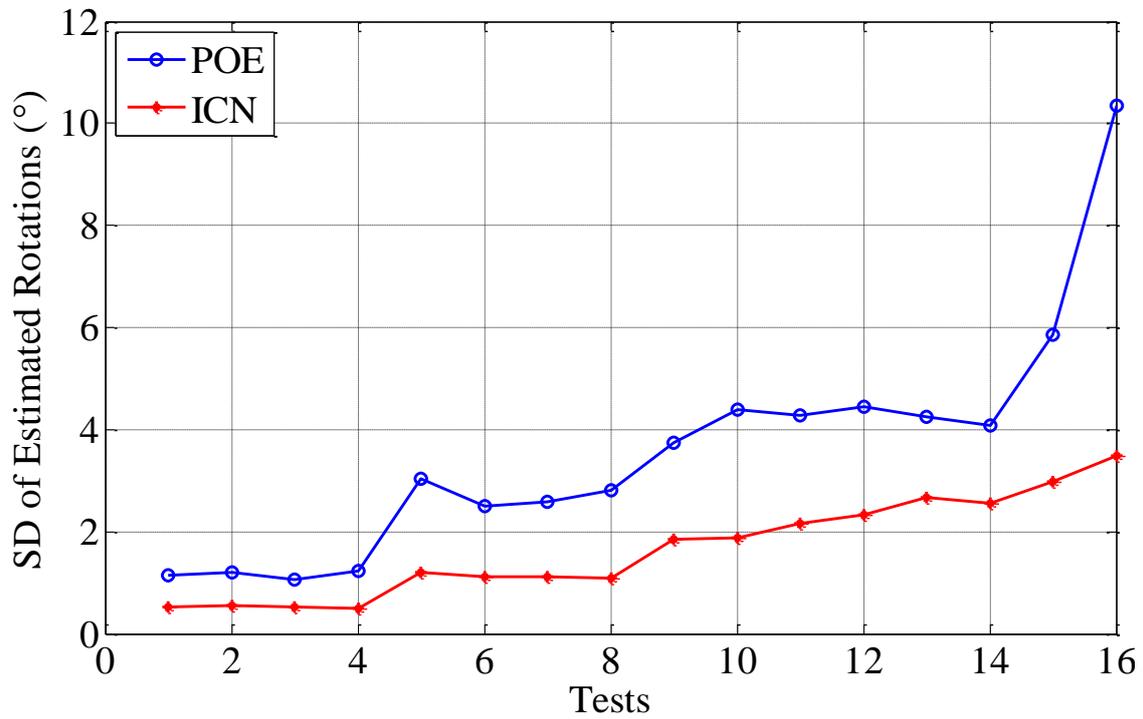


Figure 2-33 Standard deviation of rotations estimated by POE and ICN

The computational cost of ICN is similar to typical ICP which requires $O(n_p^2)$ without any accelerating method, where n_p is the number of points in the scan. POE requires $O(n_p n_s)$ for the calculation, where n_s is the shift times which is determined by search boundary and shift interval. Because n_s is smaller than n_p , POE is faster than ICN in general. However, there is method to accelerate the speed of ICN. In our approach, the points are sorted based on the range by using introspective sorting algorithm [59] with the computational cost of $O(n_p \log(n_p))$. Then the points are associated with certain difference of range with the cost of $O(n_p n_f)$, where n_f is the number of possible correspondence in association procedure of ICN and it should be much smaller than n_p . Therefore, the computational cost of ICN drops dramatically.

2.9 Conclusion

In this chapter, various scan matching methods have been introduced and the most widely approach ICP is detailed. To overcome the weakness of ICP when it matches scans with large rotation, an algorithm named ICN that can efficiently correct the unexpected large rotation between two scans has been proposed. ICN algorithm establishes the correspondence by associating the points that have similar normal vector after compensating smallest angular difference within certain range difference. The rotation is estimated by using the quadratic range as its weighting considering translational effect to the rotation and intensity of scan data. After rotation between scans has been compensated by using ICN, ICP and PLICP are applied to estimate the residual transformation, respectively. The experimental results show that ICP and PLICP are more robust and accurate after introducing the ICN algorithm. PLICP relatively performs better than ICP in the convergence speed and matching accuracy. Therefore the combination of ICN and PLICP is a better solution of LRF scan matching for indoor robot SLAM.

Chapter 3 Efficient Scan Matching in Large Scale Indoor Environment

3.1 Introduction

ICP variants are usually efficient and easy to apply, which makes it to be widely adopted in robotic researches. The most time consuming process of ICP is the association process. After reference scan and new scan being mapped into the same coordinate, for every point in one scan, association process tries to find their nearest neighboring point in another scan. This process is known as famous nearest neighbor search (NNS) problem [60][61], which is also called nearest point search or similarity search. A generalization of this problem is a k-NN search, where k nearest neighbors need to be found.

When large transformation especially large rotation exists between two scans, the validity of closest-point rule based association that adopts traditional Euclidean distance metric is prone to establish wrong correspondences. In fact, most of the fine scan matching method needs scans to be pre-aligned to avoid falling into local minimum [62].

Large scale indoor environment usually contains loops. In such kind of environment, scan matching is a difficult problem since error accumulates and cannot close large loops. High accuracy scan matching method can ease this difficulty and feasible loop closure method is the final solution [63].

In this chapter, several methods to tackle NNS problem are firstly introduced. Then, based on the analyzed good properties of geometric incident angle, two approaches are proposed. The first approach is a pre-alignment method which is able to estimate large rotation and dominant translation between scans without iteration. The second approach is a unique distance metric which can dramatically improve the performance of closest-point rule based association process. Thirdly, a framework of increment scan matching for large loop existing indoors is detailed. Finally, the efficiencies of proposed algorithms are verified by experiments and analysis.

3.2 Nearest Neighbor Search

In this study, the NNS problem is simply defined as follows: given a reference scan \mathcal{S}_{ref} with a point \mathbf{p}_{new} from new scan \mathcal{S}_{new} , find the closest point in \mathcal{S}_{ref} to \mathbf{p}_{new} . There are various methods have been proposed to solve NNS problem. The performance of the methods is evaluated by examining the complexity of querying process as well as the complexity of the search data structures that have to be maintained.

3.2.1 Linear Search

Linear search, or brute force method, is the simplest way to solve the NNS problem. This kind of method simply computes the distances from the query point \mathbf{p}_{new} to every point in the reference scan \mathcal{S}_{ref} with updating the record of "closest point so far", as shown in Figure 3-1. The advantage of this method is that it needs no search data structures to maintain, which reduces the memory storage cost in comparison with the following space partitioning approaches. However, its heavy computational cost prevents it from application in large data set. For a reference scan \mathcal{S}_{ref} that has totally n_{ref} points, this naive approach has to search n_{ref} times for each query point, which make the computational cost arrive $O(n_{ref})$. Assume there are n_{new} points in the new scan, the total query cost will reach $O(n_{new}n_{ref})$, which is a heavy burden on calculation cost even for nowadays' powerful computer.

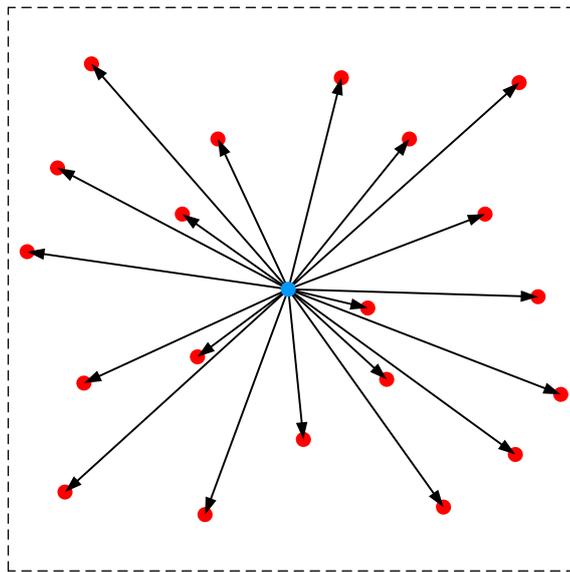


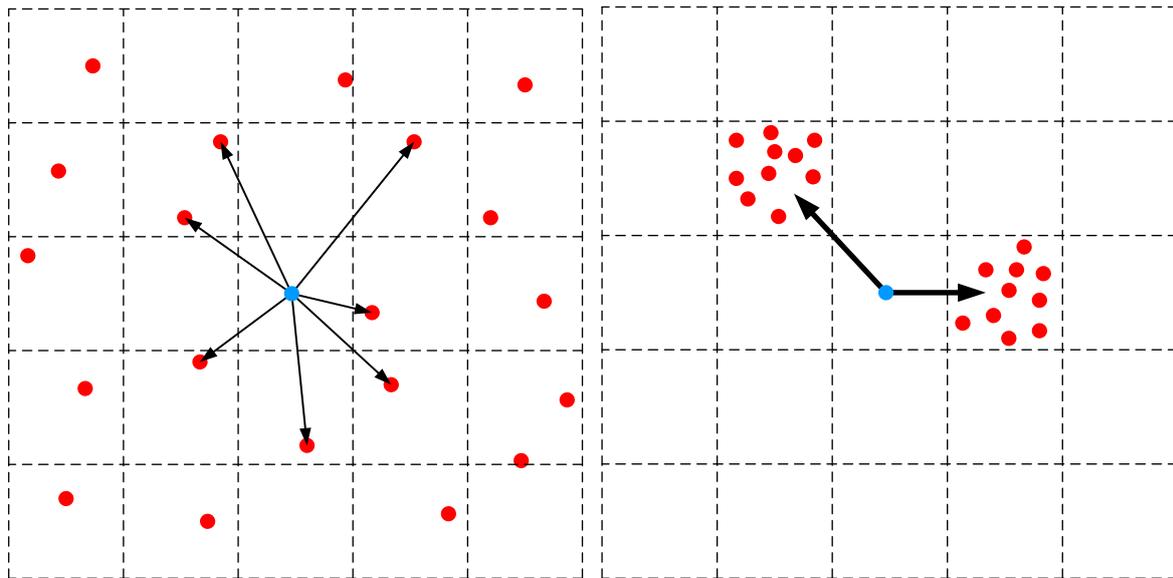
Figure 3-1 Linear Search on NNS problem

3.2.2 Grid Implementation

Another straightforward implementation is to divide data space into grid of squares. Then every reference point is affiliated to a single grid based on its position in the space. When a query point p_{new} is given, it is temporary affiliated to a grid. If there are reference points in the same grid, the nearest neighbor will be found by applying linear search on the reference points in this grid. Otherwise, from the near to the distant, surrounding grid will be included to the linear search process until the nearest neighbor has been found.

Assume the space that covers all the data points is dividing in to M grids, the linear search needs to averagely check n_{ref}/M points for a single grid if the reference data is evenly distributed in the space, as shown Figure 3-2 (a). In this case, the NNS process will be sharply accelerated since linear search only needs to check few grids with few points instead of the whole reference data set. However, the computational cost of NNS process will increase dramatically if the reference data is clustering in few grids, as shown in Figure 3-2 (b).

The number and size of grid is another problem of this application. A large number of small grids require large memory cost while a small number of large grids have to put too many points in grid. Usually the number of grids is set equal to the number of points.



(a) Evenly distributed reference data

(b) Clustering reference data

Figure 3-2 Grid based nearest neighbor search

3.2.3 K-d Tree

As a widely adopted solution in NNS problem, k-dimensional tree (k-d tree) is a binary tree that stores k-dimensional data as node of the tree, which is initially introduced in [64]. According to cycling dimension, k-d tree splits the space into hyper-rectangle by using splitting hyper-plane and subdivides data at each recursive level of the tree. For example, to build a k-d tree from two-dimensional points that comprise (x; y) coordinates, the splitting hyper-plane is a line that parallel to some axis and the split dimension would be cycled as x; y; x; y..., as shown in Figure 3-3. A more elaborate scheme for cycling the split dimension chooses the one that has the widest dispersion or largest variance to be the split dimension for a particular level of recursion.

Given a query point, the initial guess on the nearest neighbor in the k-d tree can be set to the result of last search or the root of the k-d tree. By using k-d tree, the procedure of NNS is executed by comparing "closest point so far" with nodes along the tree. The hyper-rectangle will be removed out of the search area if the distance between query point and the closest hyper-plane of the hyper-rectangle is longer than the distance between query point and the "closest point so far".

A well balanced k-d tree recursively splitting data through the median. And NNS of a query point through a balanced k-d tree is very efficient, as shown in Figure 3-4. On average, it requires $O(\log n_{ref})$ accessing behavior per NNS process. However, the depth of a badly unbalanced tree becomes very large, as shown in Figure 3-5. And it makes a single NNS process to traverse much more data points, as shown in Figure 3-6. In the worst case, it is approaching to $O(n_{ref})$ accessing behavior per NNS process [65].

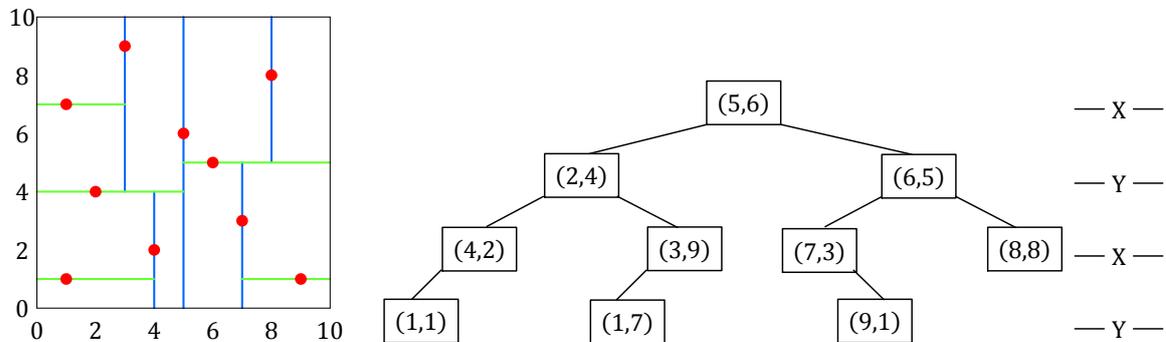


Figure 3-3 Balanced k-d tree

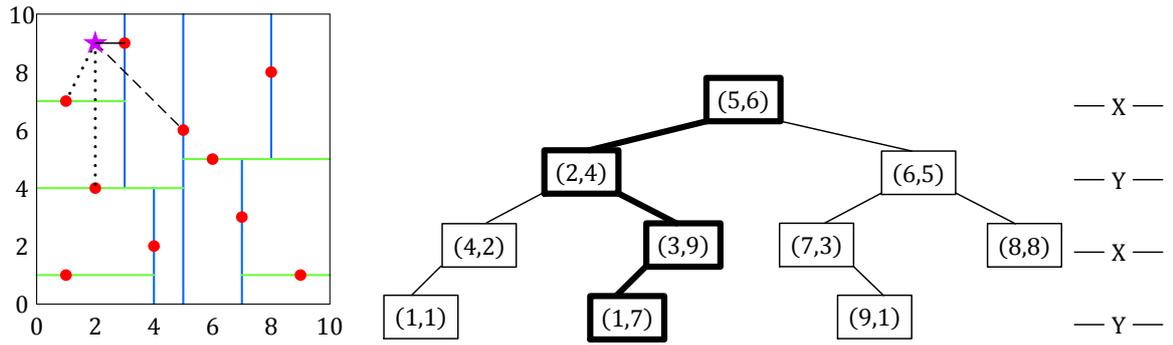


Figure 3-4 NNS by using balanced k-d tree

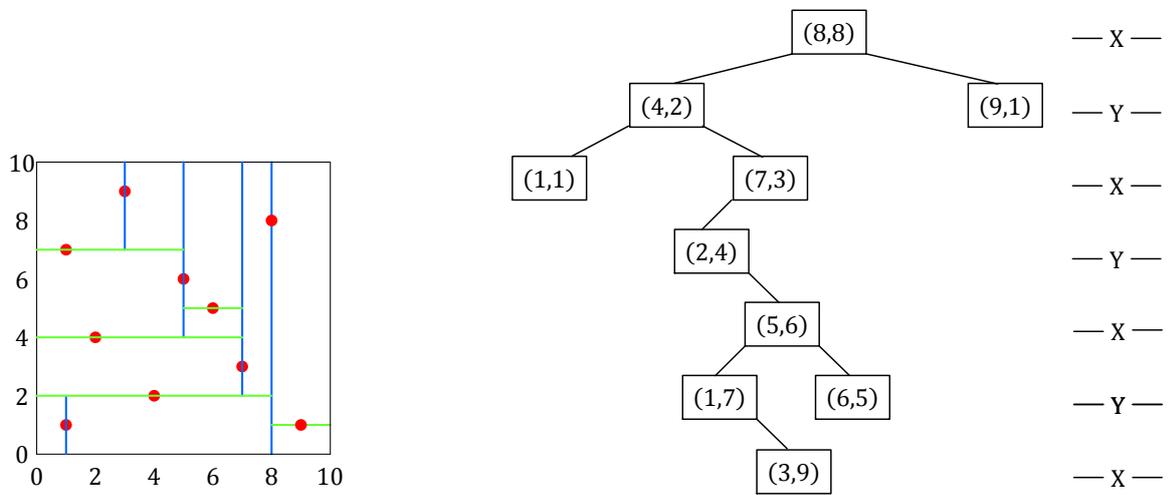


Figure 3-5 Unbalanced k-d tree

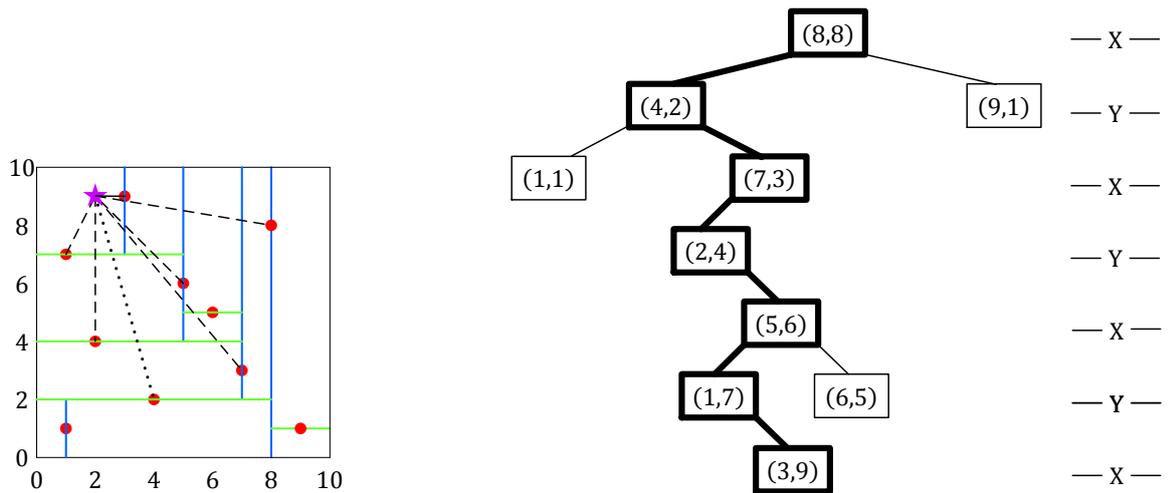


Figure 3-6 NNS by using unbalanced k-d tree

3.3 Incident Angle Utilization

A good initial estimation of transformation that is calculated from odometer readings can make two scans roughly aligned. As a result, the true correspondences can be associated based on closest point rule, which enable ICP to efficiently eliminate the residual transformation error. However, odometer suffers from unpredictable slippage, which makes the prior estimation of transformation unreliable. Thus a robust association approach between two scans without valid initial transformation estimation should be guaranteed. Given two points that are obtained from same geometric point by two scans, the incident angles are found to be insensitive to the small translation and invariant to rotation, which is a desired property for points association. In this study, incident angle is introduced for association process.

3.3.1 Analysis on Incident Angle and Transformation

Incident Angle

Assume robot that is equipped with a perfect range sensor is moving around a rectangular object. The middle point \mathbf{p}_{mid} of a surface of the object is scanned by the sensor and saved as \mathbf{p}_{ref} with respect to the robot frame $O-X_R Y_R$, as shown in Figure 3-7. In this study, the incident angle γ of a scanned point is defined as the angle from the polar line to the normal vector line. Thus the incident angle of \mathbf{p}_{ref} is calculated by using radial angle and normal vector angle as follow:

$$\gamma_{ref} = \vartheta_{ref} - \varphi_{ref} - \pi, \quad (3-1)$$

where φ_{ref} is the radial angle and ϑ_{ref} is the normal vector angle of the point \mathbf{p}_{ref} .

After robot moved a certain distance, the middle point \mathbf{p}_{mid} is again scanned by sensor and saved as \mathbf{p}_{new} with respect to the moved robot frame $O-X_R Y_R$, as shown in Figure 3-8. Similarly, the incident angle of point \mathbf{p}_{new} is calculated as

$$\gamma_{new} = \vartheta_{new} - \varphi_{new} - \pi. \quad (3-2)$$

Since the sensor is assumed to be perfect, the estimated normal vectors \mathbf{v}_{ref} and \mathbf{v}_{new} should be well aligned with the actual normal vector \mathbf{v}_{mid} .

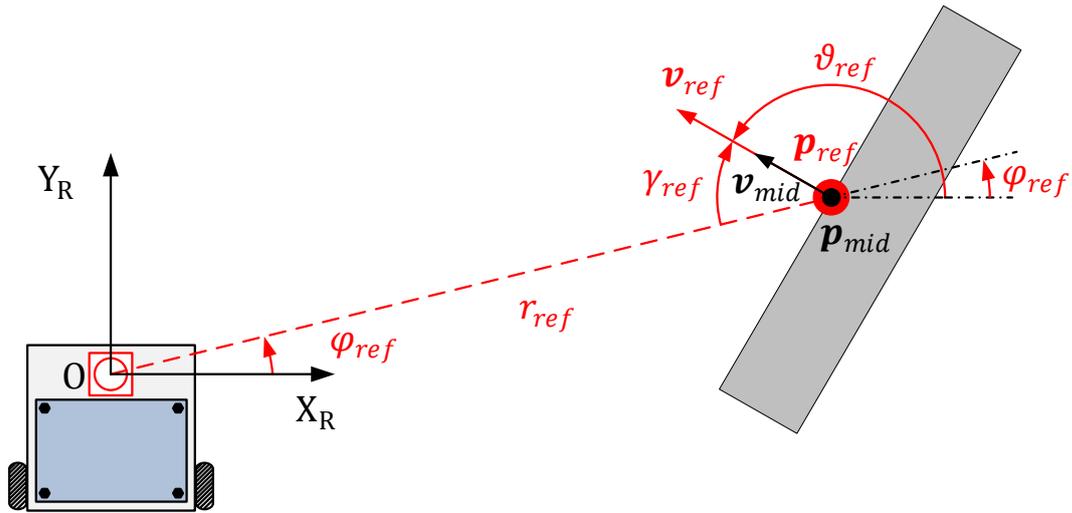


Figure 3-7 First scan on the middle point of the rectangular object

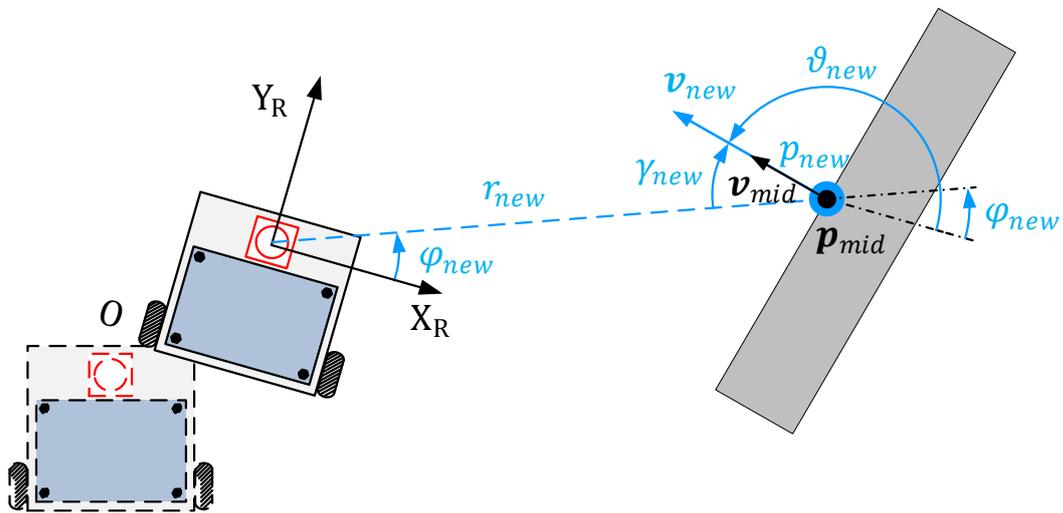
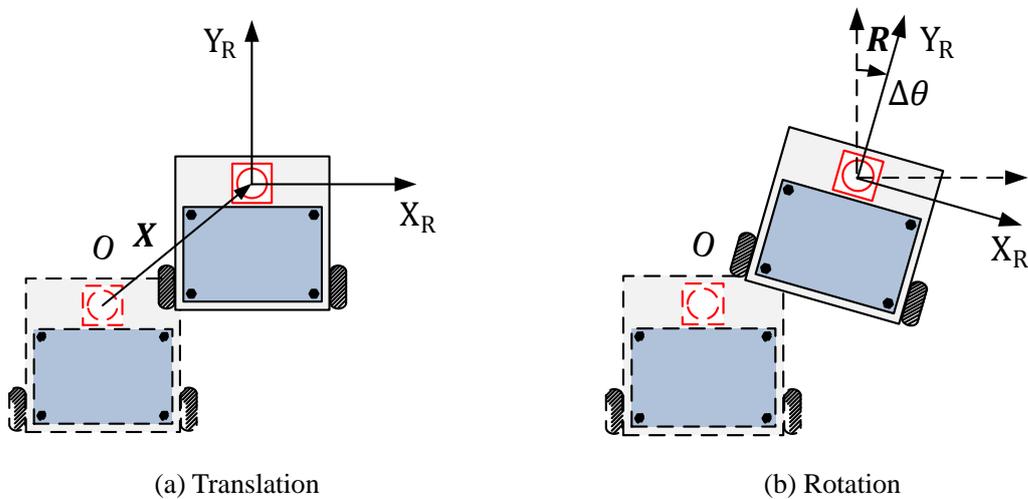


Figure 3-8 Second scan on the middle point of the rectangular object



(a) Translation

(b) Rotation

Figure 3-9 Robot transformation decomposition

A difference angle $\Delta\gamma$ between two incident angles γ_{ref} and γ_{new} can be found. Since the sensor is assumed to be perfect and two points are exactly scanned from the same geometric point, the difference angle is introduced by the motion of robot only. In order to explicitly figure out the effect brought by different components of transformation, the robot motion is decomposed into translation and rotation, as shown in Figure 3-9.

Effect of Robot Translation on Incident Angle

It is difficult to find out the relations between translation \mathbf{X} and incident angle change by using formula derivation. Fortunately, it is easy to figure out the relations from the geometrics that have been plotted in Figure 3-10. The difference of incident angle that is introduced by translation $\mathbf{X} = (\Delta x, \Delta y)^T$ can be estimated as

$$\Delta\gamma = \sin^{-1}\left(\frac{|\mathbf{X}| \sin(\Delta\theta_X - \varphi_{ref})}{r_{new}}\right), \quad (3-3)$$

where the translation distance $|\mathbf{X}| = \sqrt{\Delta x^2 + \Delta y^2}$ and translation orientation $\Delta\theta_X = \tan^{-1}(\Delta y/\Delta x)$. Considering low robot speed in indoors, two approximations are adopted:

$$|\mathbf{X}| \ll r_{ref} \text{ and } r_{new} \approx r_{ref}. \quad (3-4)$$

By using above proposition as well as small angle approximation, the change of incident angle can be approximately calculated as

$$\Delta\gamma \approx \sin^{-1}\left(\frac{|\mathbf{X}| \sin(\Delta\theta_X - \varphi_{ref})}{r_{ref}}\right) \approx \frac{|\mathbf{X}|}{r_{ref}} \sin(\Delta\theta_X - \varphi_{ref}). \quad (3-5)$$

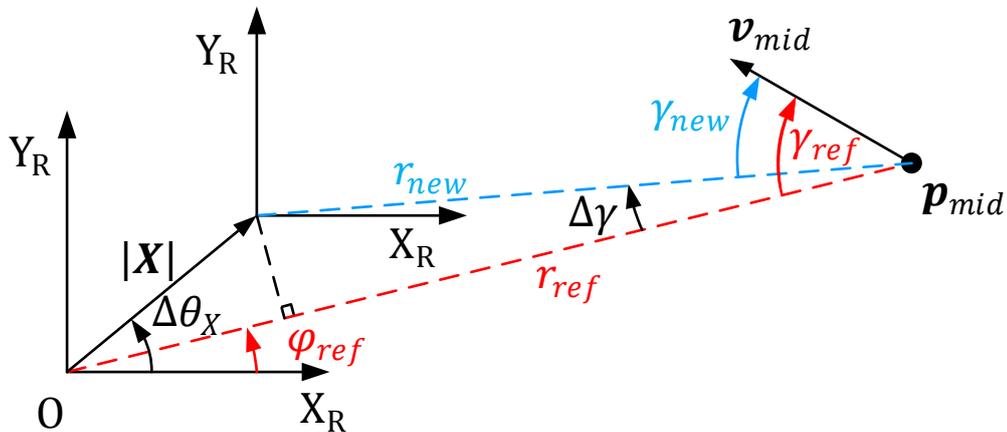


Figure 3-10 Incident angle changes because of translation

The change of incident angle reaches maximal value $\frac{|X|}{r_{ref}}$ when the sinusoidal part $\sin(\Delta\theta_X - \varphi_{ref})$ equals to 1. In this case, the orientation of the translation is orthogonal to the radial line of the geometric point. Therefore, it is not difficult to draw the conclusion that the change of incident angle is not sensitive to the small translation.

Effect of Robot Rotation on Incident Angle

The effect of robot rotation on incident angle is easy to figure out. The change of normal vector angles equals to the rotational angle as

$$\vartheta_{new} = \vartheta_{ref} - \Delta\theta. \quad (3-6)$$

In the case of pure rotation without translation, the change of radial angles also equals to the rotational angle as

$$\varphi_{new} = \varphi_{ref} - \Delta\theta. \quad (3-7)$$

By substitute Equation (3-6) and (3-7) into (3-2), it is easy to find that

$$\gamma_{new} = \gamma_{ref}, \quad (3-8)$$

which indicates that incident angle is invariant from the robot rotation. This conclusion can also be intuitively found out from the geometrical relations that are plotted in Figure 3-11. The incident angle is only affected by the position of robot. And it is invariant from the change of robot heading, namely, invariant from the rotational component in robot transformation.

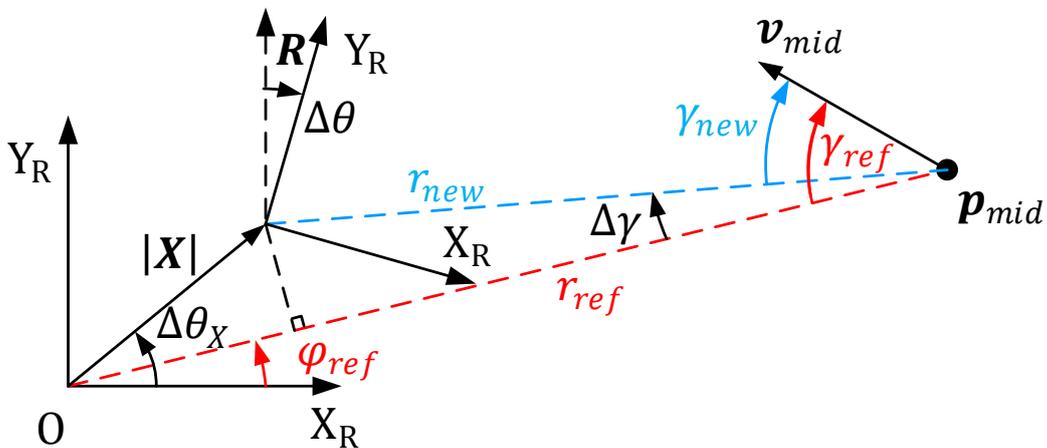


Figure 3-11 Incident angle is invariant to rotation

3.3.2 Radial Distance and Incident Angle based Alignment

Closest point rule based association which adopts traditional Euclidean distance is prone to failure when the rotational angle between two scans is large, which has been well explained in the last chapter. POE and ICN can effectively diminish the rotational angle between two scans. However, the translational information is obtained from neither POE nor ICN. Furthermore, although ICN gives more accurate and more robust estimation on rotation, its computational cost is non-negligible due to the range based sorting process.

Radial distance and Incident angle based scan Alignment (RIA), which can efficiently estimate the relative transformation between two scans, is proposed in this study. This approach is designed as a fast preprocessor for fine scan matching and it is able to pre-align scans that have large rotational and certain translational difference.

The basic concept of RIA is that the points obtained from same geometric point by two scans should capture similar incident angles and similar radial distances. Therefore the true point to point correspondences set should be included in the larger correspondences set that are associated by accounting for incident angle and radial distance only.

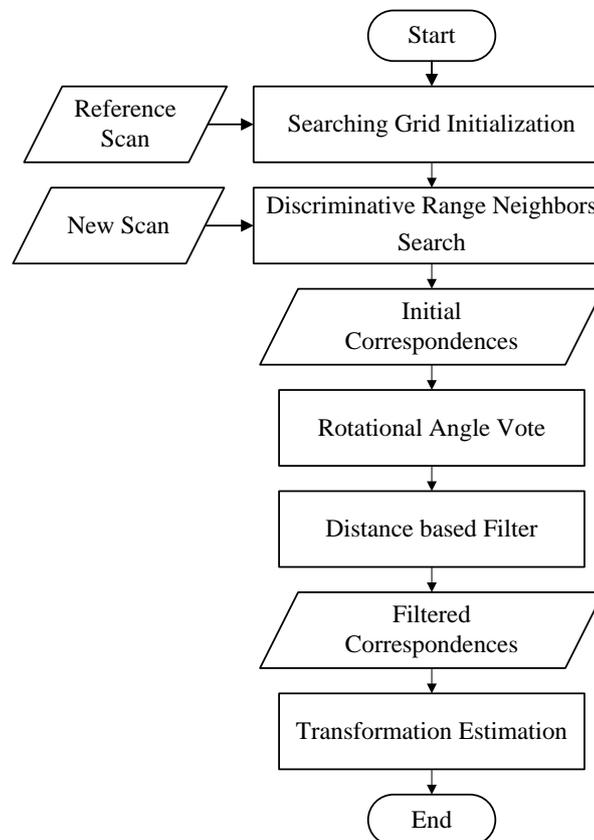


Figure 3-12 Flowchart of RIA based Approach

Algorithm Procedures

The flowchart of RIA is plotted in Figure 3-12. And the procedures of RIA can be summarized as follows:

- 1) **Searching Grid Initialization:** According to the radial distances and incident angles, the points of reference scan \mathcal{S}_{ref} are distributed into a grid space, as shown in Figure 3-13 (a). The horizontal axis of grid represents radial distance and vertical axis stands for incident angle. The horizontal interval of the grid, Δr_G is set equal to a distance that is smaller than possible maximal translation of two scans. The interval of the incident angle, $\Delta \gamma_G$ is set to be a small angle that balances storage cost of grid with the accuracy of association.
- 2) **Discriminative Range Neighbors Search:** Given a query point in the new scan \mathcal{S}_{new} , namely \mathbf{p}_{new} , it is firstly mapped into the grid space according to its radial distances r_{new} and incident angles γ_{new} . The points of \mathcal{S}_{ref} that locate in a certain range of grids are associated to \mathbf{p}_{new} as a part of initial correspondences. The neighboring number of grids in horizontal direction, n_{r_G} is a fixed value for every query point \mathbf{p}_{new} and it is calculated as

$$n_{r_G} = \left\lceil \frac{|X|_{max}}{\Delta r_G} \right\rceil \quad (3-9)$$

where $|X|_{max}$ represents for maximal translation that could be found between two scans, and operator $\lceil \cdot \rceil$ stands for ceiling function.

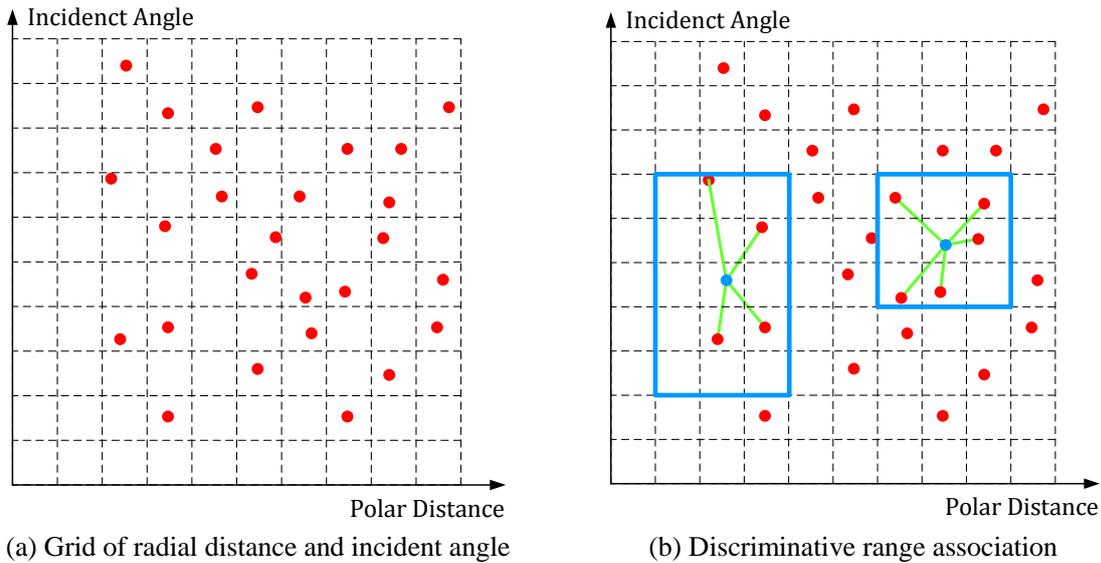


Figure 3-13 Grid based discriminative range Neighbors Search

This fixed value allows query point \mathbf{p}_{new} to find all the references points that have similar radial distance. The neighboring number of grids in horizontal direction, n_{γ_G} is a discriminative value for different query point \mathbf{p}_{new} according to the radial distance r_{new} and it is calculated as

$$n_{\gamma_G} = \left\lceil \frac{|\mathbf{X}|_{max}}{r_{new}\Delta\gamma_G} \right\rceil. \quad (3-10)$$

This setting is reasonable since the incident angle of points that have longer radial distance r_{new} changes smaller according to the analysis in the last sub-section.

All the reference points that fall in the neighboring box defined by n_{r_G} and n_{γ_G} are associated to query point \mathbf{p}_{new} , as shown in Figure 3-13 (b). Therefore, for one query point \mathbf{p}_{new} , commonly multiple correspondences are established. And the true correspondence is likely hiding inside.

- 3) **Rotational Angle Vote:** To estimate possible existing large rotation and reduce the number of false correspondences, a rotational angle voting process is conducted after the grid-based neighbors search. For every correspondence, an estimation of rotational angle can be calculated by comparing points' normal vector angles. By utilizing these estimations of rotational angle, a coarse estimation $\Delta\theta_C$ that closes to the true rotation can be found via a histogram based voting process, as shown in Figure 3-14. Furthermore, all the points' correspondences that are out of winner group can be eliminated. The new scan \mathbf{S}_{new} is then going to be rotated by using $\Delta\theta_C$.

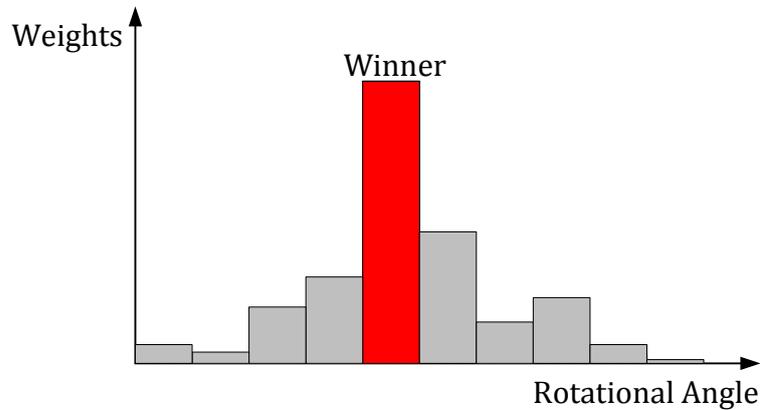


Figure 3-14 Rotational angle voting by using histogram

- 4) **Distance based Filter:** After last process, the rotational angle between two scans should be guaranteed to be a small value. Therefore the distance based filter can be applied to filter the false correspondences. The distance based filter is straightforward: if the Euclidean distance between the points of a correspondence is larger than the threshold D_{Filter} , this correspondence is treated as outlier and be eliminated from the correspondence set. For a correspondence that has reference point \mathbf{p}_{ref} , the distance threshold D_{Filter} is calculated as

$$D_{Filter} = D_{Const} + \varepsilon r_{ref} \quad (3-11)$$

where the D_{Const} is a constant value and ε is a small angular value to compensate the effect of the residual rotational angle.

- 5) **Transformation Estimation:** After the distance based filter, the majority of the correspondence set should be valid. Then the point to line distance based minimization is performed on the filtered correspondence set to estimate the residual transformation between two scans.

Since RIA is designed as a pre-processor of scan matching, it will be followed with fine scan matching process such as PLICP to obtained accurate scan matching result.

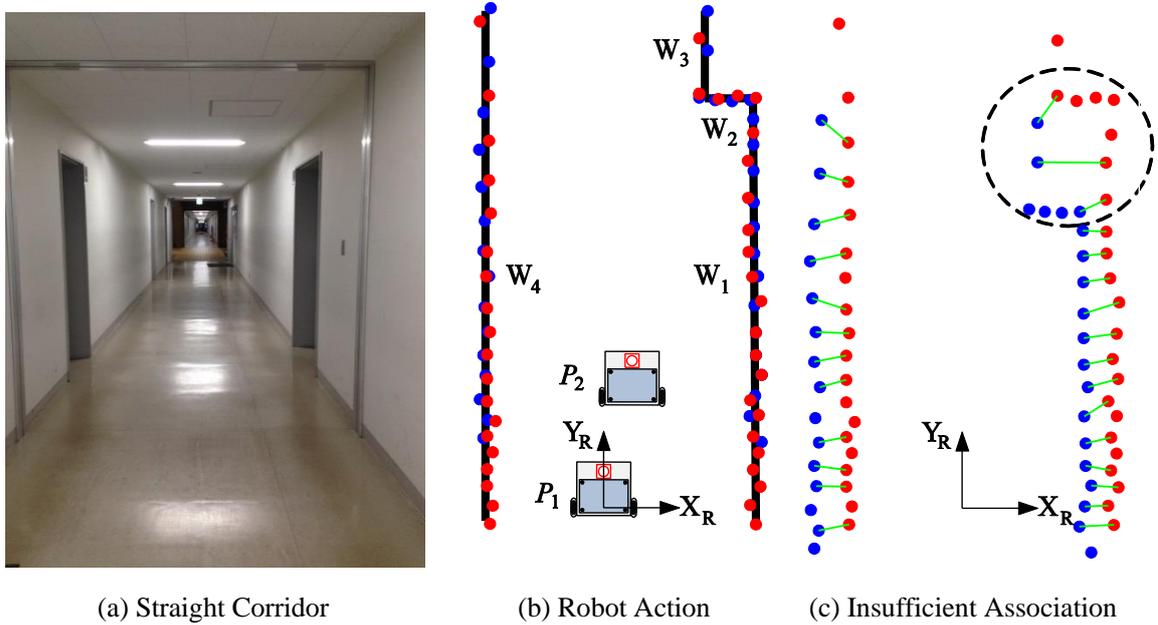
Analysis on Storage and Computational Costs

The computational costs of each step in RIA are listed in Table 3-1, where \bar{n}_{ic} stands for the number of average initial correspondences for every query points, \bar{n}_{rc} stands for the number of average residual correspondences for every query points after rotational angle voting, n_{fc} stands for the number of total filtered correspondences. Since the exact sizes of \bar{n}_{ic} , \bar{n}_{rc} are difficult to model due to the unlimited variation of environment, it is not easy to give concrete analysis on computational cost of RIA in general. In real practice, RIA does not need iterative process and it is found out to be much faster than ICN as well as POE.

Table 3-1 Computational costs of different processes in RIA

1.Searching Grid Initialization	$O(n_{ref})$	4.Distance based Filter	$O(\bar{n}_{rc}n_{new})$
2.Range Neighbors Search	$O(n_{new})$	5.Transformation Estimation	$O(n_{fc})$
3.Rotational Angle Vote	$O(\bar{n}_{ic}n_{new})$		

3.3.3 Incident Angle Fused Metric based Association



(a) Straight Corridor

(b) Robot Action

(c) Insufficient Association

Figure 3-15 Example of the limitation of Euclidean distance based association

Limitation of Traditional Euclidean Distance Metric

In a lot of application scenario, association process that adopts traditional Euclidean distance based closest point rule is not robust enough even when there is only translational component between two scans.

In indoor environment, there is a lot of long and straight walls, as shown in Figure 3-15(a). This kind of environment is simplified to the model plotted in Figure 3-15 (b), which consists of four walls. Among the walls, W_1 , W_3 , W_4 are three parallel walls and W_2 is a short wall which is placed along orthogonal direction to others. The red points are scanned at the first pose P_1 while the blue points are scanned after robot moving forward and arriving at second pose P_2 . In this situation, the valid associations of the points scanned from wall W_2 between two scans are critical to estimate transformation from P_1 to P_2 . This is because only the points from wall W_2 give valid constraint along the direction of robot translation. However, closest-point rule based association method that adopts traditional Euclidean distance metric is difficult to establish sufficient valid correspondences even by setting large reject threshold. This can be explained by using Figure 3-15 (c), the W_2 points in the second scan are difficult to be associated to their true correspondences in the first scans. This is because of the large displacement of robot towards W_2 , which drags the points far away and makes them closer to the W_1 points in

the first scan. This situation results in the absence of valid correspondences that give constraints along the direction of Y_R . As a consequence, ICP variants cannot estimate the translational component along this direction properly, at least, cannot converge to the global minimum swiftly.

Incident Angle Fused Metric

In this study, a novel distance metric named Incident Angle Fused Metric (IAFM) is proposed to solve the insufficient association problem that has been discussed. Figure 3-16 (a) and Figure 3-16 (b), which are the enlarged drawings of marked area plotted in Figure 3-15 (c), give a comprehensive geometrical explanation of the proposed metric.

For a point $\mathbf{p}_{ref}(i) = [x_{ref}(i), y_{ref}(i), \gamma_{ref}(i)]^T$ of the reference scan and a point $\mathbf{p}_{new}(k) = [x_{new}(k), y_{new}(k), \gamma_{new}(k)]^T$ of the new scan, the proposed IAFM defines the distance between two points as follows:

$$d_{IAFM}(i, k) = \sqrt{\Delta x(i, k)^2 + \Delta y(i, k)^2 + L^2 \Delta \gamma(i, k)^2} \quad (3-12)$$

where $\Delta x(i, k) = (x_{ref}(i) - x_{new}(k))$, $\Delta y(i, k) = (y_{ref}(i) - y_{new}(k))$, $\Delta \gamma(i, k) = (\gamma_{ref}(i) - \gamma_{new}(k))$ and L is the artificial parameter to project the incident angle difference from radian domain to length scale. In this study, we set L as the maximal possible displacement of robot between two scans. Since the robot has to move slowly in indoors because of security, L is a small value and therefore the new added component only affects when the difference of the incident angle of two points is oversize.

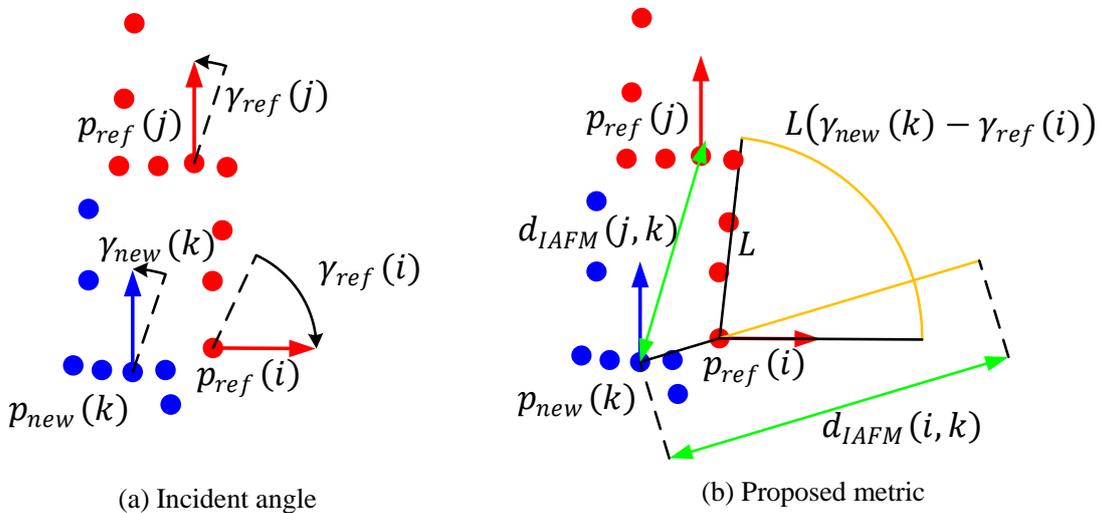


Figure 3-16 Incident angle fused metric

3.4 Points Map based Incremental Scan Matching

3.4.1 Sparse Points Map based ICP

Based on incremental scan matching concept, an efficient approach which utilizes Sparse Points Map (SPM) to represent 2D environment, SPM-ICP is presented in [66]. It maps environment and locates robot mainly based on incremental matching of LRF scans. The simplicity and efficiency of SPM-ICP are impressive. It does not need to estimate uncertainty of scan matching results since it is not a probabilistic approach. And according to the demonstrations on several datasets, it performs much faster than famous Rao-Blackwellized Particle Filters (RBPFs) based Gmapping [67] by a factor of 50 to 100 with no essential difference in the aspect of accuracy.

On the contrary, one drawback of SPM-ICP is that it cannot update its constructed map and robot trajectory when the loop is detected. If the accumulated error is out of tolerance, SPM-ICP is not able to close the loop to construct global consistent maps. Furthermore, SPM-ICP matches new scans against the whole global map, which gives a heavy burden to association process. Though SPM-ICP shows superiority in accessibility and accuracy in local area, its natural disadvantages prevent it from applying in large areas especially in the environment that contains large loops.

3.4.2 Split SPM-ICP

To overcome the drawbacks of SPM-ICP, a modified version named Split SPM-ICP (SSPM-ICP) is proposed in this study.

Architecture of SSPM-ICP

SSPM-ICP runs scan matching in the manner of sub-maps joining. The global map is split into numbers of SPMs as sub-maps. Each sub-map is coupling with a global pose named as node. The node is the global pose where the corresponding sub-map is initialized. A local map is used for the incremental scan matching and it is constructed by the sub-maps whose nodes are near to the robot. After robot has moved a certain distance or has rotated with a certain angle, the local map will be updated and a new sub-map will be generated with a node. The brief architecture of SSPM-ICP is plotted in Figure 3-17.

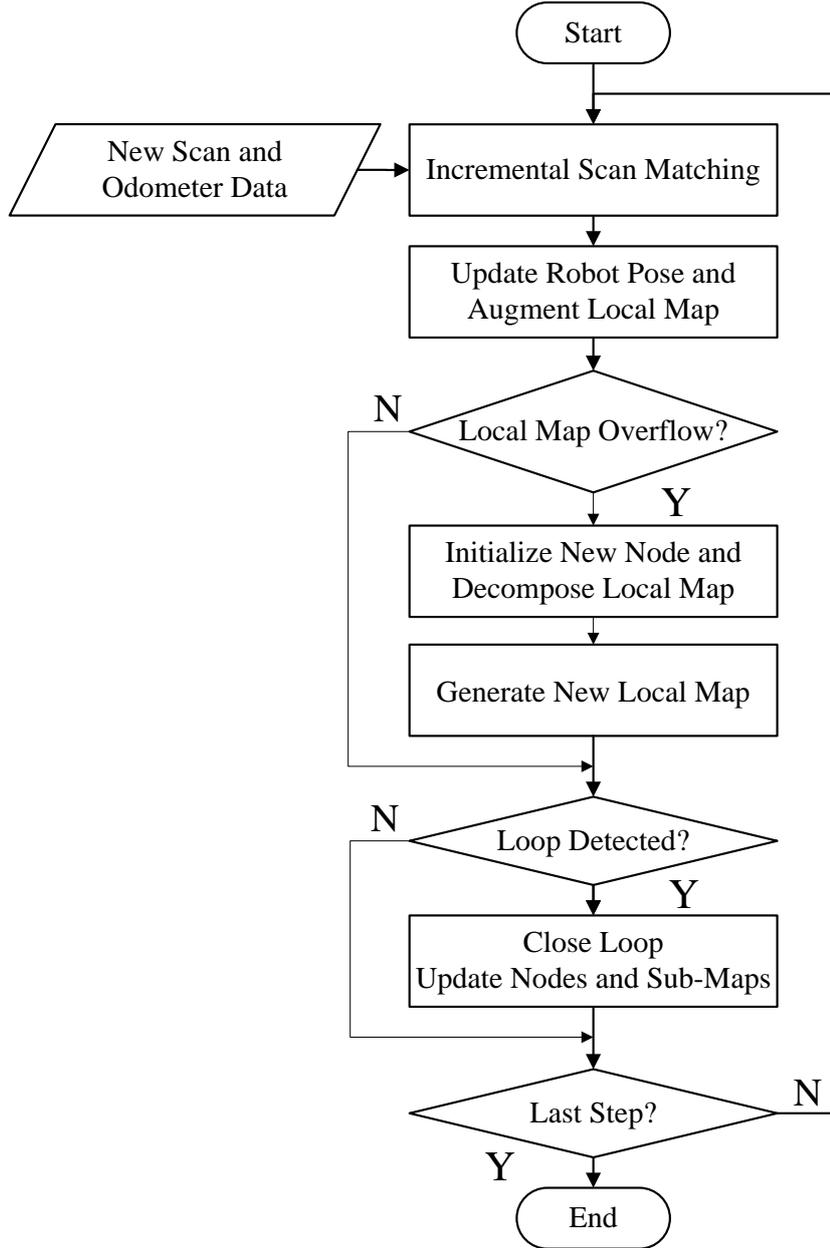


Figure 3-17 The flowchart of the SSPM-ICP

And the procedures of SSPM-ICP can be summarized as follows:

- 1. Initialization:** At time 0, the first scan \mathcal{S}_0 is going to be treated as the initial local map \mathbf{M}_{L_0} as well as first sub-map $\mathbf{M}_{Sub}(1)$. The frame of \mathcal{S}_0 is regard as the global frame. With respect to global frame, the pose of the first sub-map, node $\mathbf{N}_{Sub}(1)$, and original pose of local map, \mathbf{P}_{L_0} , as well as the robot pose \mathbf{P}_0 , are initialized as $\mathbf{N}_{Sub}(1) = \mathbf{P}_{L_0} = \mathbf{P}_0 = [0 \ 0 \ 0]^T$.

2. Relative Motion: Assuming at time $t - 1$, the robot pose is \mathbf{P}_{t-1} . The updating local map $\mathbf{M}_{L_{t-1}} = \sum \mathbf{M}_{Sub}(i)$, where $i \in \mathbf{B}_{Sub(t-1)}$ and $\mathbf{B}_{Sub(t-1)}$ is the sequence collection of sub-maps that compose local map. The original pose of local map $\mathbf{M}_{L_{t-1}}$ is $\mathbf{P}_{L_{t-1}}$. When a scan \mathbf{S}_t arrived at time t , as shown in Figure 3-18, the dead-reckoning method is conducted based on the odometer readings and its result \mathbf{T}_t^{odom} is compared with previous relative transformation \mathbf{T}_{t-1} estimated by scan matching. If both translational difference and rotational difference are below a preset threshold, the dead-reckoning result is adopted as the priori estimation of the relative motion, $\bar{\mathbf{T}}_t = \mathbf{T}_t^{odom}$. Otherwise the previous relative transformation will be employed as $\bar{\mathbf{T}}_t = \mathbf{T}_{t-1}$. The selection is based on constant velocity assumption, which can efficiently exclude outlier readings from odometer.

3. Matching: The priori estimation of robot pose at time t , $\bar{\mathbf{P}}_t$ is going to be estimated by

$$\bar{\mathbf{P}}_t = \mathbf{P}_{t-1} \oplus \bar{\mathbf{T}}_t, \quad (3-13)$$

The new scans \mathbf{S}_t is transformed based on pose $\bar{\mathbf{P}}_t$ and then matched against the local map $\mathbf{M}_{L_{t-1}}$ to obtain the estimation of robot pose \mathbf{P}_t . The posterior relative transformation of robot from time $t - 1$ to t is calculated as

$$\mathbf{T}_t = \mathbf{T}_{\bar{\mathbf{P}}_{t-1}}^{-1} \mathbf{T}_{\mathbf{P}_t}. \quad (3-14)$$

where $\mathbf{T}_{\bar{\mathbf{P}}_{t-1}}^{-1}$ is the inverse of transformation matrix of pose $\bar{\mathbf{P}}_{t-1}$.

4. Local Map Augment: Based on the scan matching, the points of transformed scan \mathbf{S}_t that has not been associated to local map $\mathbf{M}_{L_{t-1}}$ are going to be treated as new information of the exploring environment. These points will be added into local map to obtain priori local map at time t , $\bar{\mathbf{M}}_{L_t} = \mathbf{M}_{L_{t-1}} \cup \mathbf{S}_t$, as shown in Figure 3-19.

5. Judgment: With the known pose \mathbf{P}_t , the transformation of robot with respect to original pose of local map $\bar{\mathbf{M}}_{L_t}$ is calculated as

$$\mathbf{T}_{L_t} = \mathbf{T}_{P_{L_{t-1}}}^{-1} \mathbf{T}_{P_t} \quad (3-15)$$

where $\mathbf{T}_{P_{L_{t-1}}}^{-1}$ is the inverse of transformation matrix of pose $P_{L_{t-1}}$. If either translational norm $|\mathbf{X}_{L_t}|$ or rotational angle $\Delta\theta_{L_t}$ is bigger than preset thresholds, step 6 will be executed to generate a new local map by joining a new bunch of sub-maps. Otherwise step 6 will be skipped and the posterior information of local map is obtained as $\mathbf{M}_{L_t} = \bar{\mathbf{M}}_{L_t}$, $P_{L_t} = P_{L_{t-1}}$, $\mathbf{B}_{Sub(t)} = \mathbf{B}_{Sub(t-1)}$.

6. Local Map Decomposition and Reproduction: When the robot has moved too far away from the original pose of local map, a new node with a new sub-map should be initialized and local map needs to be updated in consideration of the balance of k-d tree. A new node will be initialized as $N_{Sub}(i_{sub}^{max} + 1) = P_t$, where i_{sub}^{max} is the sequence of most recently generated node and it must be the maximal sequence in $\mathbf{B}_{Sub(t)}$. And its corresponding sub-map is initialized as an empty map $\mathbf{M}_{Sub}(i_{sub}^{max} + 1) = Null$. Then the points stored in local map $\bar{\mathbf{M}}_{L_t}$ is going to be distributed to $\mathbf{M}_{Sub}(i_{sub}^{max} + 1)$ and the sub-maps in $\mathbf{B}_{Sub(t-1)}$. After that, a new brunch of sub-maps that are close to P_t are going to assemble the new local map \mathbf{M}_{L_t} , and $P_{L_t} = P_t$, as shown in Figure 3-20. Meanwhile, the sequences of sub-maps in the new brunch are saved to $\mathbf{B}_{Sub(t)}$. The process about how to find the sub-maps and their nodes that can be used for new local map production will be detailed in next part.

7. Loop Closure: After the local map has been updated, several features of the newest scan \mathbf{S}_t is going to be compared with previous scans to check if loop closure has been detected. Loop closing approach will be conducted if loop is detected. This work will be detailed in the next two parts.

8. Termination: If there is no more scan comes in, the process of SSPM-ICP will be terminated and global consistent map can be constructed by merging sub-maps.

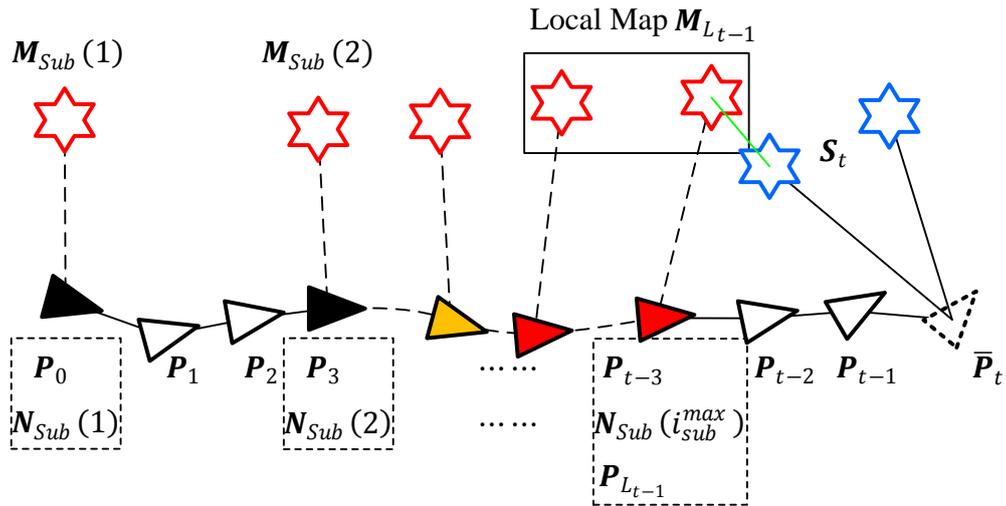


Figure 3-18 New scan arrives at time t

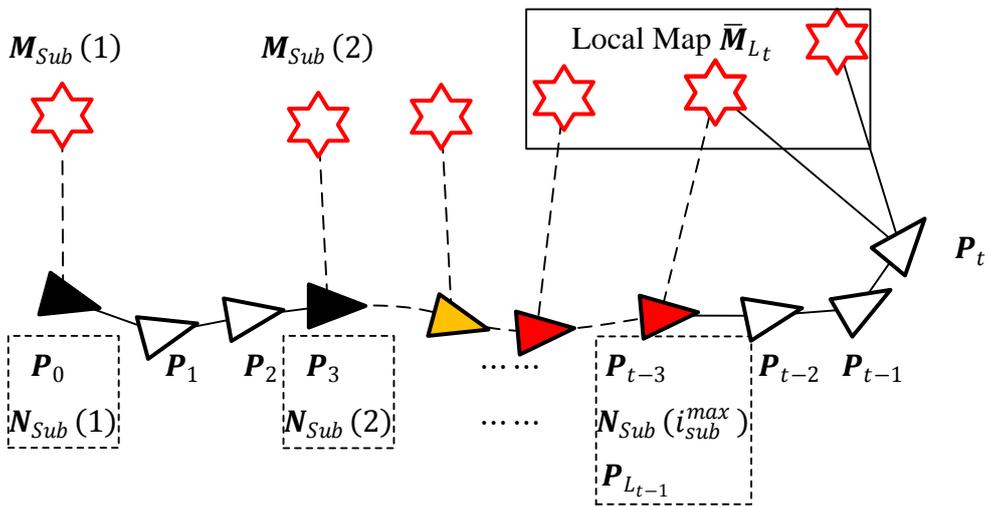


Figure 3-19 New scan is matched against local map

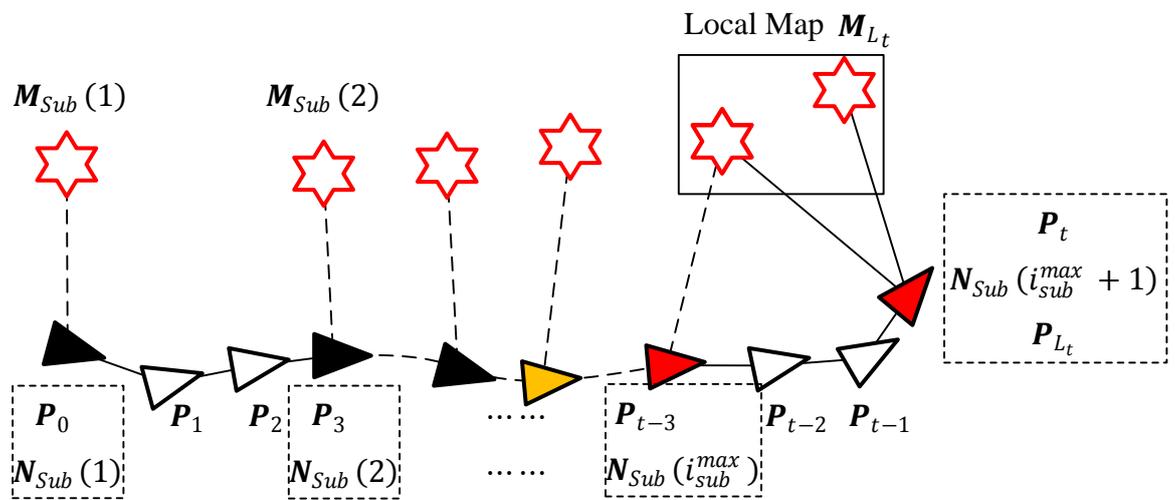


Figure 3-20 Local Map Reproduction

States of Nodes and New Local Map Generation

In order to include sufficient and reliable sub-maps that can be used for scan matching, all of the nodes of sub-maps are transferring within 3 different states based on their distance to robot pose, as shown in Figure 3-21. The 3 different states of nodes are listed as follows:

- 1) **Active Nodes:** The nodes that are locating within robot detecting range and the corresponding sub-maps are currently included in the updating local map.
- 2) **Waiting Nodes:** The nodes that were active nodes and now is out of detecting range of robot. Waiting nodes are within a certain distance from current robot pose.
- 3) **Sleeping Nodes:** The nodes that were waiting nodes and now is out of detecting range of robot. Sleeping nodes are usually very far from current robot pose except in the situation of loop closure.

For the nodes in different states, they have different abilities:

- 1) Active node has ability to “Cheer Up” a neighboring waiting node to become an active node. But it cannot “Cheer Up” any sleeping nodes.
- 2) Waiting node is able to “Wake Up” a neighboring sleeping node to become a waiting node. It can be transferred to be an active node when robot is approaching.
- 3) Sleeping node has no specific abilities. It can only be “Waked Up” by its neighboring waiting node and then becomes a waiting node. It can be directly “Cheered Up” to become an active node through loop closure approach.

This state transformation based approach is very useful. First, in non-loop existing area as shown in Figure 3-22 and Figure 3-23, it enable robot to include the near nodes that locate within its detecting range. Second, when robot has moved a large loop, the approach will prevent robot to include head node of loop directly since the current robot pose is corrupted with the accumulated errors. If the head node is included for scan matching directly, severe inconsistency will occur around this area. After loop closure work has been done, the accumulated error will be diminished and a consistent loop trajectory will be reached. After that, the head node of the loop will be directly “Cheered Up” to be an active node, as shown in Figure 3-24 and Figure 3-25.

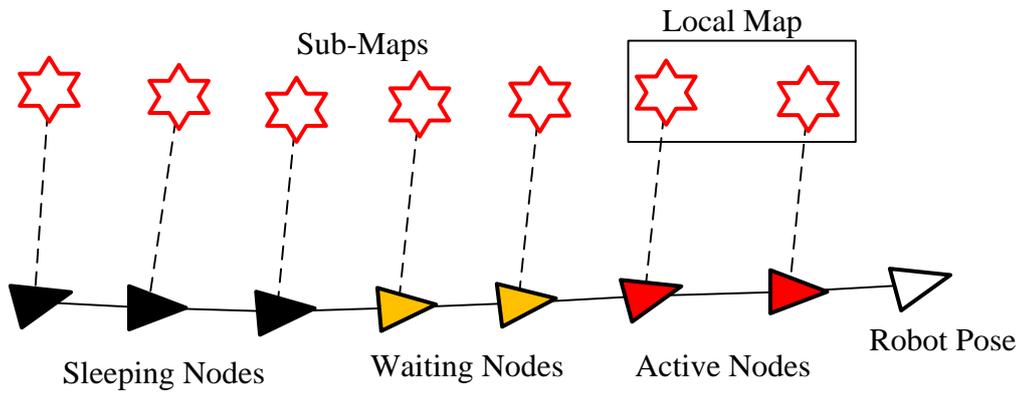


Figure 3-21 Nodes States with respect to robot pose.

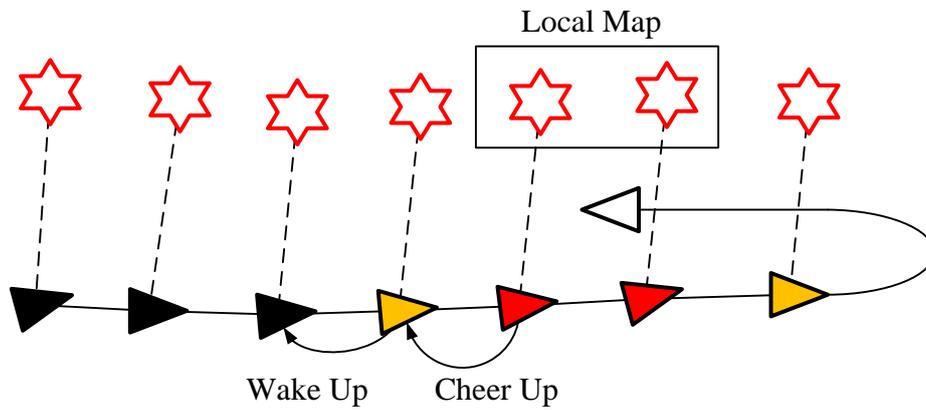


Figure 3-22 Wake Up function and Cheer Up function.

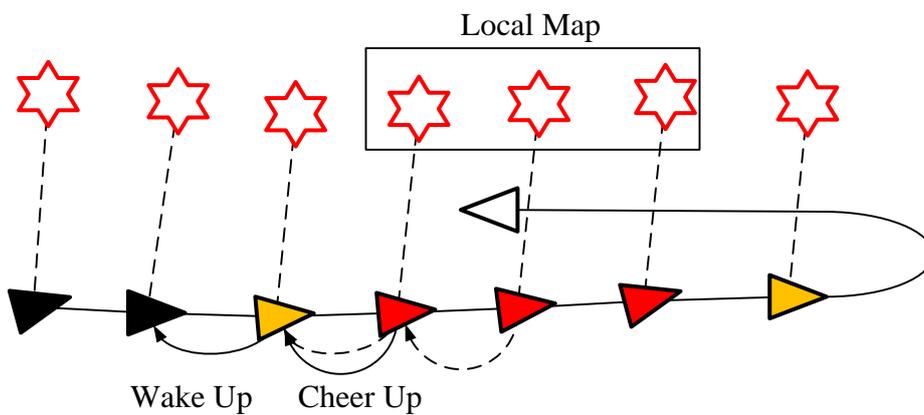


Figure 3-23 Waiting Node is cheered up by neighbor Active Node, Sleeping Nodes is waked by neighbor Waiting Node.

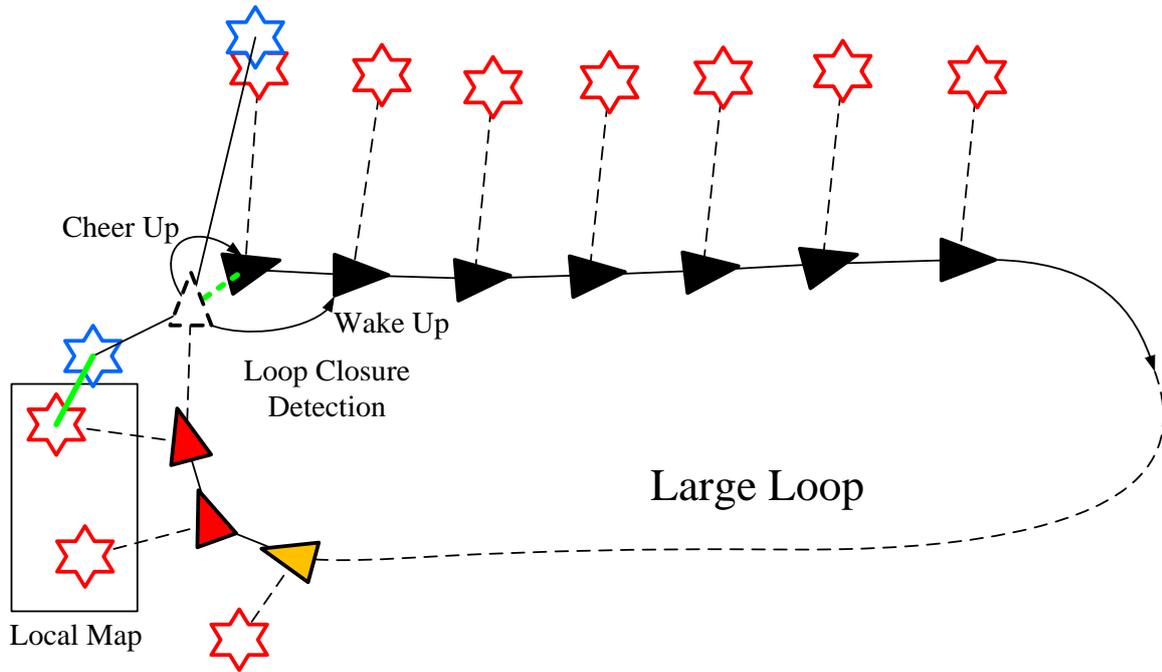


Figure 3-24 After robot has moved along a large loop, a loop head sleeping node has to be cheered up by using Loop Closure Approach.

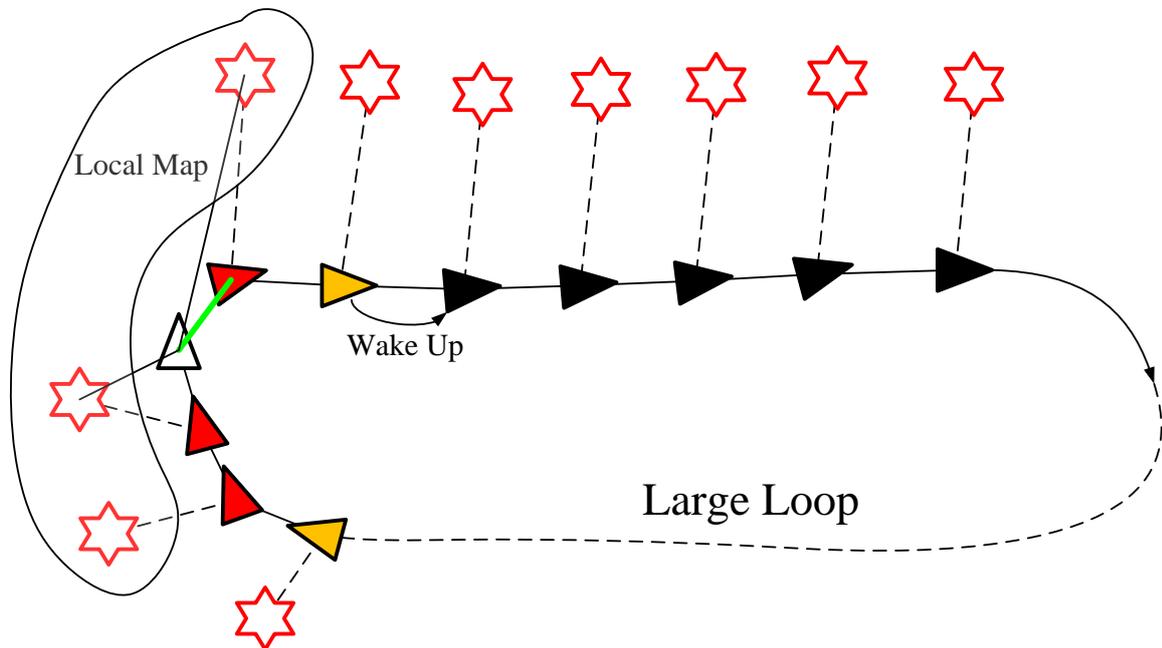


Figure 3-25 Loop closure conducted and the loop head sleeping node has been cheered up and loop second sleeping node has been waked up.

Trajectory Bending Based Loop Closure

Loop closure contains three tasks: loop detection, current pose updating, and loop closing which updates map and/or trajectory.

In this study, loop detection has not been investigated. Thus the task of loop detection has to be carried out manually. After that, scan matching is conducted to fulfill the second task, current pose updating. Finally, a node trajectory bending based loop closing method is conducted to merge the local maps and their trajectories, which refers to the third task.

The manually loop detection informs robot that a pair of scans, \mathbf{S}_{front} and \mathbf{S}_{rear} , are supposed to be taken from same area. According to this information, two corresponding poses \mathbf{P}_{front} and \mathbf{P}_{rear} as well as their nearest neighboring nodes $N_{Sub}(i)$ and $N_{Sub}(j)$ can be found out. By using PLICP to match \mathbf{S}_{rear} against \mathbf{S}_{front} , the relative transformation \mathbf{T} is estimated. Therefore an pose of scan \mathbf{S}_{rear} , which is free from error accumulation during the motion in loop, target pose \mathbf{P}_{rear}^{tar} is calculated by

$$\mathbf{P}_{rear}^{tar} = \mathbf{P}_{front} \oplus \mathbf{T}. \quad (3-16)$$

With the known pose, the target pose of $N_{Sub}(j)$, $N_{Sub}^{tar}(j)$ can be easily acquired based on the relative transformation between $N_{Sub}(j)$ and \mathbf{P}_{rear} . To eliminate the inconsistency between $N_{Sub}(j)$ and $N_{Sub}^{tar}(j)$, the process of loop closing is carried out among nodes whose index is among i and j .

With regard to the loop closing, researchers have proposed a plenty of sound approaches [68][69]. In order to obtain more robust and accurate result, most of them are based on the probabilistic approach which regards the uncertainties between estimated poses are anisotropic and inhomogeneous. Their common drawback is the heavy computational cost and difficulties in noise modelling. The noises in motion estimation are typically biased instead of widely adopt pure Gaussian noise model, which may lead to the unsatisfactory results [70].

The trajectory bending method applied in this study is inspired from a trajectory estimation approach that is proposed in [71]. The procedures are summarized as follows:

1. **Loop Extract and Target Node Acquisition:** The part of global trajectory that contains nodes $N_{Sub}(i)$ to $N_{Sub}(j)$ is selected in the global frame. And the desired pose of the ending node $N_{Sub}(j)$, $N_{Sub}^{tar}(j)$ is obtained based on the scan matching result, as shown in Figure 3-26.

2. **Duplicated Trajectory Alignment:** A copy of the trajectory is aligned to overlap its last node $N_{ub}^{copy}(j)$ to $N_{Sub}^{tar}(j)$, as shown in Figure 3-27.
3. **Merging Weights Calculation:** The weights between the corresponding nodes in two trajectories are calculated, as shown in Figure 3-28. In this study, the weights are calculated based on the travelled distances as follow:

$$w_k = d_{ik}/d_{ij}, \quad k=i,i+1\dots j, \quad (3-17)$$

where d_{ik} stands for the travelled distance from node i to node k .

4. **Trajectories Merge:** As shown in Figure 3-29, the new nodes are calculated based on the weighted merge of original node and its realigned copy as:

$$N_{Sub}^{Close}(k) = (1 - w_k)N_{Sub}(k) + w_k N_{Sub}^{copy}(k), \quad k=i,i+1\dots j, \quad (3-18)$$

5. **Sub-Maps Update:** Based on the updated nodes inside the loop, the sub-maps from $M_{Sub}(i)$ to $M_{Sub}(j)$ are going to be transformed and updated.

This approach can efficiently eliminate the inconsistency between $N_{Sub}(j)$ and $N_{Sub}^{tar}(j)$ by being split and distributed into the relative transformations inside the loop weighted on the travelled distances.

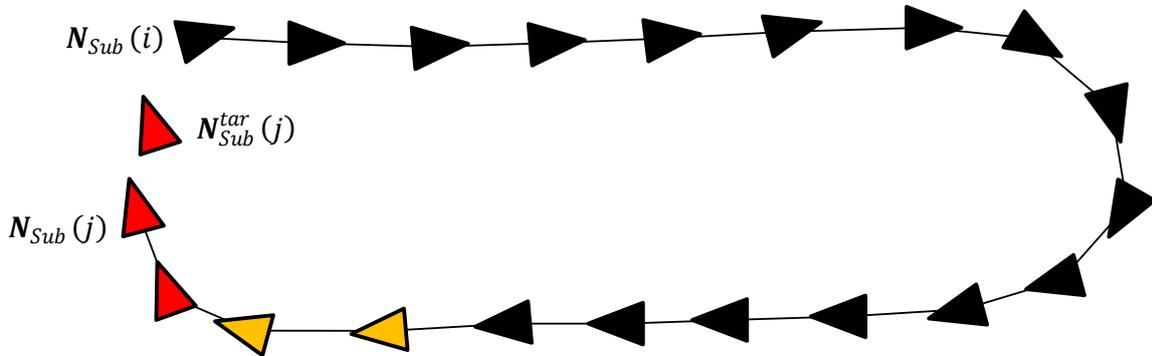


Figure 3-26 Origin trajectory and desired ending pose

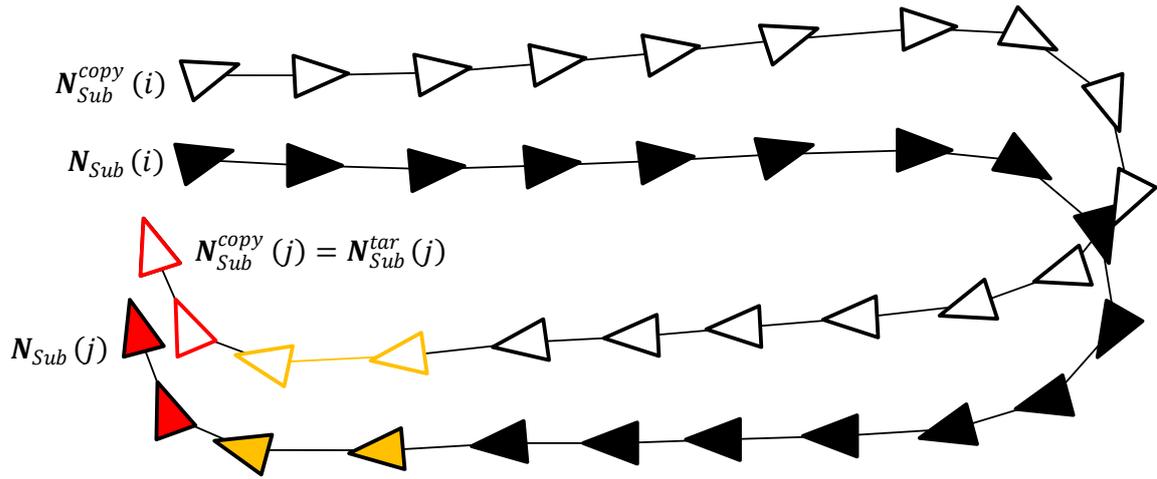


Figure 3-27 Duplicated trajectory aligned

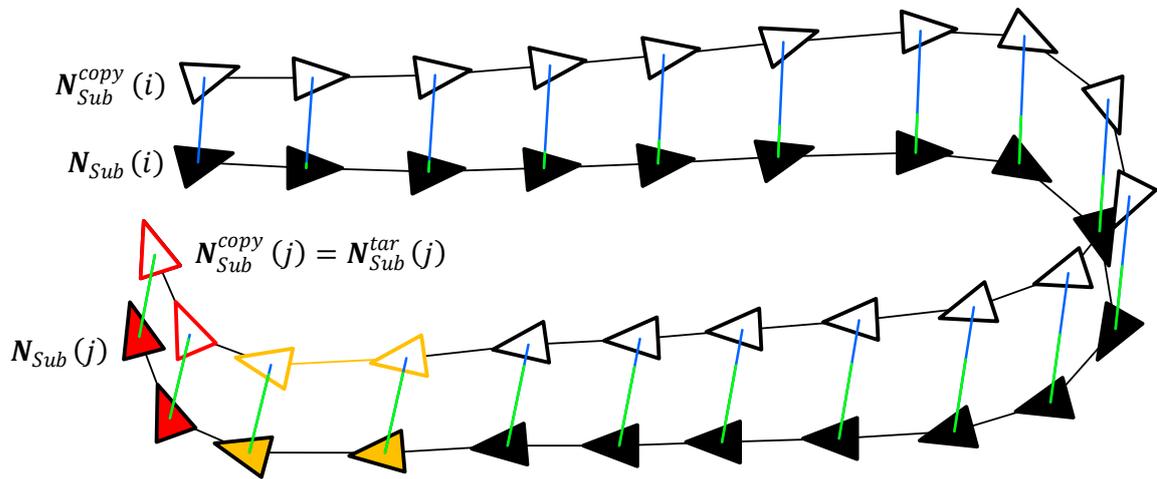


Figure 3-28 Weights Calculation

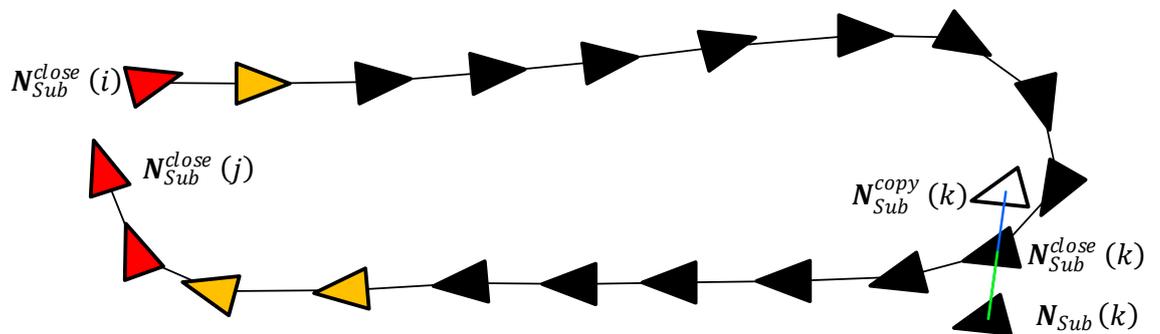


Figure 3-29 Merged trajectory

3.5 Experiments and Discussions

In this section, experiments are conducted to verify the efficiencies of the proposed algorithms. Discussions and analyses are given on the experiment results.

3.5.1 Verification of RIA

The experimental datasets of “Cluttered Office” and “Long Corridor” that have been used in ICN verification are adopted to show the efficiency of the RIA algorithm.

Cluttered Office

Figure 3-30 shows the scan matching result obtained by PLICP after RIA gives coarse estimation on the transformation between successive two scans. Since no mismatching has been found in this scan matching result, the final transformation estimated by PLICP is regarded as the ground truth and errors of RIA can be obtained. The coarse estimation on translations along X_R and Y_R axes are plotted in Figure 3-31 and Figure 3-32, respectively. RIA efficiently eliminates the majority of the translational error especially along the direction of Y_R . The rotation estimation is very robust and accurate, as shown in Figure 3-33.

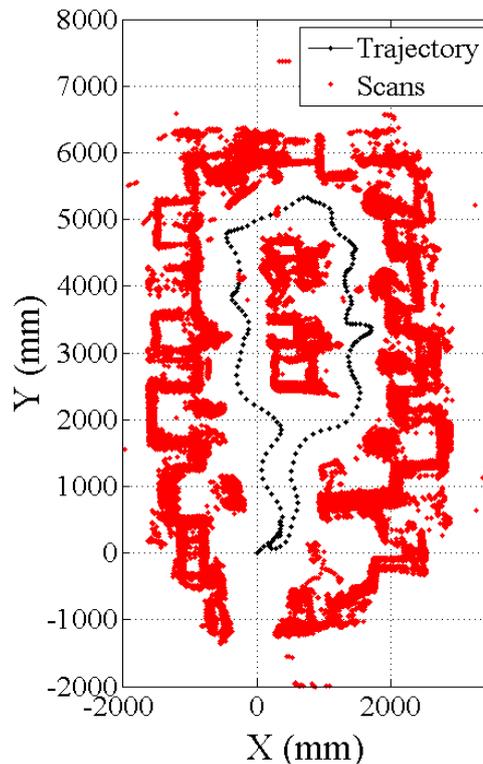


Figure 3-30 Pairwise scan matching obtained by RIA and PLICP, Cluttered Office dataset

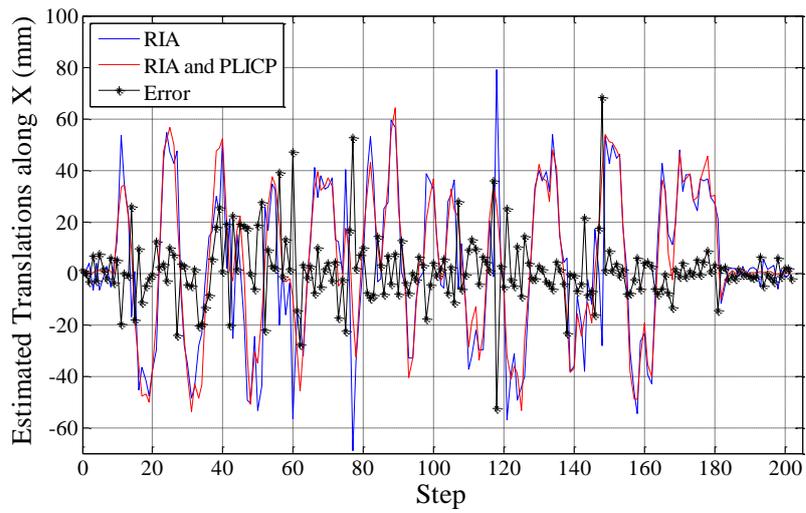


Figure 3-31 Comparison of estimated translation along X_R , Cluttered Office dataset

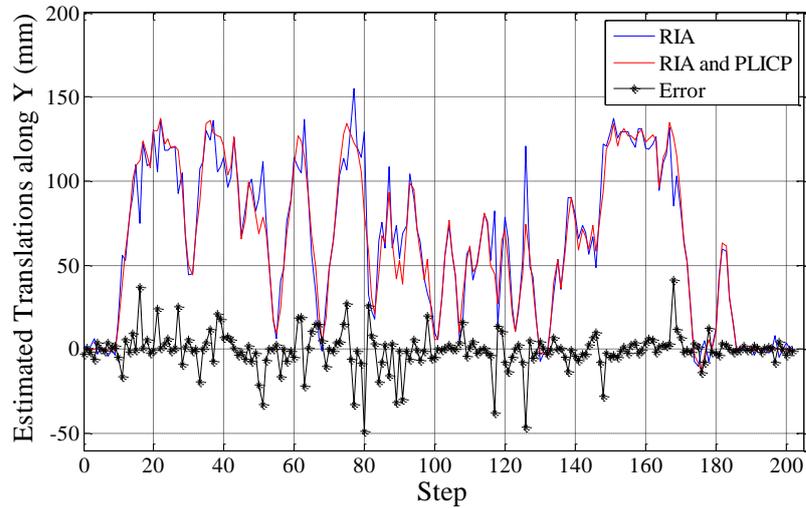


Figure 3-32 Comparison of estimated translation along Y_R , Cluttered Office dataset

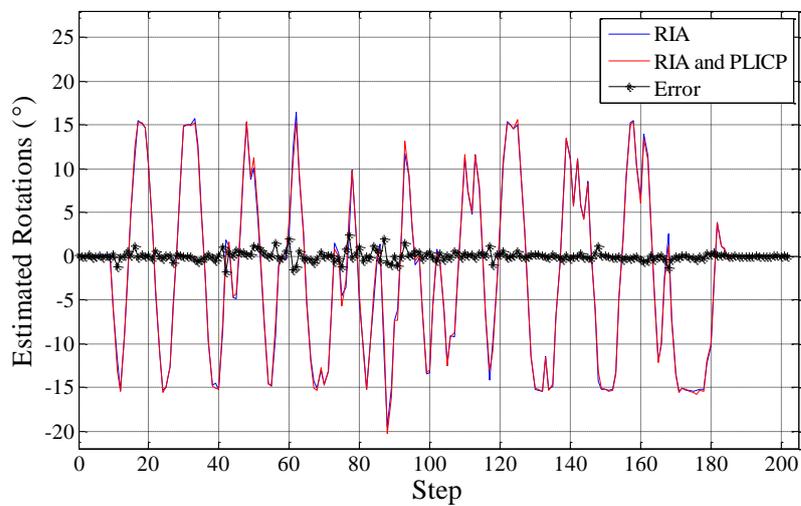


Figure 3-33 Comparison of estimated rotation, Cluttered Office dataset

Long Corridor

Similar results can be found by applying RIA and PLICP on the dataset of Long Corridor. Robust pairwise scan matching result is obtained, as shown in Figure 3-34. Again, the coarse estimations on transformation provided by RIA are compared with final PLICP scan matching results, as shown in Figure 3-35, Figure 3-36 and Figure 3-37.

In conclusion, RIA can accurately pre-align two scans even with existence of large rotational angle between two scans. Though the translation estimation of RIA is not as impressive as its rotation estimation, it is able to eliminate the majority of translational error. Therefore RIA is a good pre-processor of fine scan matching method.

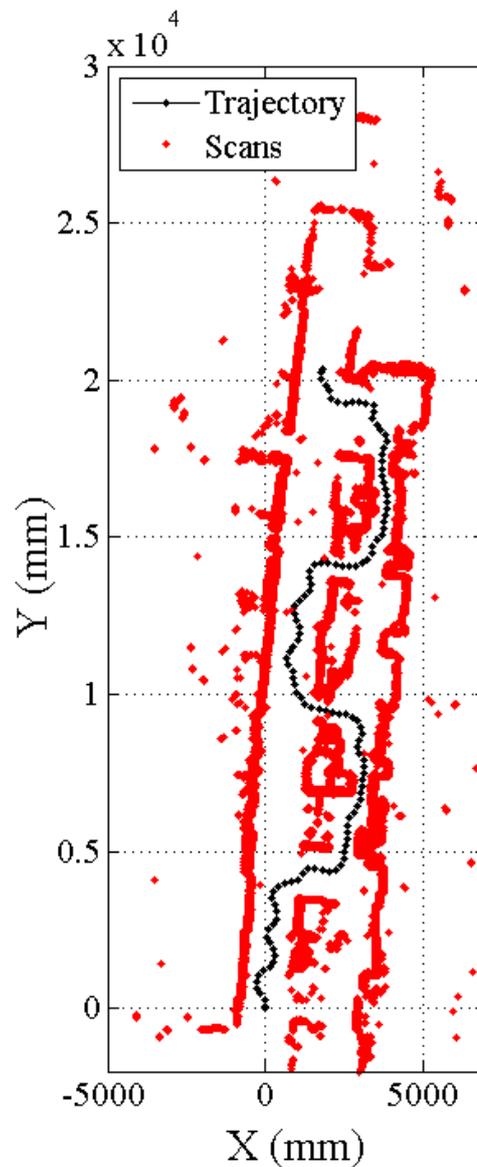


Figure 3-34 Pairwise scan matching obtained by RIA and PLICP, Long Corridor dataset

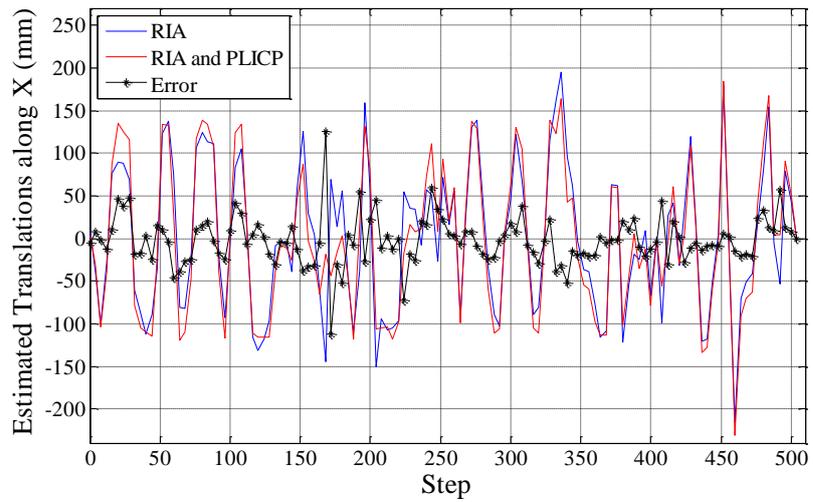


Figure 3-35 Comparison of estimated translation along X_R , Long Corridor dataset

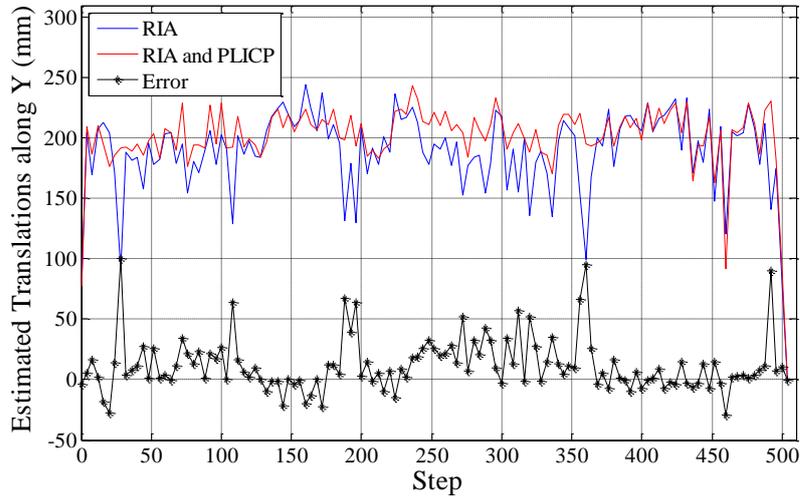


Figure 3-36 Comparison of estimated translation along Y_R , Long Corridor dataset

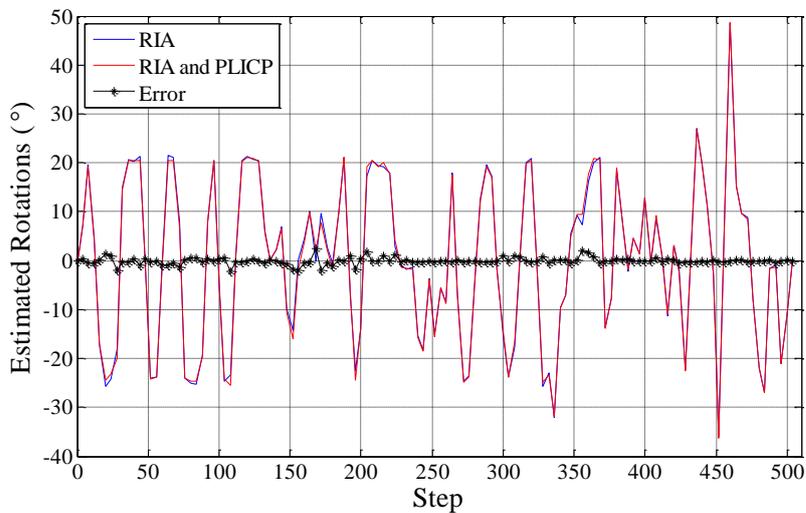


Figure 3-37 Comparison of estimated rotation, Long Corridor dataset

3.5.2 Incremental Scan Matching in Small Environment

Environmental environment



(a) Office room



(b) Corridor



(c) Meeting room

Figure 3-38 Office room, corridor and meeting room for experiment

Experimental result

The first experiment is conducted in a normal indoor which is mainly consisted of a small cluttered office, long corridor and a meeting office where is filled with chairs and desks, as shown in Figure 3-38. The SSPM-ICP with IAFM based association is applied to the scans obtained by the LRF. The scan matching result is consisted of the global sparse points map as well as the trajectory of robot, as shown in Figure 3-39. Although three loops can be found in this environment, loop closing method has not been applied since these loops are small and the errors of SSPM-ICP accumulated within these loops have not brought severe inconsistency.

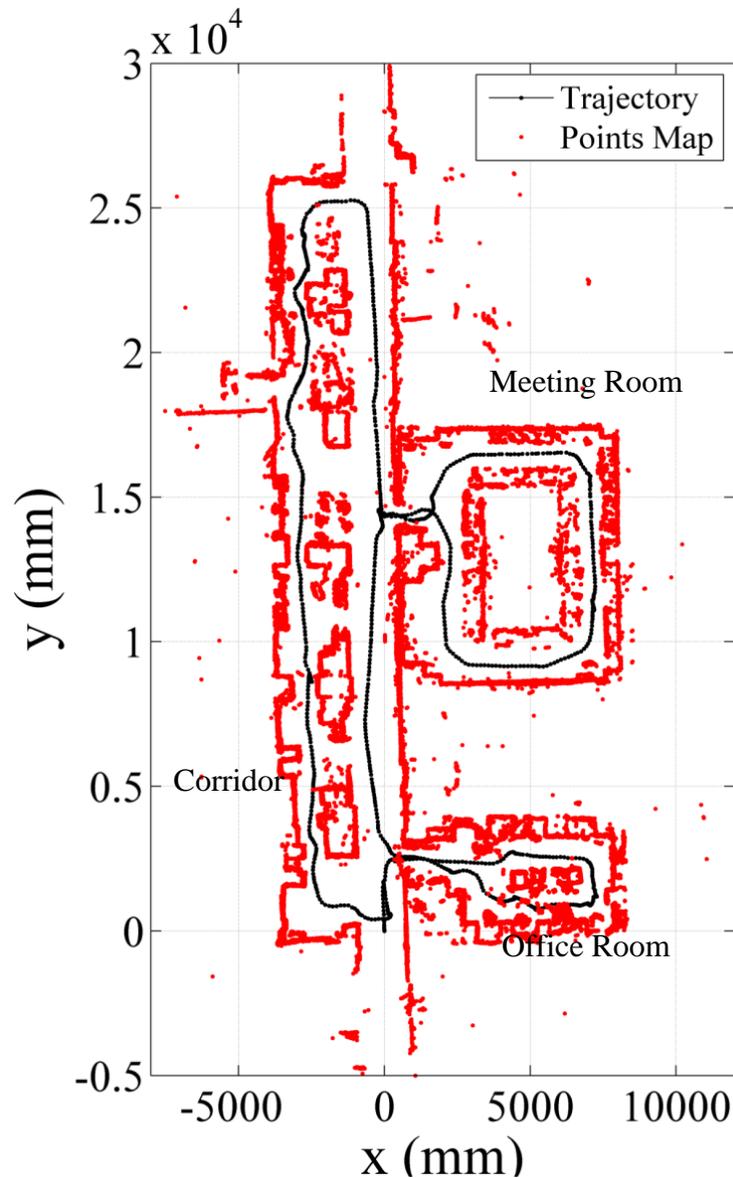


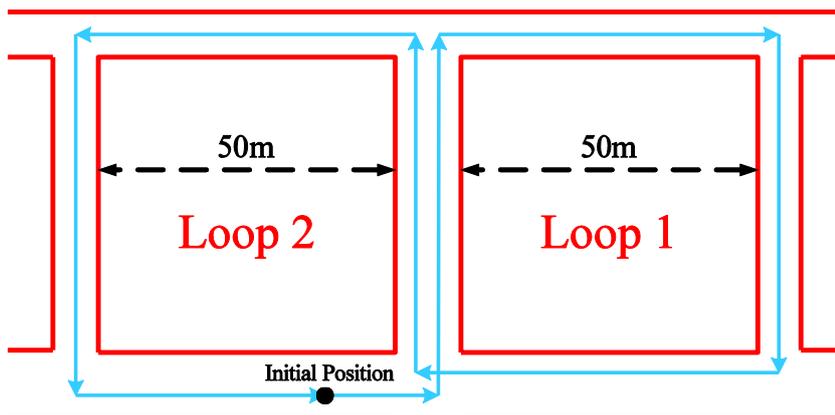
Figure 3-39 Experimental result fulfilled by SSPM-ICP without loop closing

3.5.3 Incremental Scan Matching in Large Loops

Environment and robot motion



(a) Satellite image (Google Map) of building that captures dual loops



(b) Sketch of experimental environment and robot trajectory



(c) Inside scene of the building



(d) Long Corridor

Figure 3-40 Experimental environment consists of dual large loops

This experiment has been conducted inside buildings where there are two square loops, as shown in Figure 3-40 (a). The side length of each square is about 50 meters. The robot is controlled to move starting from the initial position that is marked as the black dot in the bottom right of second loop. It firstly moves around the first loop in clockwise direction and then explores the second loop in anticlockwise direction. Finally it returns to the initial position and stops. A sketch trajectory of robot is plotted in Figure 3-40 (b). Inside scenes of the building have been presented in Figure 3-40 (c) and (d).

Comparison between the IAFM and Euclidean distance

To verify the proposed association metric, the experimental scans obtained from the first loop are picked out and be matched by applying different distance metric in association process. The scan matching results are obtained without introducing odometer data. SPM-ICP is carried out and two scan matching results that are based on different distance metric are obtained, as shown in Figure 3-41 and Figure 3-42. Comparing with the traditional approach, the scan matching result is sharply improved by employing the proposed distance metric, which can be easily found out from the enlarged drawings in Figure 3-41 and Figure 3-42.

Result of Loop Closing fulfilled by SSPM-ICP

After introducing odometer data with dead-reckoning approach, SSPM-ICP is applied on whole scans to obtain consistent global map and robot trajectory.

The scan matching result of first loop before closing can be found in Figure 3-44. Although the inconsistent is small in consideration of the detecting range of LRF and the size of the environment, it is non-ignorable for it prevents robot to obtain a consistent global map. The closed map and trajectory can be found in Figure 3-45. The inconsistency is efficiently eliminated after applying proposed loop closing method.

The scan data obtained from the slant placed LRF is registered based on the closed loop trajectory, as shown in Figure 3-46. This points cloud accurately represents this area and no obvious inconsistency can be found. In order to compress the huge storage cost of points cloud, OctoMap [72] has been applied on the obtained points cloud. Though some unexpected data missing occurs, OctoMap sufficiently compresses the storage cost and maintained the main information, as shown in Figure 3-47 and Figure 3-48.

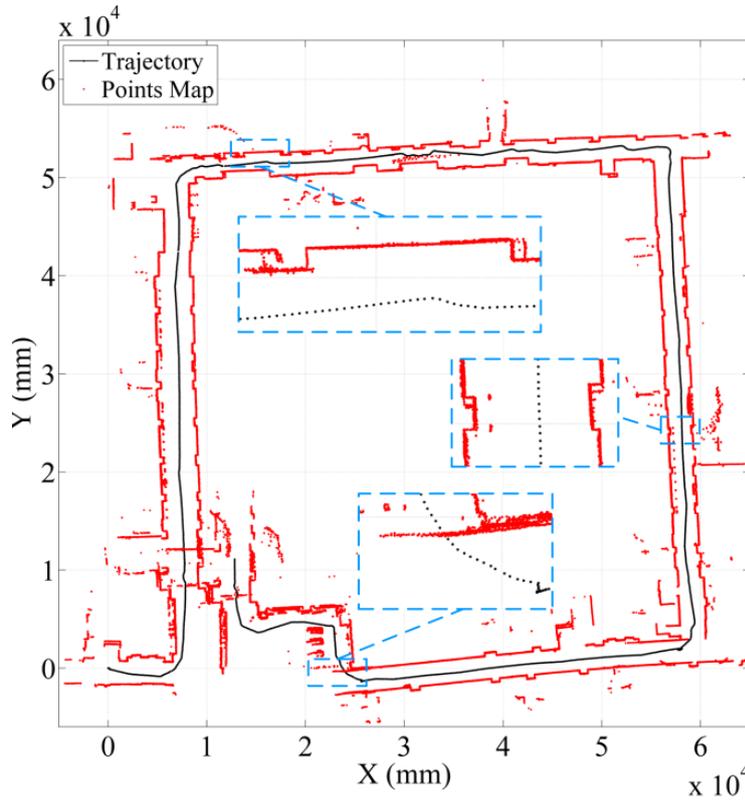


Figure 3-41 Result of SPM-ICP using traditional Euclidean distance without odometer

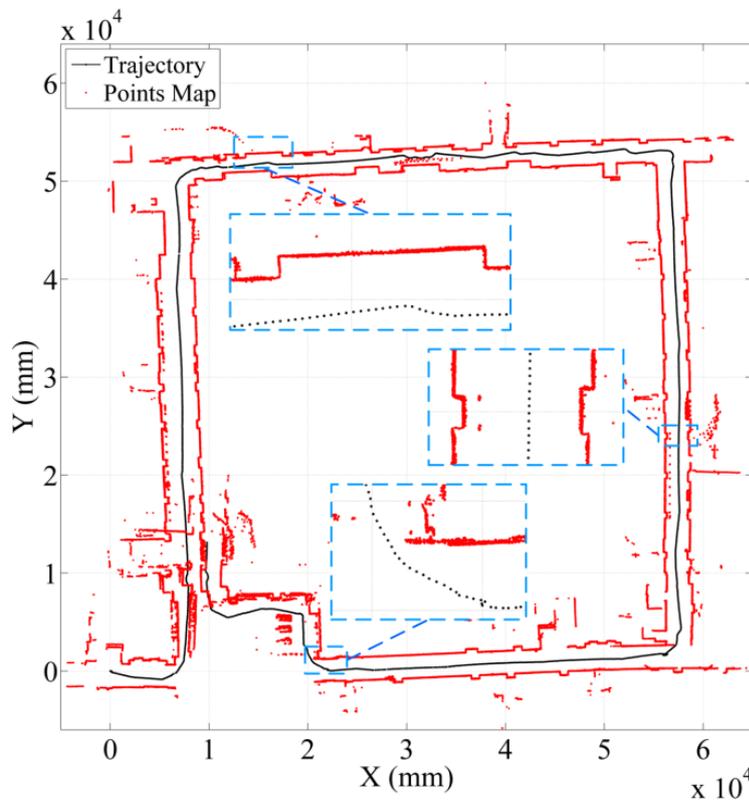


Figure 3-42 Result of SPM-ICP using proposed IAFM without odometer

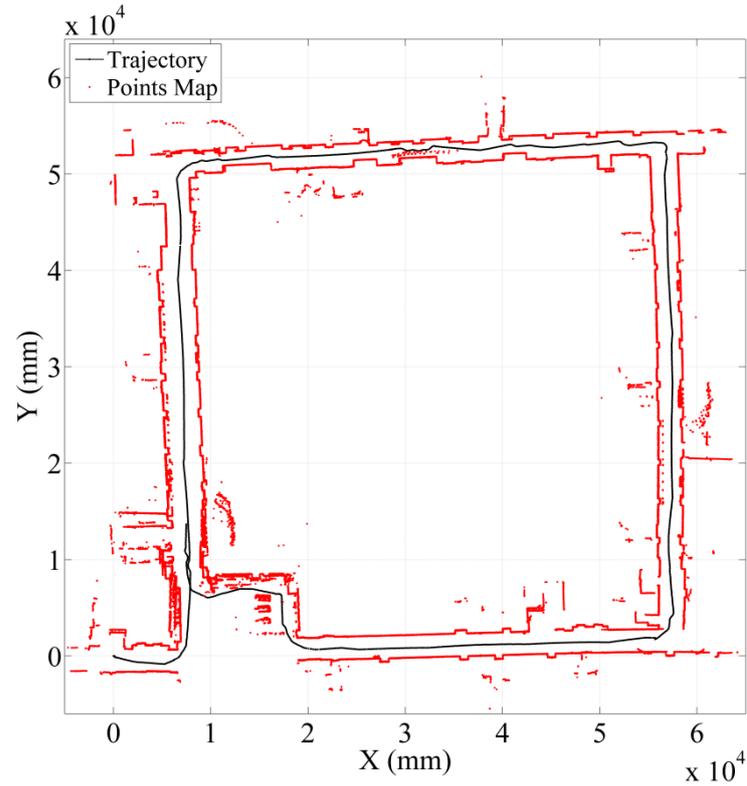


Figure 3-43 Result of SPM-ICP with odometer available

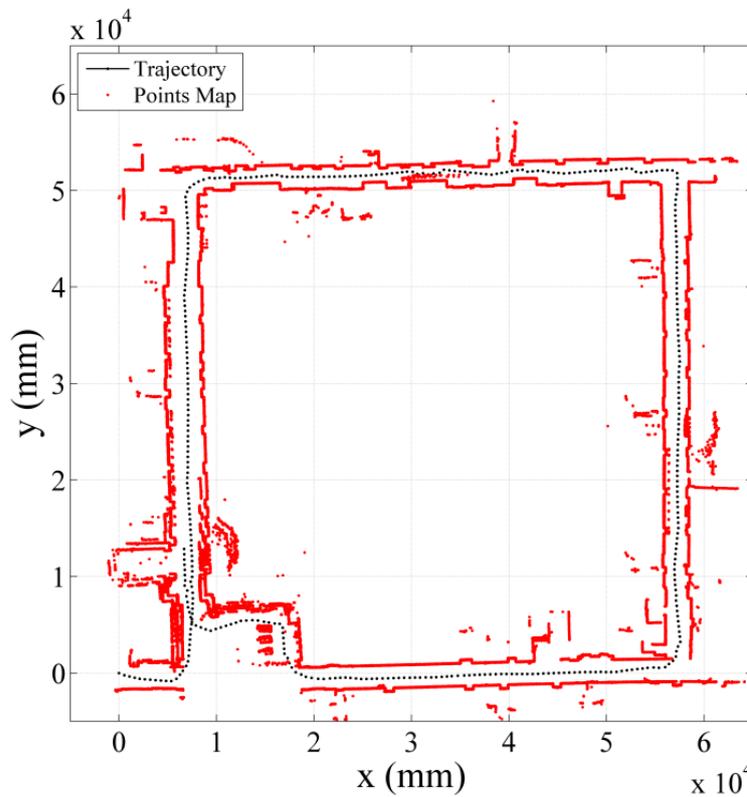


Figure 3-44 Result of SSPM-ICP with odometer available before first loop closing

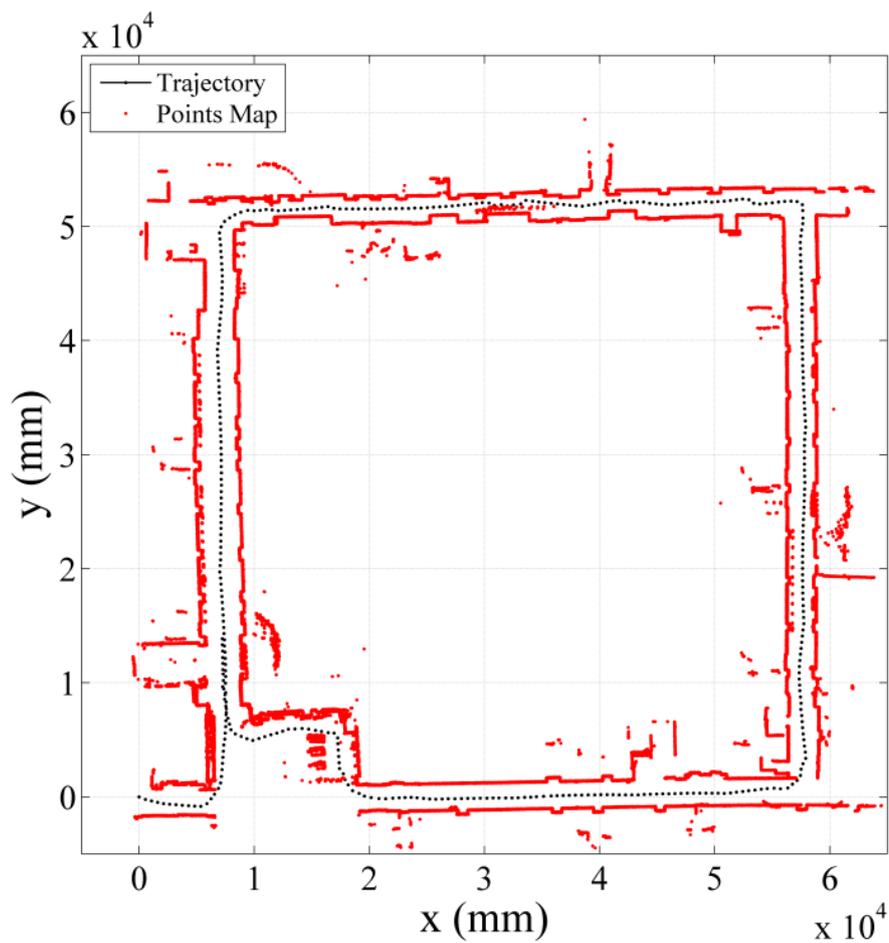


Figure 3-45 Result of SSPM-ICP with odometer available after first loop closing

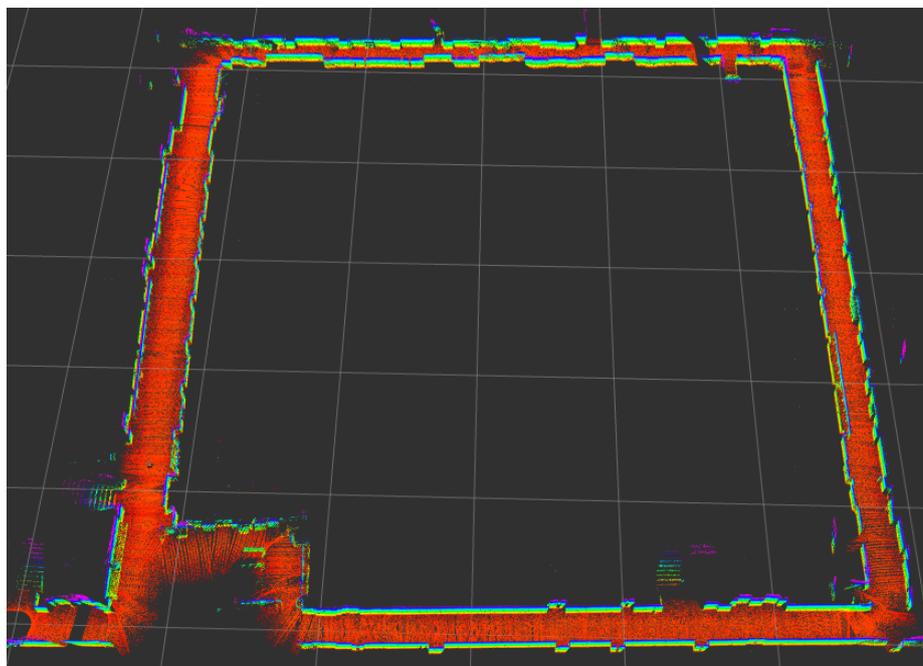


Figure 3-46 Points cloud of the first loop

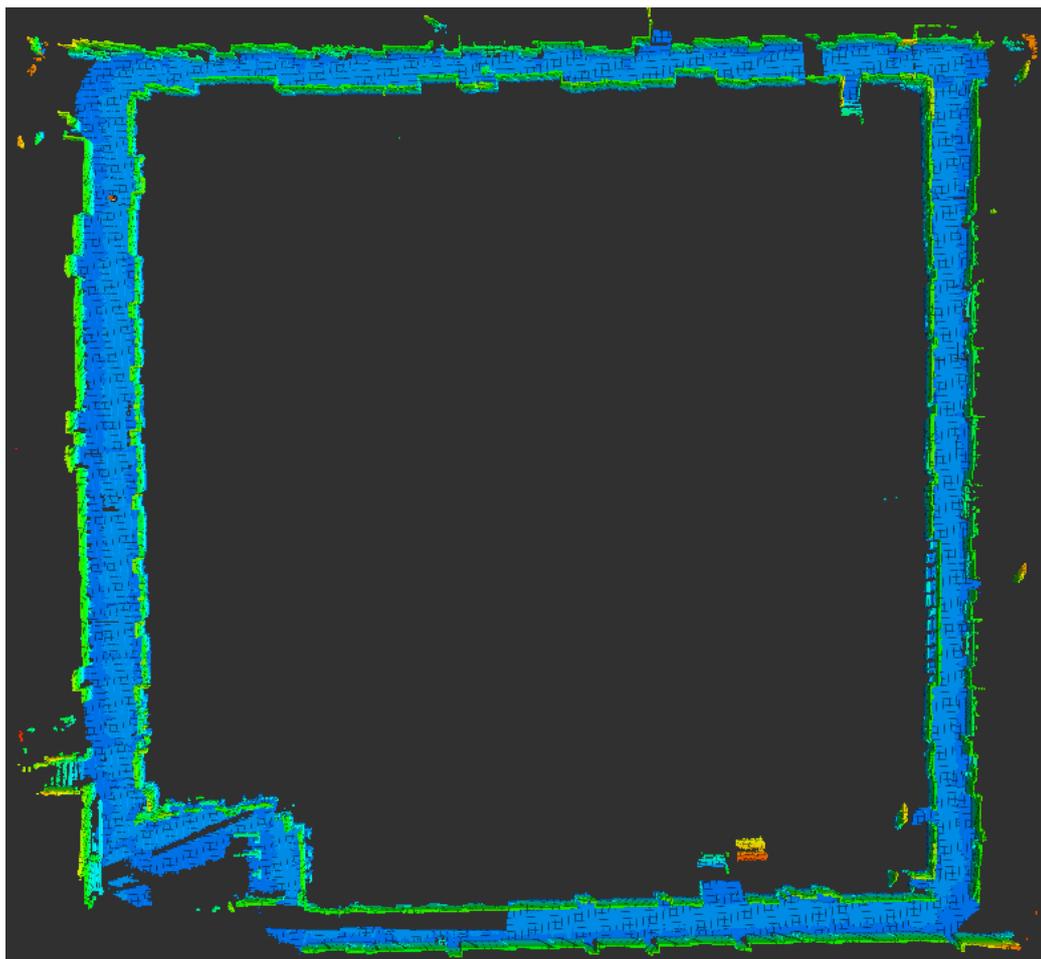


Figure 3-47 Top view of the constructed Octomap

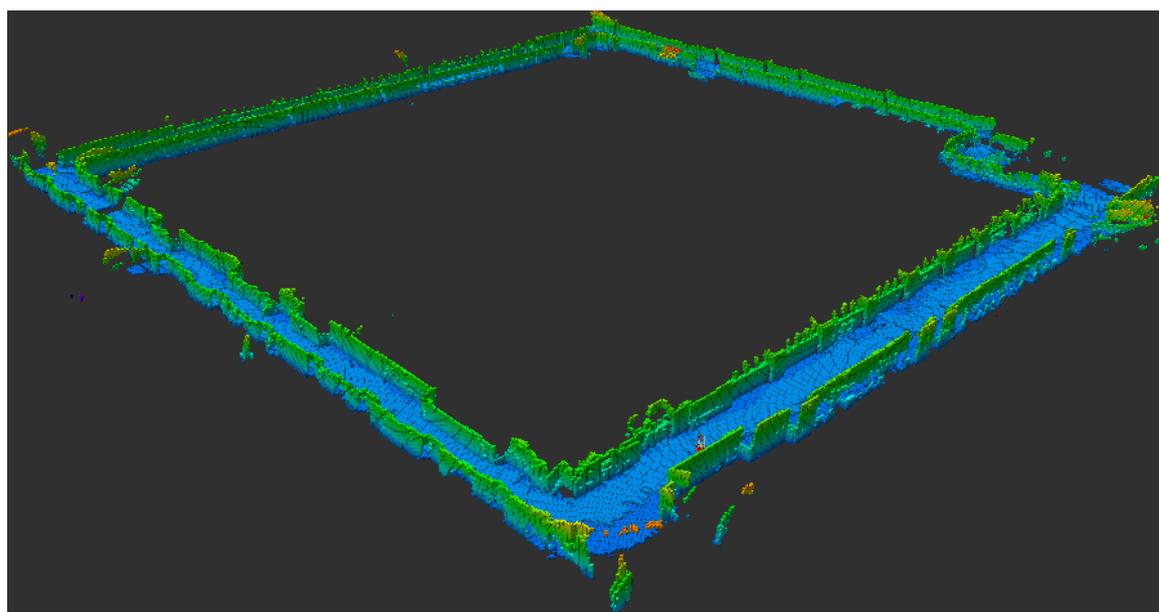


Figure 3-48 View of the constructed Octomap

The result before second loop closing can be found in Figure 3-49. Similar small inconsistency has appeared and it is substantially eliminated by loop closing method, as shown in Figure 3-50. The final map fits the structure of the building well and there is no severe inconsistency can be found. In consideration of the limited range of LRF and large scale of the environment, this scan matching result is fairly good.

Comparison between SPM-ICP and SSPM-ICP

In comparison with first loop's scan matching result of SSPM-ICP that is plotted in Figure 3-44, SPM-ICP has not shown superior performance after introducing odometer data, as shown in Figure 3-43. The inconsistency of the map fulfilled by SPM-ICP is out of the tolerance, which prevents SPM-ICP from making a consistent dual loop map. Figure 3-51 shows the comparison of computational time at each step of SPM-ICP and SSPM-ICP during the whole exploration. Since SPM-ICP is stopped after first loop for its inconsistency is oversize.

As analyzed before, SSPM-ICP is much faster than SPM-ICP because of less reference points in scan matching and more balancing k-d tree. The mean time cost of SSPM-ICP is around 60ms. Two steps that conduct loop closing cost much more time than normal steps. These huge time costs are mainly blamed on the sub-maps updating, which has to deal with huge number of points.

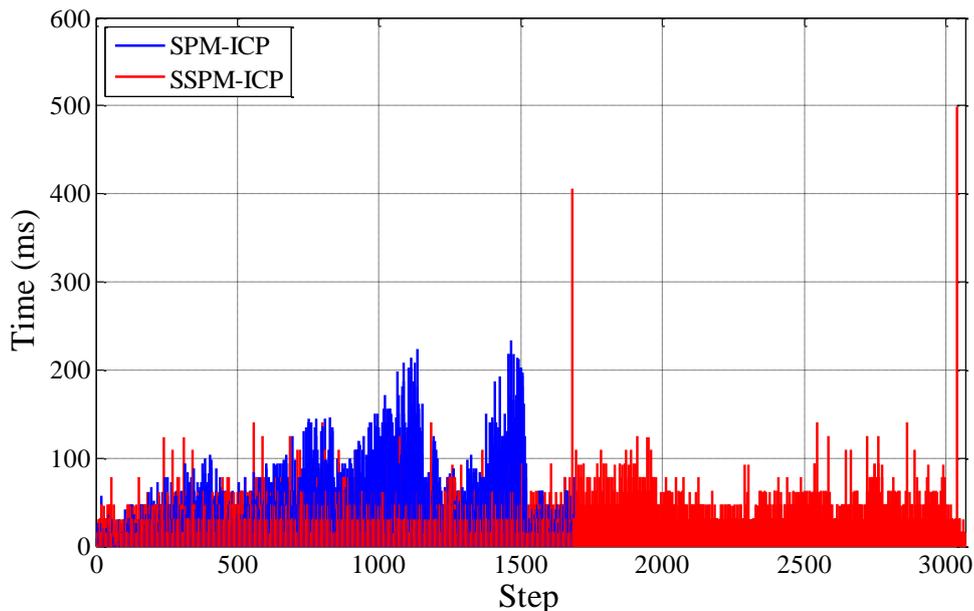


Figure 3-51 Computational cost of SPM-ICP and SSPM-ICP

3.5.4 Statistical Tests and Comparisons of Distance Metrics

In order to statistically verify the validation of the proposed IAFM, simulation that is similar to what has been utilized in last chapter is conducted by using the scans obtained from previous experiments. The copies of scans are transformed and then be matched to original one by using scan matching. The transformations are detailed in Table 3-2 and Figure 3-52. Scan matching that adopts traditional Euclidean distance and IAFM in association process have been conducted and their results are recorded and compared, respectively.

Table 3-2 Detail transformations of 25 tests

	x (mm)	y (mm)	θ ($^{\circ}$)		x (mm)	y (mm)	θ ($^{\circ}$)
Test 1	200	0	-20	Test 14	0	200	10
Test 2	200	0	-10	Test 15	0	200	20
Test 3	200	0	0	Test 16	-141	141	-20
Test 4	200	0	10	Test 17	-141	141	-10
Test 5	200	0	20	Test 18	-141	141	0
Test 6	141	141	-20	Test 19	-141	141	10
Test 7	141	141	-10	Test 20	-141	141	20
Test 8	141	141	0	Test 21	-200	0	-20
Test 9	141	141	10	Test 22	-200	0	-10
Test 10	141	141	20	Test 23	-200	0	0
Test 11	0	200	-20	Test 24	-200	0	10
Test 12	0	200	-10	Test 25	-200	0	20
Test 13	0	200	0				

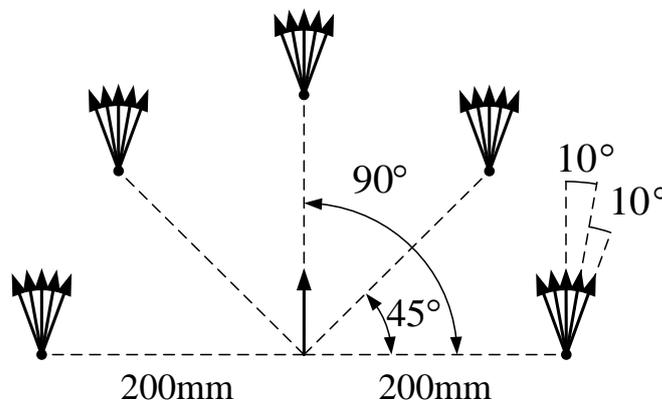


Figure 3-52 Translations and rotations of the tests

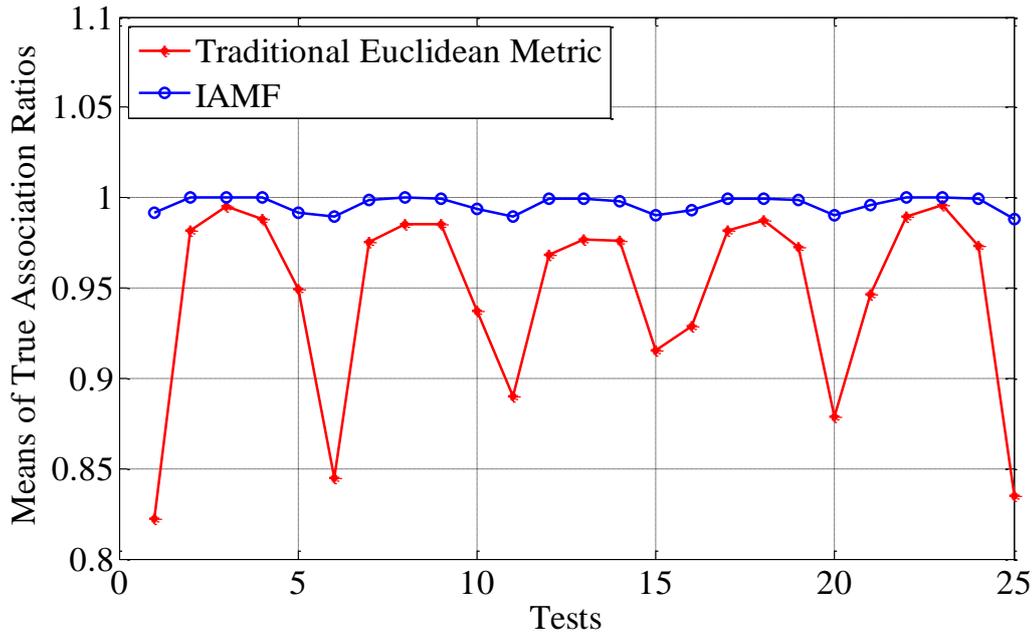


Figure 3-53 Means of true association ratios obtained by different distance metrics

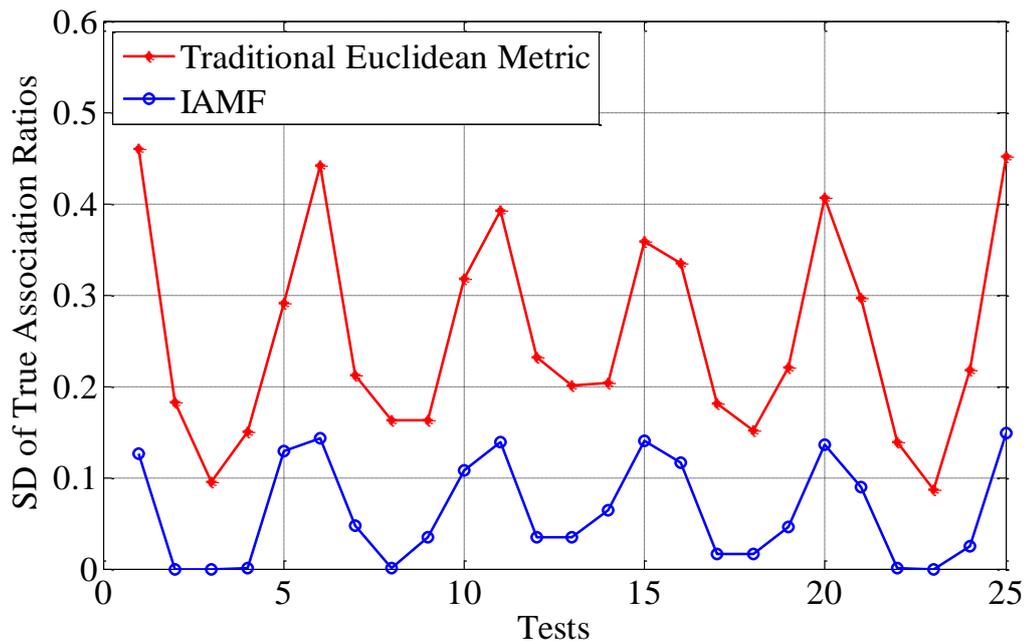


Figure 3-54 SD of true association ratios obtained by different distance metrics

Since the scan matching is applied on scans that originally same, by checking the point sequences of the correspondences, the true association ratio can be obtained after scan matching. If the transformation error is completely compensated by the scan matching, the true association ratio should be equal to 1. Otherwise, the true association ratio should be smaller than 1 and is decreasing with the increasing error of estimated transformation.

The means of true association ratio obtained by using IAFM based scan matching and traditional Euclidean Metric based scan matching have been plotted in Figure 3-53. It shows that IAMF is much more robust to the rotations and gives more accurate association results. Standard Deviation (SD) of true association ratios are presented in Figure 3-54. IAFM captures lower SD than traditional Euclidean Metric, which means it performs more stable.

3.6 Conclusion

In this chapter, several nearest neighbor search methods have been introduced at the very beginning, which is introduced to speed up scan matching methods. Then, the superior property of incident angle in point to point association has been detailed and a scan matching preprocessor RIA has been proposed. To improve the data association of ICP variants in under-constraint environment, an incident angle fused metric is proposed. A sub-maps joining version of SPM-ICP, SSPM-ICP has been presented to conduct scan matching in large loops with the manner of incremental scan matching. And it does not without require probabilistic processing. Based on the maps joining framework and a straightforward, efficient trajectory bending method, loop closure task has been fulfilled. Experiments have been conducted in various real indoor environments. The satisfactory results and corresponding analysis have verified the validities of the proposed methods.

Chapter 4 Line-segment based EKF SLAM with Slope Detection

4.1 Introduction

Comparing with a 3D map, a 2D map only carries planar information about the environment. Therefore, the memory, computational costs of manufacturing, updating, and consultation are dramatically reduced. It is mainly applied to represent indoor environments, especially neat interior corridors. However, in some buildings, even in the same floor, there are often stairs because of the height variation between different areas. Some of them are accompanied by slopes or elevators for the convenience of using wheel chairs. It is extremely important to detect these types of slopes and edges so that the robot can avoid falling and make the decision whether it is possible to move along the slope. By using RGBD cameras or some other 3D sensors, 3D SLAM technology [73][74] is capable of mapping and updating these features with various increased costs. The cost not only increases in mapping process but also rises in localization since 3D feature has more parameters. To overcome it, an alternative option is to enrich the 2D map by introducing key information of the essential 3D features [75]. Using ultrasonic sensors, a slope detection system has been proposed in [76]. However, it only estimates the gradient of slope and the robot is still under the risk of falling. 2D maps that record all crucial information of slope and edge are reported in [77] and [78], by using stereo camera and 3D LRF, respectively. Both of them extract the features from 3D data and then transfer to the 2D map. The limitation is that they only generate the local map instead of integrating it into the SLAM algorithm to get global map which is necessary for global path planning during autonomous navigation.

In this chapter, an ICP assisted line-segment based EKF-SLAM framework to construct 2D maps that record essential information on slopes and edges has been presented. The proposed algorithm uses an odometer and two fixed LRFs as the main sensors. The

horizontally placed LRF is used for the traditional planar scanning that is parallel to the ground, whereas the vertically fixed LRF is used to monitor the ground in front of the robot to detect slopes and edges. There are three main contributions that result from this study. Firstly, a 2D ICP algorithm is applied to evaluate the dead reckoning result and make the decision whether to step further to compensate error. Secondly, the slopes and edges that have been detected in the vertical scans are modeled into 2D line-segments with the angular parameter estimated by local Orthogonal Assumption. Thirdly, a method to integrate and update the extracted features (the line-segments that represent the slope and edge) into the EKF-SLAM framework is proposed.

4.2 System Architecture

4.2.1 Robot Platform and Sensors

The mobile robot platform Plat-F1 is employed in this research, as shown in Figure 4-1. Comparing with the robot system plotted in Figure 2-14, the modification is the installation of LRF2. Instead of being fixed on the aluminum frame with its scanning plane orthogonal to the forward direction, LRF2 is vertically installed with its scanning plane parallel to the forward direction. This design makes the robot system be able to keep scanning on the front ground.

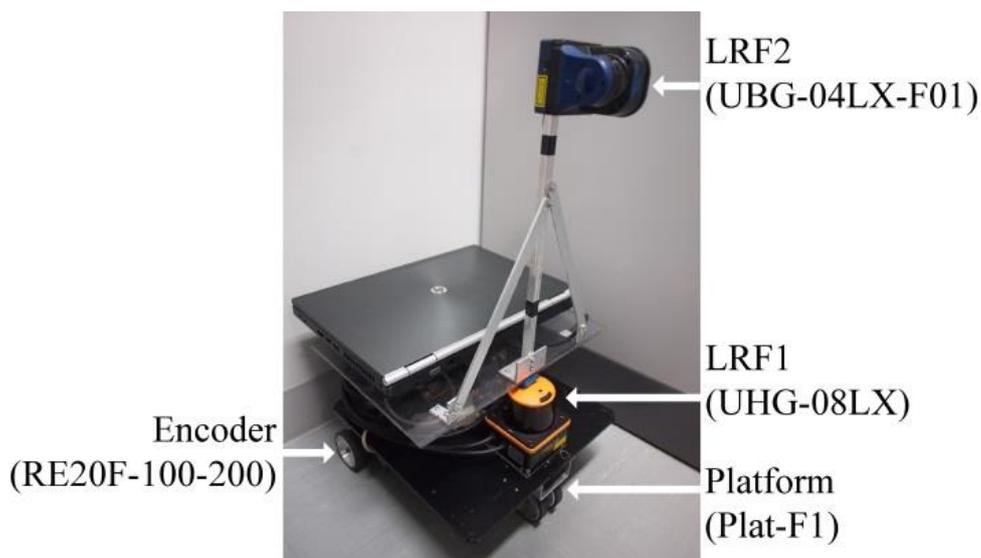


Figure 4-1 The mobile robot and sensors used for the experiments.

4.2.2 System Architecture

The whole system architecture can roughly be divided into 3 procedures with regard to the data read from 3 sensors, as shown in Figure 4-2.

The dead reckoning process is executed to estimate the transformation of the robot during the time interval between time $t - 1$ and time t . After transformation by utilizing the estimation result, a copy of the horizontal scan $S_{h(t)}$ is applied to match the former horizontal scan $S_{h(t-1)}$ to evaluate the quality of the dead reckoning result. Based on this evaluation, the decision is made whether ICP must be executed to compensate the odometer error. Finally, the EKF prediction process is conducted by utilizing the original or compensated dead reckoning result.

Meanwhile, the line-segments are extracted from the horizontal scan and then imported into the EKF-SLAM to execute the feature association and correction. Besides, the orthogonal angle is estimated based on the extracted line-segments if any slope or edge has been detected in the vertical scan process.

Other than the horizontal procedure, the orthogonal estimation is conducted at every step of the vertical scan $S_{v(t)}$ to make sure the central axis of the scan plane is parallel to the ground. The vertical feature extraction is mainly utilized in the line-segments scanned from the slope and the points that are next to the edge. With the angular parameter provided by horizontal orthogonal estimation, the vital information of vertical features can be estimated and then transported into the EKF-SLAM for feature association and correction.

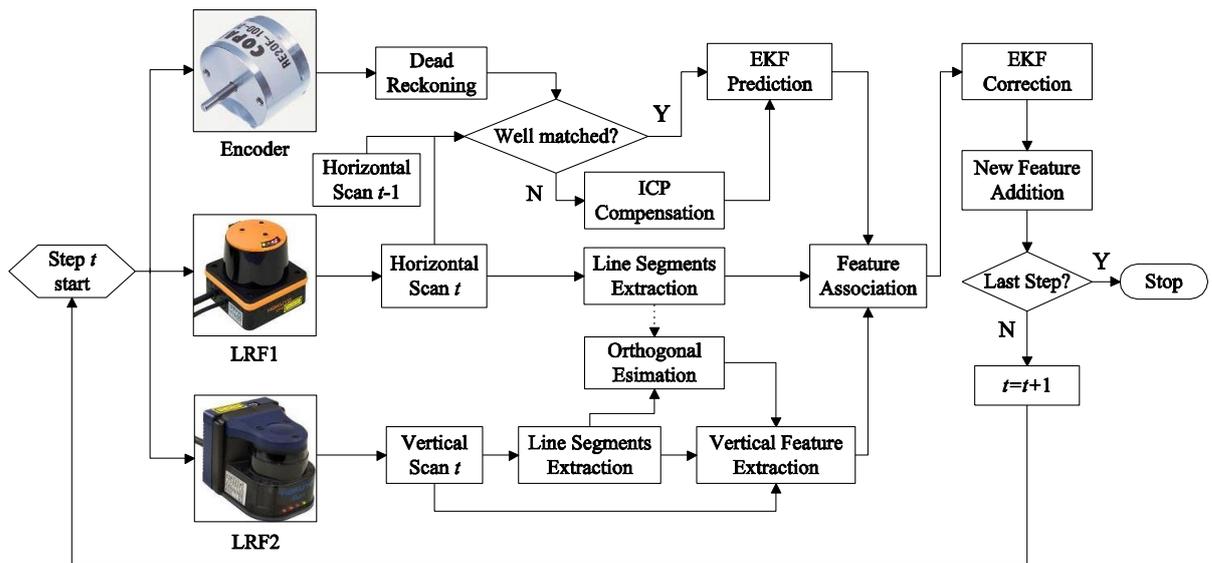


Figure 4-2 Overview of the system architecture

4.3 Line-segments Extraction

A good line extraction algorithm is expected to extract the parameters of the straight line-segments from raw points scanned by LRF accurately and swiftly. There are many famous algorithms that have been proposed in the past including incremental, split and merge, Hough transform, line regression, RANSAC, and expectation maximization algorithm. Based on the comparison that is conducted in [79] on these algorithms, the split and merge is employed in this study because of its superior speed and accuracy. As the name indicates, it firstly splits raw points into collinear clusters and then fits and merges the line-segments.

4.3.1 Points Split

Split procedure can be treated as a kind of data-preprocessing which splits the whole raw data from the LRF into several collinear data clusters. It consists of Point Distance Based Segmentation methods (PDBS) and Modified Iterative End Points Fitting (IEPF).

Point Distance Based Segmentation Methods

Breakpoints are the discontinued measurements in scanning process. As shown in Figure 4-3, there are clusters which are scanned from two separate objects. And $\mathbf{p}(i)$ and $\mathbf{p}(j)$ are the breakpoints between them. Point Distance Based Segmentation (PDBS) [80] methods detect these breakpoints by comparing Euclidean distance between two consecutive points with the breakpoint criteria D_{PDBS} .

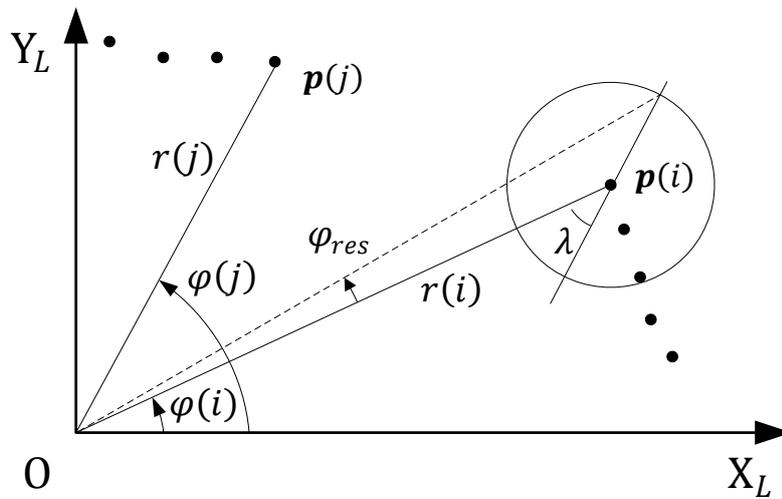


Figure 4-3 Principle of Adaptive Breakpoint Detector

The PDBS method that was adopted during this study is a famous approach called Adaptive Breakpoint Detector (ABD), which was proposed by Borges and Aldon [81].

This intuitive methodology determines breakpoint criteria D_{ABD} according to the range of the point. Figure 4-3 shows the principle of this methodology, a virtual line passing on the scan point $\mathbf{p}(i)$ has been defined, which represents an extreme case where an environment line can be reliably detected. Such a virtual line makes an angle λ with respect to the direction $\varphi(i)$ and aims to extrapolate the largest acceptable range of point $\mathbf{p}(i)$. Under this constraint, breakpoint criteria $D_{ABD}(i)$ of point $\mathbf{p}(i)$ can be calculated by using

$$D_{ABD}(i) = r(i) \cdot \frac{\sin(\varphi_{res})}{\sin(\lambda - \varphi_{res})} + 3\sigma_r \quad (4-1)$$

Intuitively, φ_{res} stands for the resolution angle of the LRF, and the parameter λ corresponds to the worst case of incidence angle of the laser scan ray with respect to a line for which the scan points are still reliable. This can be determined with user experience. Thus, such a detector generates a threshold circle which is centered at $\mathbf{p}(i)$ with radius $D_{ABD}(i)$, which adapts according to $r(i)$. In the case of Figure 4-3, since the next scan point $\mathbf{p}(j)$ is outside this circle, both $\mathbf{p}(i)$ and $\mathbf{p}(j)$ are treated as breakpoint points.

The parameter λ needs to be set by researchers based on how much the complementary angle of laser incidence is accepted. Most experiments done by other researchers [81][82] have shown that ABD performs satisfactorily when $\lambda = 10^\circ$.

Modified Iterative End Points Fitting

Iterative End Points Fitting (IEPF) is a well-known recursive algorithm for line extraction. Figure 4-4 demonstrates the principle of IEPF. For one cluster of points named $\mathcal{C}(i)$, this algorithm uses one hypothetical line $\mathbf{l}_{h(i)}$ that links the two end points first. Then it calculates distances of every point to the line $\mathbf{l}_{h(i)}$ so it can find the maximal distance d_i^{max} . If d_i^{max} is bigger than the predefined splitting threshold D_{Split} , IEPF splits the cluster of points $\mathcal{C}(i)$ into two subsets $\mathcal{C}(i)_1$ and $\mathcal{C}(i)_2$. This procedure is iterated to each subset until no new subsets can be found. IEPF performs satisfactory when it is applied to deal with some raw data scanned by short range LRF since usually the LRF shows high accuracy when it scans some short-distance objects. The threshold D_{Split} can be set to a small value and the points will be split sufficiently.

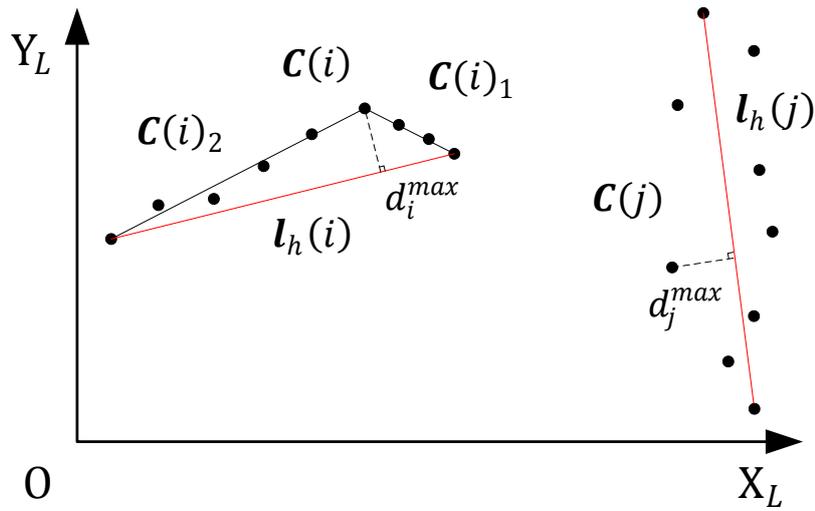


Figure 4-4 Principle of IEPF algorithm

However, the performance of IEPF deteriorates when it is applied to the raw data scanned by long range LRF. The problem can be explained by introducing the second cluster $C(j)$ in Figure 4-4. $C(i)$ is scanned from a close object that is constructed by 2 line-segments, while $C(j)$ is scanned from a far object that contains 1 line-segment only. The dilemma happens that the classic IEPF can not choose an appropriate threshold D_{split} which can split these two clusters properly. That is because the accuracy of the LRF decreases with the increment of scanning range, which leads to the consequence that the noises in the $C(j)$ are bigger than the maximum distance in the $C(i)$.

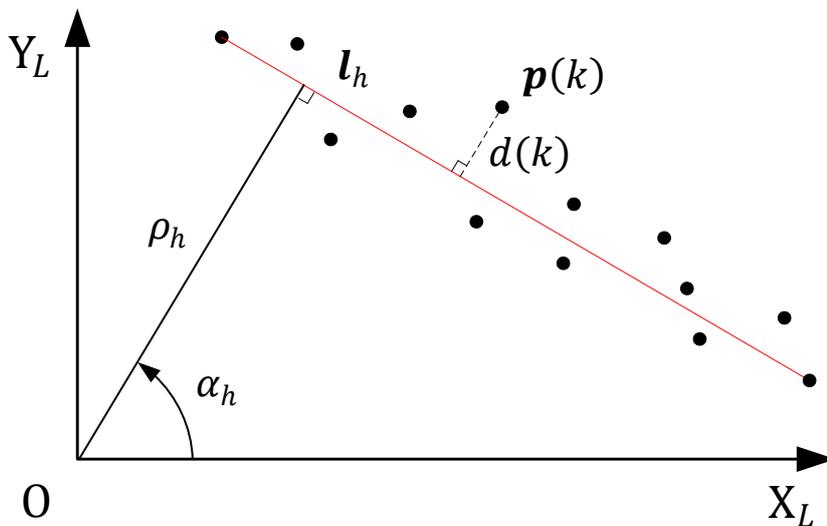


Figure 4-5 Points and its distance to the hypothetical line

To solve this problem, we propose MIEPF which changes its splitting criterion from a constant value D_{split} to a variable depending on the parameters of the hypothetical line $l_h = (\rho_h \ \alpha_h)^T$, as shown in Figure 4-5. For a measured point $p(k) = (r(k) \ \varphi(k))^T$ whose reality is $p_r(k) = (r_r(k) \ \varphi_r(k))^T$, according to the properties of LRF, the variance of $r(k)$ can be estimated as follows:

$$\sigma_{r(k)}^2 = \sigma_{r_0}^2, r_r(k) \leq r_0, \quad (4-2)$$

$$\sigma_{r(k)}^2 = (\varepsilon_0 r_r(k))^2, r_r(k) > r_0, \quad (4-3)$$

where ε_0 and r_0 are provided by the LRF manufacturer, $\sigma_{r_0}^2$ is the variance of the measurement at range r_0 . Since $r(k)$ is the good estimation of $r_r(k)$, the variance of $r(k)$ can be expressed as:

$$\sigma_{r(k)}^2 = \sigma_{r_0}^2, \text{ when } r(k) \leq r_0, \quad (4-4)$$

$$\sigma_{r(k)}^2 = (\varepsilon_0 r(k))^2, \text{ when } r(k) > r_0, \quad (4-5)$$

$d(k)$ is the distance between $p(k)$ and l_h . From the geometrical relationship, $d(k)$ can be calculated as:

$$d(k) = r(k) \cos(\alpha_h - \varphi(k)) - \rho_h, \quad (4-6)$$

In order to define the splitting criterion properly, the relationship between the variance of $d(k)$ and the distance of the object is required. Two simple assumptions are introduced here to simplify the deduction:

- 1) Assume the noises of the LRF's measurements are Gaussian noise and the variance of the $\varphi(k)$, $\sigma_{\varphi(k)}^2$ equals zero.
- 2) Assume the hypothetical line can well represent the object which contains 1 line-segment only. Thus the covariance of its parameters can be neglected.

The first assumption has been widely adopted by researchers and the second assumption is also acceptable since this is a qualitative deduction.

Based on Equation (4-6) and the assumptions, the variance of $d(k)$ can be obtained:

$$\sigma_{d(k)}^2 = \left(\cos(\alpha_h - \varphi(k)) \right)^2 \sigma_{d(k)}^2. \quad (4-7)$$

By substituting Equation (4-5) into Equation (4-7), we can estimate the variance of $d(k)$:

$$\sigma_{d(k)}^2 \approx (\varepsilon_0 \rho_h)^2. \quad (4-8)$$

Based on this conclusion, we introduce the following equation to define the splitting criterion:

$$D_{Split} = 3\sigma_{r_0} + \varepsilon(\rho_h - r_0), \quad (4-9)$$

where ε is one artificial setting ratio that takes the performance of the LRF and the user's experience into consideration. And the maximum D_{Split}^{max} will be set to restrict over loosing threshold based on the real performance of LRF.

4.3.2 Line-segments Fitting

By minimizing the sum of squared orthogonal distances between observed points and fitted line, Orthogonal Least-Squares Method (OLSM) extracts parameters of line $\mathbf{l} = [\rho \ \alpha]^T$ with respect to LRF's frame $O-X_L Y_L$ as follows

$$\begin{bmatrix} \rho \\ \alpha \end{bmatrix} \triangleq \begin{bmatrix} f_1(\mathbf{p}(1), \mathbf{p}(2), \dots, \mathbf{p}(n)) \\ f_2(\mathbf{p}(1), \mathbf{p}(2), \dots, \mathbf{p}(n)) \end{bmatrix} = \begin{bmatrix} \bar{x}\cos\alpha + \bar{y}\sin\alpha \\ \frac{1}{2} \arctan \frac{-2A_{xy}}{A_{y^2} - A_{x^2}} \end{bmatrix} \quad (4-10)$$

where

$$\begin{aligned} \bar{x} &= \frac{1}{n} \sum_{i=1}^n x(i), & \bar{y} &= \frac{1}{n} \sum_{i=1}^n y(i), \\ A_{x^2} &= \frac{1}{n} \sum_{i=1}^n (x(i) - \bar{x})^2, & A_{y^2} &= \frac{1}{n} \sum_{i=1}^n (y(i) - \bar{y})^2, \\ A_{xy} &= \frac{1}{n} \sum_{i=1}^n (x(i) - \bar{x})(y(i) - \bar{y}). \end{aligned}$$

As a necessary requirement for probabilistic approach such as EKF-SLAM processing, the covariance of parameters Σ_l of the extracted parameters $[\rho \ \alpha]^T$ is going to be estimated by using

$$\Sigma_l = \sum_{i=1}^n \mathbf{C}_i \mathbf{B}_i \Sigma_{m_i} \mathbf{B}_i^T \mathbf{C}_i^T = \begin{bmatrix} \sigma_\rho^2 & \sigma_{\rho\alpha} \\ \sigma_{\alpha\rho} & \sigma_\alpha^2 \end{bmatrix}, \quad (4-11)$$

where \mathbf{C}_i is the Jacobian matrix of $f(\mathbf{p})$ in Equation (4-10)

$$\mathbf{C}_i = \begin{bmatrix} C_i(1,1) & C_i(1,2) \\ C_i(2,1) & C_i(2,2) \end{bmatrix},$$

$$C_i(1,1) = \frac{\cos\alpha}{n} - \bar{x}\sin\alpha C_i(2,1) + \bar{y}\cos\alpha C_i(2,1),$$

$$C_i(1,2) = \frac{\sin\alpha}{n} - \bar{x}\sin\alpha C_i(2,2) + \bar{y}\cos\alpha C_i(2,2),$$

$$C_i(2,1) = \frac{(\bar{y} - y_i)(S_{y^2} - S_{x^2}) + 2S_{xy}(\bar{x} - x_i)}{(S_{y^2} - S_{x^2})^2 + 4S_{xy}^2},$$

$$C_i(2,2) = \frac{(\bar{x} - x_i)(S_{y^2} - S_{x^2}) - 2S_{xy}(\bar{y} - y_i)}{(S_{y^2} - S_{x^2})^2 + 4S_{xy}^2}.$$

And \mathbf{B}_i is the Jacobian matrix of $g(r, \varphi)$ in Equation (2-1).

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial g_1}{\partial r(i)} & \frac{\partial g_1}{\partial \varphi(i)} \\ \frac{\partial g_2}{\partial r(i)} & \frac{\partial g_2}{\partial \varphi(i)} \end{bmatrix} = \begin{bmatrix} \cos\varphi(i) & -r(i) \sin\varphi(i) \\ \sin\varphi(i) & r(i) \cos\varphi(i) \end{bmatrix}$$

The covariance of $[r(i), \varphi(i)]^T$ is

$$\Sigma_{m_i} = \begin{bmatrix} \sigma_{r(i)}^2 & \sigma_{r(i)\varphi(i)} \\ \sigma_{\varphi(i)r(i)} & \sigma_{\varphi(i)}^2 \end{bmatrix}.$$

For more mathematical principle details about OLSM, please refer to [84] and [85].

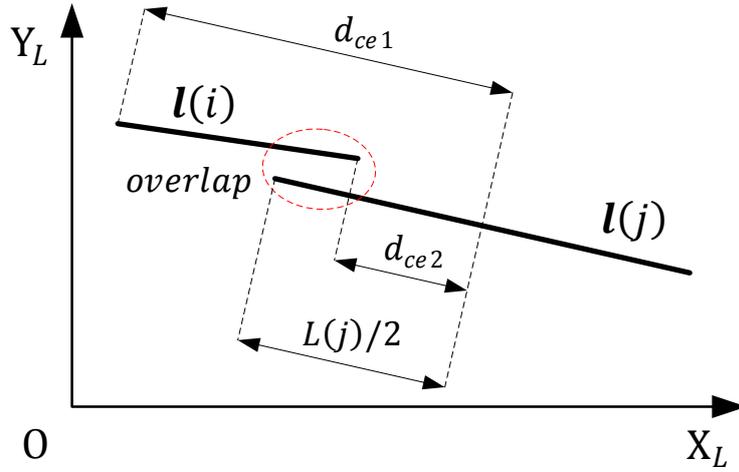


Figure 4-6 Line-segments overlap each other

4.3.3 Line-segment Merge

In the merge procedure, the lines that close enough and have similar parameters will be merged into one line. We use “three steps identification” to find and check the similar lines.

First, the coarse parameter test is conducted by setting thresholds T_ρ and T_α to find the line-segment pairs that are similar to each other.

Second, the overlap test is executed to check if two line-segments overlap each other. Given a pair of similar line-segments, the shorter line is marked as $\mathbf{l}(i)$ while the longer line is marked as $\mathbf{l}(j)$. The length of $\mathbf{l}(j)$ is given by $L(j)$. Along the direction of $\mathbf{l}(j)$, the distances from the endpoints of $\mathbf{l}(i)$ to the center of $\mathbf{l}(j)$, which are given by d_{ce1} and d_{ce2} respectively, are calculated. If either of d_{ce1} and d_{ce2} is smaller than $L(j)/2$, the overlap between $\mathbf{l}(i)$ and $\mathbf{l}(j)$ is confirmed and the pair will be treated as candidate association pair, as shown in .

Finally, given two line-segments $\mathbf{l}(i)$ and $\mathbf{l}(j)$ that satisfy the above criteria, the Mahalanobis Distance between two lines is calculated to distinguish whether they are qualified to be merged:

$$d(i, j) = (\mathbf{l}(i) - \mathbf{l}(j))^T (\boldsymbol{\Sigma}_{\mathbf{l}(i)} + \boldsymbol{\Sigma}_{\mathbf{l}(j)})^{-1} (\mathbf{l}(i) - \mathbf{l}(j)) < \chi_t^2. \quad (4-12)$$

In this study, the threshold χ_t^2 is set for 95% confidence level [86].

If this condition holds, then the lines are sufficiently similar to be merged. We can derive the final merged line estimation using a maximum likelihood formulation and can calculate the final merged line coordinates l_{new} and uncertainty $\Sigma_{l_{new}}$ as follows:

$$\Sigma_{l_{new}} = ((\Sigma_{l(i)})^{-1} + (\Sigma_{l(j)})^{-1})^{-1} \quad (4-13)$$

$$l_{new} = \Sigma_{l_{new}}((\Sigma_{l(i)})^{-1}l(i) + (\Sigma_{l(j)})^{-1}l(j)) \quad (4-14)$$

4.4 ICP Assisted Line-segments based EKF-SLAM

The first two parts of this section are brief reviews of work on 2D line-segment based EKF-SLAM algorithm [83]. The last subsection firstly introduce the ICP based EKF-SLAM prediction model. Then, a simple selection to pick a suitable prediction model is proposed.

4.4.1 Line-segments based EKF-SLAM

EKF-SLAM algorithm utilizes a large state vector to record the localization (robot pose state P) and mapping result (feature state F). The state vector is modeled by a Gaussian variable and maintained by the EKF through prediction, feature association, correction, and new feature addition.

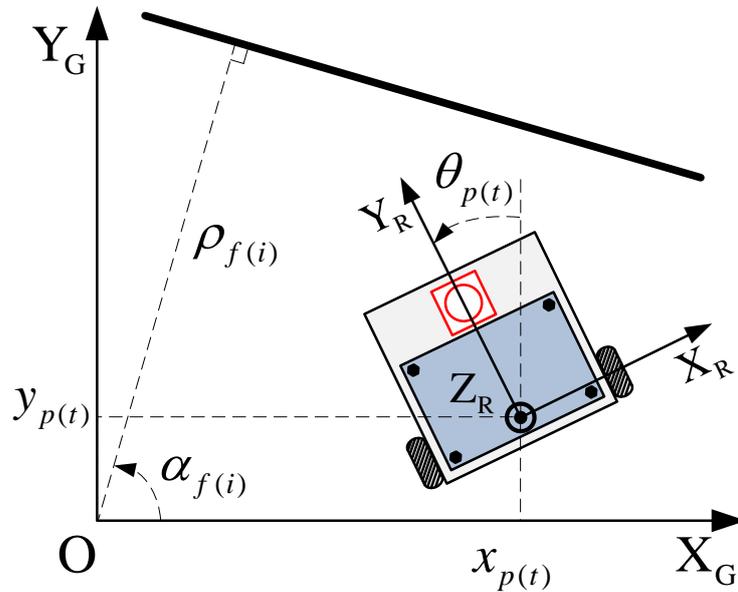


Figure 4-7 Geometrical relationship between the robot pose and feature parameters

At time t , the state vector of EKF-SLAM is expressed as follows:

$$\boldsymbol{\mu}_t = (\mathbf{P}_t \quad \mathbf{F}_1 \quad \cdots \quad \mathbf{F}_n)^T, \quad (4-15)$$

where the subscript " n " stands for features' number. As shown in Figure 4-7, the robot pose and the parameters of the features with respect to the global frame $O-X_G Y_G$ are as follows:

$$\mathbf{P}_t = (x_{p(t)} \quad y_{p(t)} \quad \theta_{p(t)}), \quad \mathbf{F}_i = (\rho_{f(i)} \quad \alpha_{f(i)}), \quad (4-16)$$

where the subscript " p " stands for pose, and " f " stands for feature. Moreover, the covariance matrix of state vector is expressed as follows:

$$\boldsymbol{\Sigma}_t = \begin{pmatrix} \boldsymbol{\Sigma}_{P_t} & \boldsymbol{\Sigma}_{P_t F_1} & \cdots & \boldsymbol{\Sigma}_{P_t F_n} \\ \boldsymbol{\Sigma}_{F_1 P_t} & \boldsymbol{\Sigma}_{F_1} & \cdots & \boldsymbol{\Sigma}_{F_1 F_n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{F_n P_t} & \boldsymbol{\Sigma}_{F_n F_1} & \cdots & \boldsymbol{\Sigma}_{F_n} \end{pmatrix}. \quad (4-17)$$

The endpoints of line-segments are stored and updated out of EKF frame.

4.4.2 Dead reckoning based EKF-SLAM Prediction

EKF-SLAM prediction is a procedure to estimate the transformation of robot pose between two steps such as from time $(t - 1)$ to time t . Based on the kinematic model shown in Figure 4-8, the predicted new pose of the robot as derived from the odometer based dead reckoning is calculated as follows:

$$\bar{\boldsymbol{\mu}}_t = g_{dr}(\boldsymbol{\mu}_{t-1}, \mathbf{u}_t), \quad (4-18)$$

which in detail:

$$\bar{x}_{p(t)} = x_{p(t-1)} - \frac{(\Delta R + \Delta L)}{2} \sin(\theta_{p(t-1)} + \frac{(\Delta R - \Delta L)}{2D}), \quad (4-19)$$

$$\bar{y}_{p(t)} = y_{p(t-1)} + \frac{(\Delta R + \Delta L)}{2} \cos(\theta_{p(t-1)} + \frac{(\Delta R - \Delta L)}{2D}), \quad (4-20)$$

$$\bar{\theta}_{p(t)} = \theta_{p(t-1)} + \frac{(\Delta R - \Delta L)}{D}, \quad (4-21)$$

where D stands for the distance between the rear wheels, $\mathbf{u}_t = (\Delta R \ \Delta L)$ denotes the travel distance of wheels measured by the odometer during the time interval, ΔR corresponds to the right wheel, while ΔL stands for the left one. The corresponding covariance matrix of the odometer measurement is as follows:

$$\mathbf{C}_w = \begin{pmatrix} \sigma_{\Delta R}^2 & 0 \\ 0 & \sigma_{\Delta L}^2 \end{pmatrix}. \quad (4-22)$$

The covariance matrix of state vector is updated by the equation given below:

$$\bar{\Sigma}_t = \mathbf{G}_p \Sigma_{t-1} \mathbf{G}_p^T + \mathbf{G}_w \mathbf{C}_w \mathbf{G}_w^T. \quad (4-23)$$

with the Jacobi matrices $\mathbf{G}_p = \partial g_{dr}(\boldsymbol{\mu}_{t-1}, \mathbf{u}_t) / \partial \boldsymbol{\mu}_{t-1}$ and $\mathbf{G}_w = \partial g_{dr}(\boldsymbol{\mu}_{t-1}, \mathbf{u}_t) / \partial \mathbf{u}_t$.

This typical dead reckoning based prediction model usually performs very well when the robot is moving at low rotational speeds with no serious slippage occurring. Another merit is that it requires low computational cost because of its simplicity.

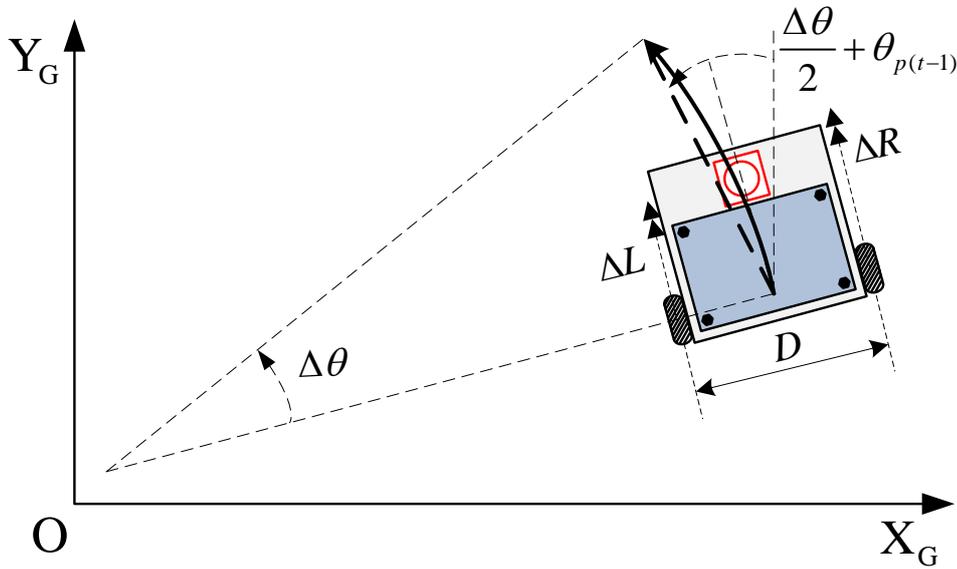


Figure 4-8 Kinematic model of robot

4.4.3 EKF-SLAM Features Association

After the horizontal scan, the LRF explores the environment at time t , with respect to horizontal robot frame $O-X_R Y_R$ (marked as frame R_h), a number of line-segments are extracted as the measurements:

$$\mathbf{z}_i = \left(\rho_{l(i)}^{R_h} \quad \alpha_{l(i)}^{R_h} \right)^T, i = 0, 1 \dots m, \quad (4-24)$$

with the corresponding covariances:

$$\mathbf{C}_i = \begin{pmatrix} \sigma_{\rho_{l(i)}^{R_h}}^2 & \sigma_{\rho_{l(i)}^{R_h} \alpha_{l(i)}^{R_h}} \\ \sigma_{\alpha_{l(i)}^{R_h} \rho_{l(i)}^{R_h}} & \sigma_{\alpha_{l(i)}^{R_h}}^2 \end{pmatrix}, i = 0, 1 \dots m. \quad (4-25)$$

Some of these measurements are scanned from the same objects that have been stored as features in the state vector. The main task of feature association is to find and associate these types of re-observed features so that the EKF-SLAM can correct the predicted state by comparing the measurements and stored features.

For j th stored feature, the estimated observation based on predicted state can be calculated as follows:

$$\hat{\mathbf{z}}_j = \begin{pmatrix} \hat{\rho}_{f(j)}^{R_h} \\ \hat{\alpha}_{f(j)}^{R_h} \end{pmatrix} = h(\bar{\boldsymbol{\mu}}_t) = \begin{pmatrix} \rho_{f(j)} - \bar{x}_{p(t)} \cos \alpha_{f(j)} - \bar{y}_{p(t)} \sin \alpha_{f(j)} \\ \alpha_{f(j)} - \bar{\theta}_{p(t)} \end{pmatrix}. \quad (4-26)$$

The corresponding covariance matrix of $\hat{\mathbf{z}}_j$ is as follows:

$$\hat{\mathbf{C}}_j = \mathbf{H}_j \bar{\boldsymbol{\Sigma}}_t \mathbf{H}_j^T, \quad (4-27)$$

with the Jacobi matrix $\mathbf{H}_j = \partial h(\bar{\boldsymbol{\mu}}_t) / \partial \bar{\boldsymbol{\mu}}_t$, and $\bar{\boldsymbol{\Sigma}}_t$ is the latest covariance matrix of the state vector.

To associate measurements with estimated observations, from all possible line pairs, the pairs that have similar parameters are firstly picked out by simply comparing the differences of their parameters with the preset thresholds. After a series of candidate association pairs $\{(\mathbf{z}_i, \hat{\mathbf{z}}_j), \dots\}$ can be picked out. And the Mahalanobis Distance of each pair is calculated as given below:

$$d_{ij} = (\mathbf{z}_i - \hat{\mathbf{z}}_j)^T (\mathbf{C}_i + \hat{\mathbf{C}}_j)^{-1} (\mathbf{z}_i - \hat{\mathbf{z}}_j). \quad (4-28)$$

Between the calculated results of the pairs, the minimum value d_{min} is compared with an artificial criterion χ^2 .

$$d_{min} < \chi^2. \quad (4-29)$$

If this judgment is satisfied, the corresponding pair is successfully associated. After the correction procedure based on this associated pair is executed, another iteration of feature association will be carried out until no more pair can be associated.

4.4.4 EKF-SLAM Correction

For the observation \mathbf{z}_i that has been associated with the feature \mathbf{F}_j , EKF-SLAM will implement the correction process to the state vector and its covariance matrix. The EKF correction process is typically written as follows:

$$\mathbf{K}_j = \bar{\boldsymbol{\Sigma}}_t \mathbf{H}_j^T (\mathbf{C}_i + \hat{\mathbf{C}}_j)^{-1}, \quad (4-30)$$

$$\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \mathbf{K}_j (\mathbf{z}_i - \hat{\mathbf{z}}_j), \quad (4-31)$$

$$\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_j \mathbf{H}_j) \bar{\boldsymbol{\Sigma}}_t. \quad (4-32)$$

\mathbf{K}_j is usually called Kalman gain, which is calculated by utilizing the uncertainties of the real measurements and the estimated observations. After the Kalman gain has been calculated, the state vector and corresponding covariance are updated by (4-31) and (4-32).

4.4.5 EKF-SLAM New Feature Addition

For the measurement \mathbf{z}_i that has failed to associate with any stored features, EKF-SLAM will treat it as a new detected feature. Thus \mathbf{z}_i will become the $(n+1)^{th}$ feature \mathbf{F}_{n+1} in the state vector:

$$\begin{aligned} \mathbf{F}_{n+1} &= \begin{pmatrix} \rho_{f(n+1)} \\ \alpha_{f(n+1)} \end{pmatrix} = f(\mathbf{P}_t, \mathbf{z}_i) \\ &= \begin{pmatrix} \rho_{l(i)}^{Rh} + x_{p(t)} \cos \alpha_{f(n+1)} + y_{p(t)} \sin \alpha_{f(n+1)} \\ \theta_{p(t)} + \alpha_{l(i)}^{Rh} \end{pmatrix}. \end{aligned} \quad (4-33)$$

Moreover, the covariance matrix of the state vector also has to be augmented as follows:

$$\Sigma_t = \begin{pmatrix} \Sigma_{P_t} & \cdots & \Sigma_{P_t F_n} & \Sigma_{P_t}^T J_p^T \\ \vdots & \ddots & \vdots & \vdots \\ \Sigma_{F_n P_t} & \cdots & \Sigma_{F_n} & \Sigma_{F_n P_t} J_p^T \\ J_p \Sigma_{P_t} & \cdots & J_p \Sigma_{P_t F_n} & J_p \Sigma_{P_t} J_p^T + J_l C_l J_l^T \end{pmatrix}, \quad (4-34)$$

where $J_p = \partial f(\mathbf{P}_t, \mathbf{z}_i) / \partial \mathbf{P}_t$ and $J_l = \partial f(\mathbf{P}_t, \mathbf{z}_i) / \partial \mathbf{z}_i$ are the Jacobi matrices that linearize the new feature with respect to robot pose and measurement variables, respectively.

4.4.6 Dead Reckoning Result Evaluation and ICP Compensation

The prediction procedure is critical for EKF-SLAM since it usually has a great influence on the subsequent feature association process. In spite of its high speed, pure dead reckoning based prediction is risky because of the unpredictable slippage and increasing linearization error of the kinematic model when the rotational speed is increased. Some fatal errors may lead to the failure of the feature association process and contribute to the divergence of EKF-SLAM. To overcome this inconvenience and make the prediction more robust, ICP is introduced to evaluate the dead reckoning result and to compensate it when necessary.

ICP based EKF-SLAM prediction

In order to estimate the accurate relative transformation of the robot pose $\mathbf{T}_{m(t)} = (\Delta x_{m(t)} \quad \Delta y_{m(t)} \quad \Delta \theta_{m(t)})^T$ between time $(t-1)$ and time t , ICP is applied to match horizontal scans $\mathbf{S}_{h(t-1)}$ and $\mathbf{S}_{h(t)}$. Noted that instead of referring to the global frame, $\mathbf{T}_{m(t)}$ is a relative transformation with respect to the robot frame at time $(t-1)$. The initial estimation $\mathbf{T}_{m(t)}^0$ is given by the odometer-based dead reckoning result as follows:

$$\Delta\theta_{m(t)}^0 = \frac{(\Delta R - \Delta L)}{D}, \quad (4-35)$$

$$\Delta x_{m(t)}^0 = -\frac{(\Delta R + \Delta L)}{2} \sin\left(\frac{\Delta\theta_{m(t)}^0}{2}\right), \quad (4-36)$$

$$\Delta y_{m(t)}^0 = \frac{(\Delta R + \Delta L)}{2} \cos\left(\frac{\Delta\theta_{m(t)}^0}{2}\right). \quad (4-37)$$

After ICP has run k iterations and arrived at a fit solution $\mathbf{T}_{m(t)}^k$, the corresponding covariance $\boldsymbol{\Sigma}_{m(t)}$ has to be determined with the intention of EKF-SLAM prediction. By utilizing a closed-form estimation method [87], the covariance is derived as follows:

$$\boldsymbol{\Sigma}_{m(t)} = \left(\frac{\partial^2 E_{icp}}{\partial \mathbf{T}^2}\right)^{-1} \frac{\partial^2 E_{icp}}{\partial \mathbf{S} \partial \mathbf{T}} \boldsymbol{\Sigma}(\mathbf{S}) \frac{\partial^2 E_{icp}}{\partial \mathbf{S} \partial \mathbf{T}}^T \left(\frac{\partial^2 E_{icp}}{\partial \mathbf{T}^2}\right)^{-1}, \quad (4-38)$$

where E_{icp} is the sum of distance error in Equation (2-14), and $\boldsymbol{\Sigma}(\mathbf{S})$ stands for the covariance of measurements $\mathbf{S}_{h(t-1)}$ and $\mathbf{S}_{h(t)}$. The necessary derivatives $\partial^2 E_{icp} / \partial \mathbf{T}^2$ and $\partial^2 E_{icp} / \partial \mathbf{S} \partial \mathbf{T}$ can easily be computed in a closed form based on the analytical solution of $\mathbf{T}_{m(t)}$ as stated in [49]. Finally, the robot pose is predicted by ICP as follows:

$$\bar{\boldsymbol{\mu}}_t = g_m(\boldsymbol{\mu}_{t-1}, \mathbf{T}_{m(t)}^k), \quad (4-39)$$

which is detailed as:

$$\bar{x}_{p(t)} = x_{p(t-1)} + \Delta x_{m(t)}^k \cos\theta_{p(t-1)} - \Delta y_{m(t)}^k \sin\theta_{p(t-1)}, \quad (4-40)$$

$$\bar{y}_{p(t)} = y_{p(t-1)} + \Delta x_{m(t)}^k \sin\theta_{p(t-1)} + \Delta y_{m(t)}^k \cos\theta_{p(t-1)}, \quad (4-41)$$

$$\bar{\theta}_{p(t)} = \theta_{p(t-1)} + \Delta\theta_{m(t)}^k. \quad (4-42)$$

The covariance matrix of the state vector is updated by the following equation:

$$\bar{\boldsymbol{\Sigma}}_t = \mathbf{G}_p \boldsymbol{\Sigma}_{t-1} \mathbf{G}_p^T + \mathbf{G}_m \boldsymbol{\Sigma}_{m(t)} \mathbf{G}_m^T. \quad (4-43)$$

with the Jacobi matrices $\mathbf{G}_p = \partial g_m(\boldsymbol{\mu}_{t-1}, \mathbf{T}_{m(t)}^k) / \partial \boldsymbol{\mu}_{t-1}$ and $\mathbf{G}_m = \partial g_m(\boldsymbol{\mu}_{t-1}, \mathbf{T}_{m(t)}^k) / \partial \mathbf{T}_{m(t)}^k$.

Optional EKF-SLAM prediction

Although the ICP based EKF prediction gives robust result, it is associated with a much higher computational cost than dead reckoning for both the iterative process and the final covariance estimation of the ICP result. Therefore, an optional EKF-SLAM prediction model is proposed by selecting two addressed models based on the evaluation of the dead reckoning result.

To verify whether the dead reckoning result is precise enough, only one iteration of ICP is executed with the rejection criterion D_{icp}^{eva} , which is determined by the D_{icp}^{min} plus one small error tolerance. The association ratio P_{match} and average distance \bar{E}_{icp} can be found after a single iteration. Then, a simple judgment is conducted:

$$P_{match} > P_{match}^{eva} \ \&\& \ \bar{E}_{icp} < E_{icp}^{eva}, \quad (4-44)$$

where P_{match}^{eva} is the association ratio threshold and E_{icp}^{eva} is the error index threshold. The judgment (4-44) is satisfied only when two successive scans overlap mostly with low errors, which means the dead reckoning result provides a good estimation. Thus, if the judgment is satisfied, the faster dead reckoning based EKF prediction is going to be conducted. Otherwise, further ICP iteration has to be employed with the reinitialized rejection threshold D_{icp}^{max} so that ICP based EKF prediction will be executed to guarantee the robustness of the prediction.

4.5 Slope and edge feature in EKF-SLAM

4.5.1 Vertical Feature Extraction and Integration

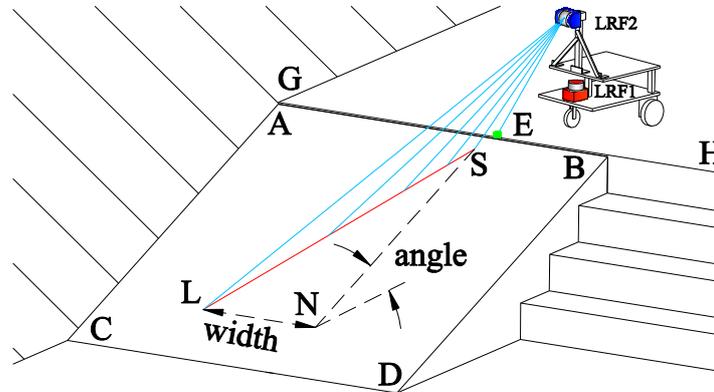


Figure 4-9 Slope and edge detection by the vertical scan

The basic concept of feature detection and extraction in the vertical scan is shown in Figure 4-9. A slope marked as $ABCD$ (corresponding to 4 corners) is placed next to an edge marked as GH . When LRF2 finds a line-segment SL scanned from the slope, partial information of the slope can be extracted by decomposing vector \overline{SL} into \overline{SN} and \overline{NL} . And the distance between the robot and the edge can be estimated from the detected edge points E by similar decomposition. For correct decomposition, the exact orientation of the slope and edge line needs to be known in advance; this can be solved by utilizing the horizontal scanned line-segments with the orthogonal assumption.

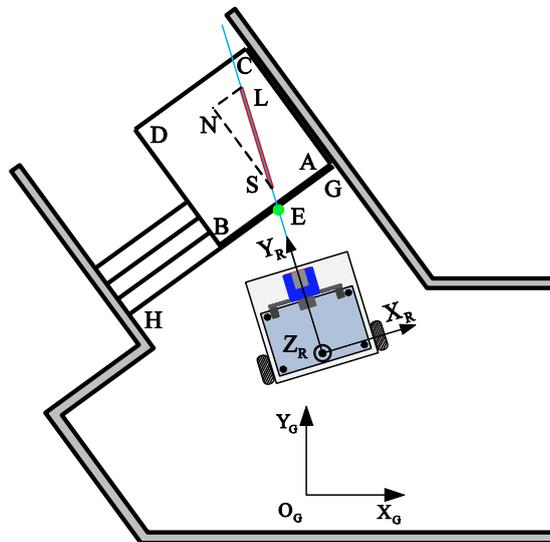


Figure 4-10 Simplified top view of the vertical feature detection

Since this research focuses on the 2D map, a top view of Figure 4-9 is represented in Figure 4-10. To simplify the explanation of the slope and edge detection, the vertical scanning plane is designed to be in the same plane as $O\text{-}Y_R Z_R$. The main extraction procedure is executed in horizontal frame $O\text{-}X_R Y_R$ (marked as frame R_h) and vertical frame $O\text{-}Y_R Z_R$ (marked as frame R_v).

Orthogonal angle estimation

The exact orientation of the slope and edge line is estimated with a well-known geometrical constraint named orthogonal assumption. When a slope or edge is detected by LRF2, the local orthogonal angle $\alpha_o^{R_h}$ and its variance $\sigma_{\alpha_o^{R_h}}^2$ are calculated [88] based on the horizontal scanned line-segments as shown in Figure 4-11. Since the slope is typically located in the corridor, $\alpha_o^{R_h}$ can precisely indicate the orientation of the slope with respect to the robot's horizontal frame $O\text{-}X_R Y_R$.

The global orthogonal angle α_o can be added to the EKF-SLAM state vector μ_t as a feature \mathbf{F}^O .

$$\mathbf{F}^O = (\alpha_o) = f_o(\mathbf{P}_t, \alpha_o^{R_h}) = (\theta_{p(t)} + \alpha_o^{R_h}). \quad (4-45)$$

Moreover, the covariance matrix of the state vector can be augmented in a similar way as in Equation (4-34), and it will not be detailed here.

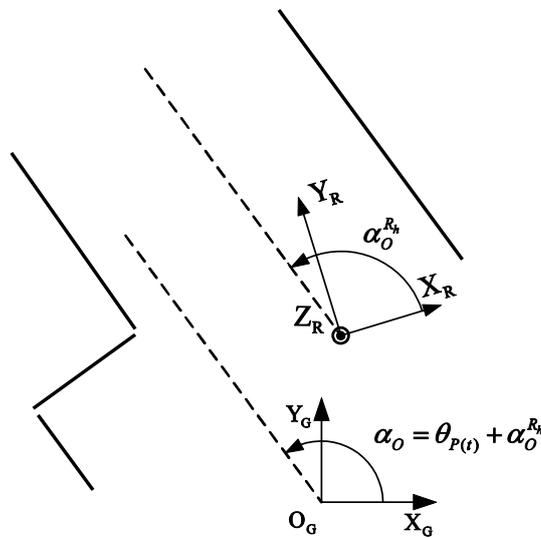


Figure 4-11 Orthogonal angle estimation of the horizontal scan

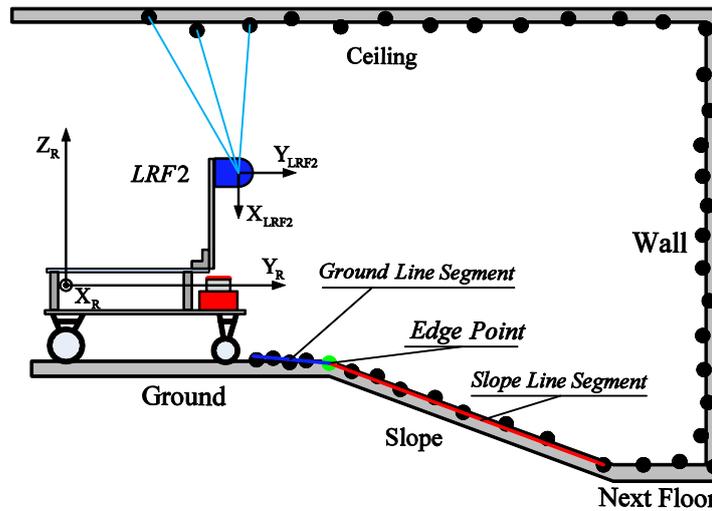


Figure 4-12 Vertical scanning plane and features of interest

Another application of the orthogonal assumption is the vertical scan correction. The central axis of LRF2 is designed to be parallel to the ground. However, even in the indoor environment, the ground where the robot is moving cannot be absolutely flat and smooth. The space between tiles can easily affect the parallelism, which may lead to errors during the vertical feature extraction. To overcome this problem, orthogonal estimation of the vertical scan is executed in every step to compensate the error. This is feasible because most features in the vertical scan plane, such as ceiling, wall, and ground, are parallel or perpendicular to each other as shown in Figure 4-12.

Vertical scan classification

Before starting to build the line-segment model of the slope and edge, the vertical scanned data needs to be classified under the frame of LRF2. Basically, the classification is executed after extraction of line-segments. The slope line-segment can be easily found by checking the parameters such as length, range, angle, and covariance. For edge point detection, two situations must be considered. The first situation is that the edge point is the intersection point of the ground line and down slope line-segment, as shown in Figure 4-12. The second situation is shown in Figure 4-13; a short ground line-segment has been detected without following slope line-segment. In this case, the remote endpoint of the ground line-segment will be treated as the edge point if the incidence angle of its ray is bigger than a given threshold ω_T^E and no points are scanned from other features in the range of d_T^E to the edge point.

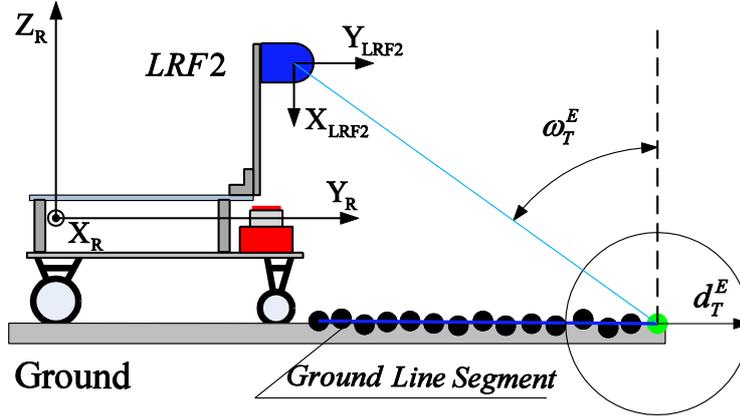


Figure 4-13 Second situation of edge point detection

Line-segments modeling and integration of edge

In a 2D map, the key characteristic of the edge can be perfectly represented by the edge line, which is marked as line-segment HG in Figure 4-10. However, the vertical scan plane can only detect one edge point that is supposed to be the closest one to the real edge line. Thus, as shown in Figure 4-14, the edge feature F^E that is going to be added into the state vector can be estimated by utilizing the single detected edge point E , the robot pose P_t , and the corresponding orthogonal angle α_o as follows:

$$\begin{aligned} \mathbf{F}^E = (\rho_e) &= f_E(\mathbf{P}_t, \mathbf{F}^O, y_e^R) \\ &= (y_e^R \sin(\alpha_o - \theta_{p(t)}) + x_{p(t)} \cos \alpha_o + y_{p(t)} \sin \alpha_o). \end{aligned} \quad (4-46)$$

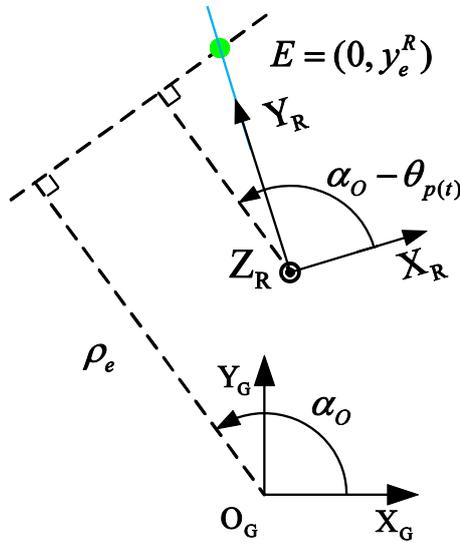


Figure 4-14 Edge line estimation based on the edge point E

And the covariance matrix of the state vector also has to be augmented as follows:

$$\begin{aligned} \Sigma_t &= \begin{pmatrix} \Sigma_{(P_t F^O)} & \Sigma_{(P_t F^O)F_i} & \Sigma_{(P_t F^O)F^E} \\ \Sigma_{F_i(P_t F^O)} & \Sigma_{F_i} & \Sigma_{F_i F^E} \\ \Sigma_{F^E(P_t F^O)} & \Sigma_{F^E F_i} & \Sigma_{F^E} \end{pmatrix} \\ &= \begin{pmatrix} \Sigma_{(P_t F^O)} & \Sigma_{(P_t F^O)F_i} & \Sigma_{(P_t F^O)}^T J_\mu^T \\ \Sigma_{F_i(P_t F^O)} & \Sigma_{F_i} & \Sigma_{F_i(P_t F^O)} J_\mu^T \\ J_\mu \Sigma_{(P_t F^O)} & J_\mu \Sigma_{(P_t F^O)F_i} & J_\mu \Sigma_{(P_t F^O)} J_\mu^T + J_y C_y J_y^T \end{pmatrix} \end{aligned} \quad (4-47)$$

where C_y is the variance of the measurement y_e^R , the Jacobi matrices $J_\mu = \partial f_E(\mathbf{P}_t, \mathbf{F}^O, y_e^R) / \partial (\mathbf{P}_t, \mathbf{F}^O)$ and $J_y = \partial f_E(\mathbf{P}_t, \mathbf{F}^O, y_e^R) / \partial y_e^R$.

$$\Sigma_{(P_t F^O)} = \begin{pmatrix} \Sigma_{P_t} & \Sigma_{P_t F^O} \\ \Sigma_{F^O P_t} & \Sigma_{F^O} \end{pmatrix}. \quad (4-48)$$

The first added edge feature F^E is incomplete because it has no length with only one endpoint instead of two. This problem will be solved when another edge point is scanned and then the length can be estimated.

Line-segments modeling and integration of slope

There are several key characteristics of slope such as width, gradient, length, and location. The location and width can be represented by the slope start line, which can be treated as line-segment AB in Figure 4-10. The estimation of slope start line, however, differs from the edge line estimation. Instead of finding a point that is closest to the slope start line, the intersection point between the scanned slope line-segment and ground is adopted. This is because the start line may not be scanned even when some line-segment is scanned from the slope, as shown in Figure 4-15.

From the geometrical relationship shown in Figure 4-16 (a), the distance y_s^R , which is measured from robot origin to the intersection point in Y_R axis direction, can be calculated by using the robot frame height h_R and the scanned slope line-segment $l_{sl}^{R_v}$ in the plane $O-Y_R Z_R$ as

$$y_s^R = \frac{\rho_{sl}^{R_v} - h_R \cos \alpha_{sl}^{R_v}}{\sin \alpha_{sl}^{R_v}}, \quad (4-49)$$

where $\alpha_{sl}^{R_v}$ is the angular parameter of $l_{sl}^{R_v}$ with respect to the negative Z_R axis and is measured positive in counter-clockwise direction. After y_s^R has been estimated, the slope feature F^S , which contains the distance parameter of slope start line, can be calculated as follows:

$$\begin{aligned} \mathbf{F}^S &= (\rho_s) = f_s(\mathbf{P}_t, \mathbf{F}^O, y_s^R) \\ &= (y_s^R \sin(\alpha_o - \theta_{p(t)}) + x_{p(t)} \cos \alpha_o + y_{p(t)} \sin \alpha_o). \end{aligned} \tag{4-50}$$

Moreover, the covariance matrix of the state vector can be augmented in a similar way as with the edge line.

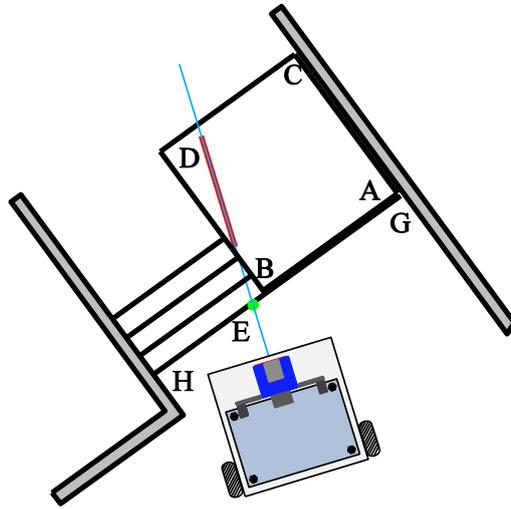
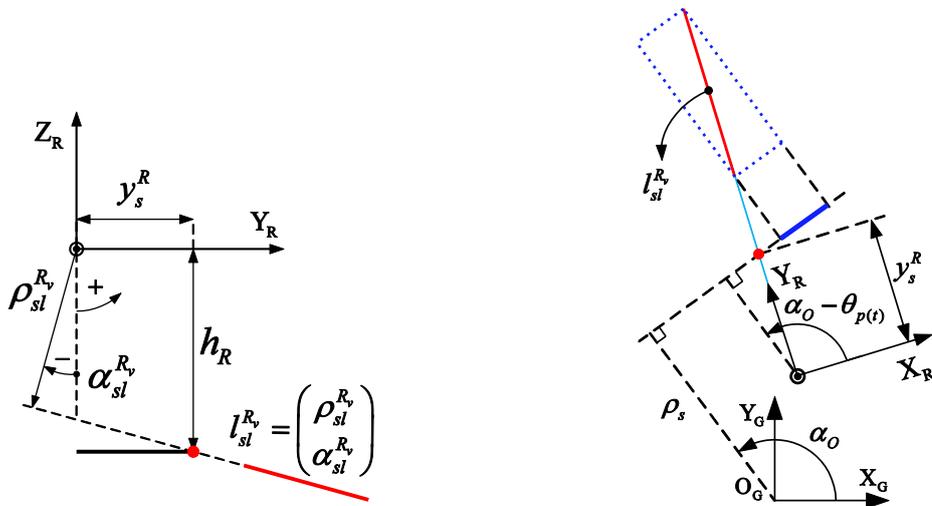


Figure 4-15 Slope is detected without scanning the start line



(a) Scanned slope line-segment in vertical scan (b) Scanned slope line and start line of slope

Figure 4-16 Slope start line and scanned area estimation

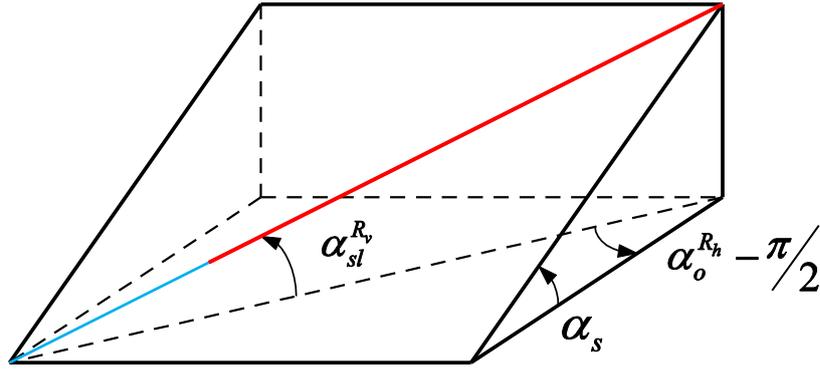


Figure 4-17 Slope model with the scanned slope line-segment

The last key characteristic of the slope is the gradient. From the slope model plotted in Figure 4-17, the geometric relationship between the slope gradient $\tan\alpha_s$ and the scanned line angle $\alpha_{sl}^{R_v}$ can be found as follows:

$$\tan\alpha_s = \frac{\tan\alpha_{sl}^{R_v}}{\cos\alpha_o^{R_h}}, \quad (4-51)$$

where $\alpha_o^{R_h}$ is the local orthogonal angle that has been discussed before. The slope gradient can be excluded from the EKF-SLAM frame because it is independent of the coordinate transformation in 2D frame. From every vertical scan that finds the same slope, the slope gradient $\tan\alpha_s$ can be updated by a simple independent one-dimensional EKF with the observation Equation (4-51).

Until this point, three necessary characteristics of edge and slope have been successfully integrated into 2D EKF-SLAM frame. They are treated as one set but added as three individual features because slope and edge may not be detected simultaneously. Other features such as the endpoints are stored and updated out of EKF framework.

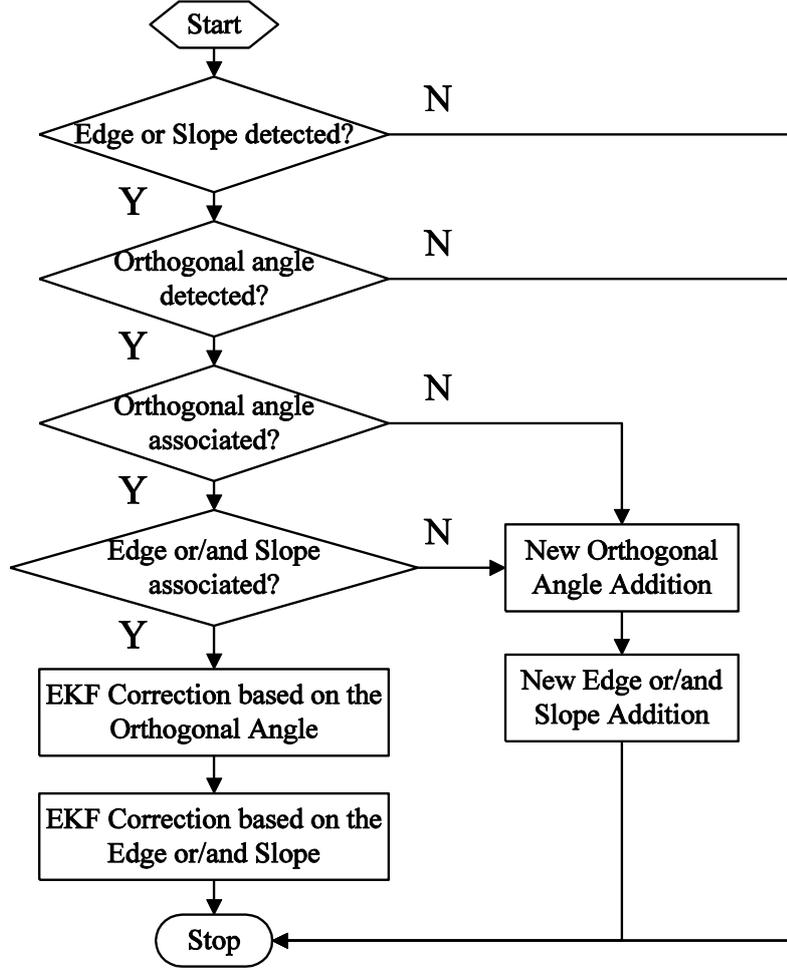


Figure 4-18 The detail flowchart of the vertical features process in the EKF-SLAM frame

4.5.2 Vertical Feature Association and Correction

Although a rough EKF processing of vertical features can be found in Figure 4-2, a detailed flowchart is plotted in Figure 4-18 because the feature association of orthogonal angle and vertical feature (edge or slope) are dependent; vertical features differ from the horizontal features. For a stored orthogonal angle α_o and its corresponding edge and slope, the estimated observations are as follows:

$$\hat{\alpha}_o^{Rh} = h_o(\mu_t) = (\alpha_o - \theta_{p(t)}), \quad (4-52)$$

$$\hat{\rho}_e^{Rh} = h_E(\mu_t) = (\rho_e - x_{p(t)}\cos\alpha_o - y_{p(t)}\sin\alpha_o), \quad (4-53)$$

$$\hat{\rho}_s^{Rh} = h_S(\mu_t) = (\rho_s - x_{p(t)}\cos\alpha_o - y_{p(t)}\sin\alpha_o). \quad (4-54)$$

Moreover, the estimation of their corresponding covariance can be expressed in a general equation as follows:

$$\widehat{\mathbf{C}}_X = \mathbf{H}_X \boldsymbol{\Sigma}_t \mathbf{H}_X^T, \quad (4-55)$$

where Jacobi matrix $\mathbf{H}_X = \partial h_X(\boldsymbol{\mu}_t) / \partial \boldsymbol{\mu}_t$ and the subscript X can stand for each feature. The association between measurement \mathbf{z}_X and estimated observation $\hat{\mathbf{z}}_X$ is based on their Mahalanobis distance:

$$d_X = (\mathbf{z}_X - \hat{\mathbf{z}}_X)^T (\mathbf{C}_X + \widehat{\mathbf{C}}_X)^{-1} (\mathbf{z}_X - \hat{\mathbf{z}}_X) < \chi^2. \quad (4-56)$$

When a vertical feature has been detected in the scan, a local horizontal orthogonal angle α_o^{Rh} is firstly going to be estimated as the measurement and then going to associate the measured angle with a stored orthogonal angle. If no stored orthogonal angle can be associated with this measurement, the local orthogonal angle and detected vertical feature will be added as a new set of features. Otherwise, the stored vertical feature that corresponds to the associated α_o will try to associate with the detected vertical features. If the vertical feature association fails, the orthogonal angle and detected vertical features are still going to be added as a new set of features. If the association is successful, the EKF correction will be executed based on the associated features.

4.6 Simulation

Figure 4-19 shows the environment of the simulation. The black lines stand for the walls. And the black patches that are marked from (1) to (4) represent for the sequences of slopes while the black dashed lines that are marked from {1} to {7} denote the sequences of edges. Slope 1 and Slope 4 are up slopes while Slope 2 and Slope 3 are down slopes. Finally, the red lines with arrows show the robot route. The red dot means that the robot stops at this position and rotates following the arrows' pointing.

Figure 4-20 plots out the trajectory estimated by odometer based dead-reckoning method. The error is accumulated and makes result poor. Figure 4-21 (a) presents the result of EKF-SLAM without slope and edge detection while Figure 4-21 (b) plots out the result of EKF-SLAM that not only finds the feature walls but also maintains the key information of the detected edges and slopes. The differences between the errors of the poses estimated by two EKF-SLAM algorithms are neglectable, as shown in the Figure 4-22.

Figure 4-23 demonstrates the error of the detected 4 slopes' paramters. The first column gives the error of the estimated orthogonal angles which are the angular value of slope start line. The second column presents the error of the distance value of slope start line. Note that the errors of start lines are more or less diminished after loop has been detected and corrected in the final steps. The last column shows the error of the estimated slopes' gradients which is only updated when the robot is scanning on the slope. This is because the gradient of slope is free from robot location error.

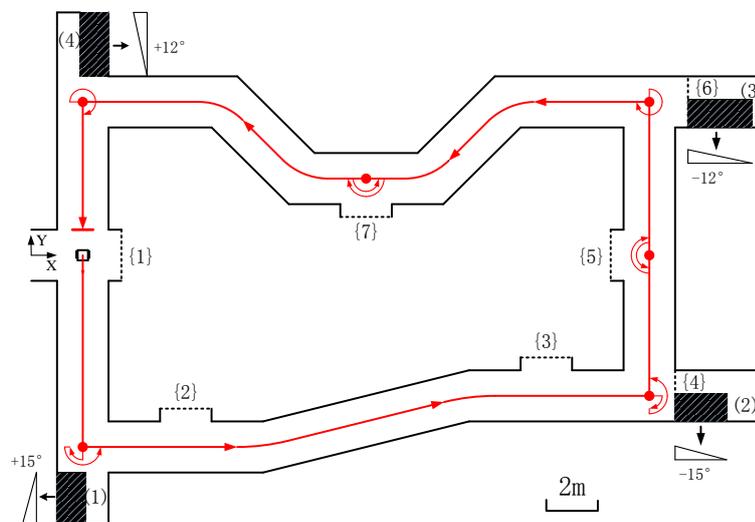


Figure 4-19 Simulation Environment and Robot Trajectory

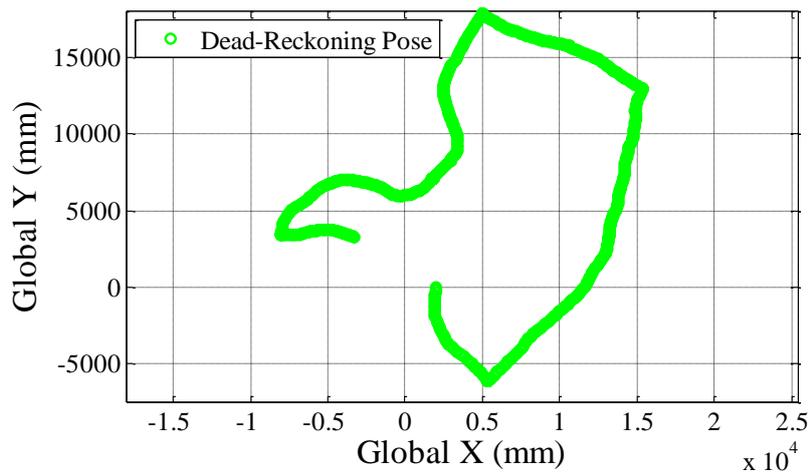
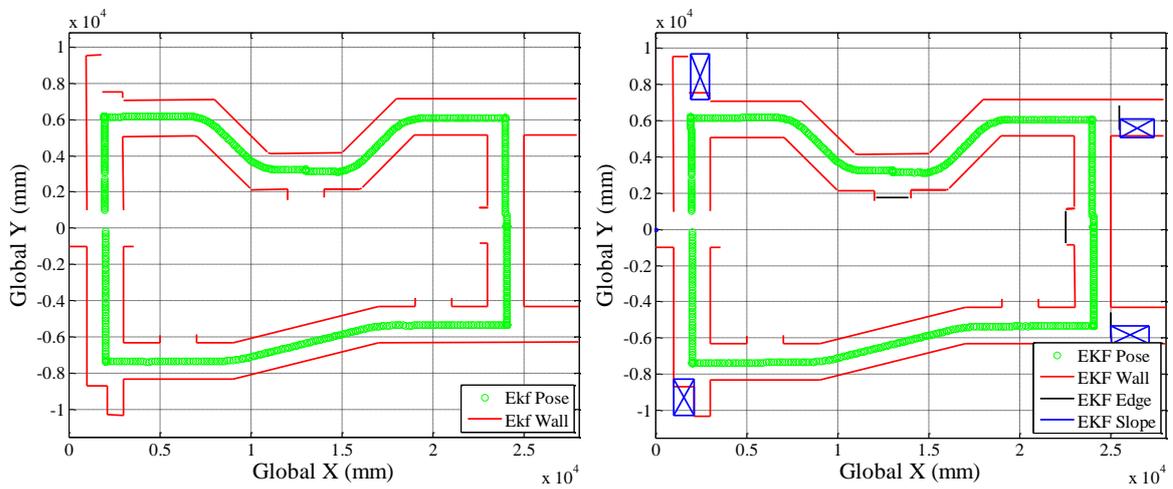


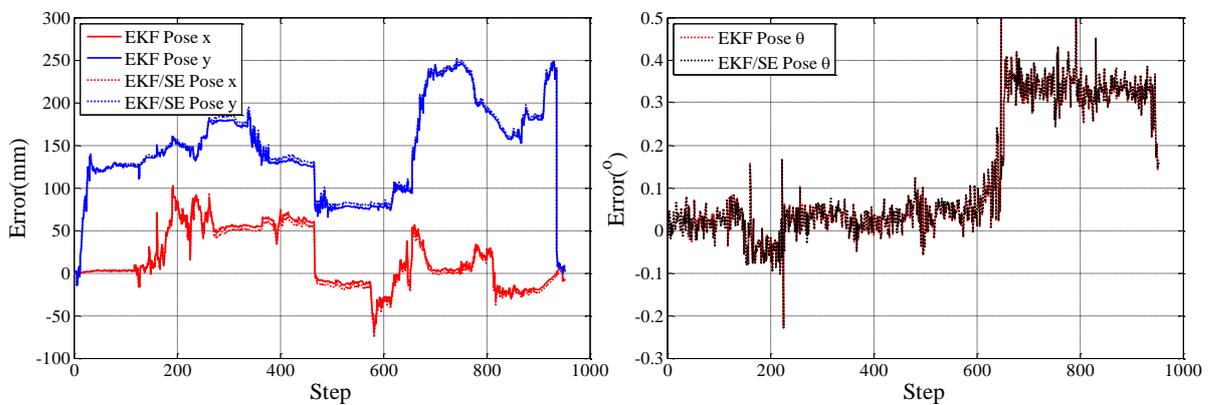
Figure 4-20 Trajectory estimated by odometer based dead-reckoning method



(a) without slope and edge detection

(b) with slope and edge detection

Figure 4-21 Line-segments based EKF-SLAM results with/without slope and edge detection



(a) translational error

(b) rotational error

Figure 4-22 The errors of the estimated poses along the motion

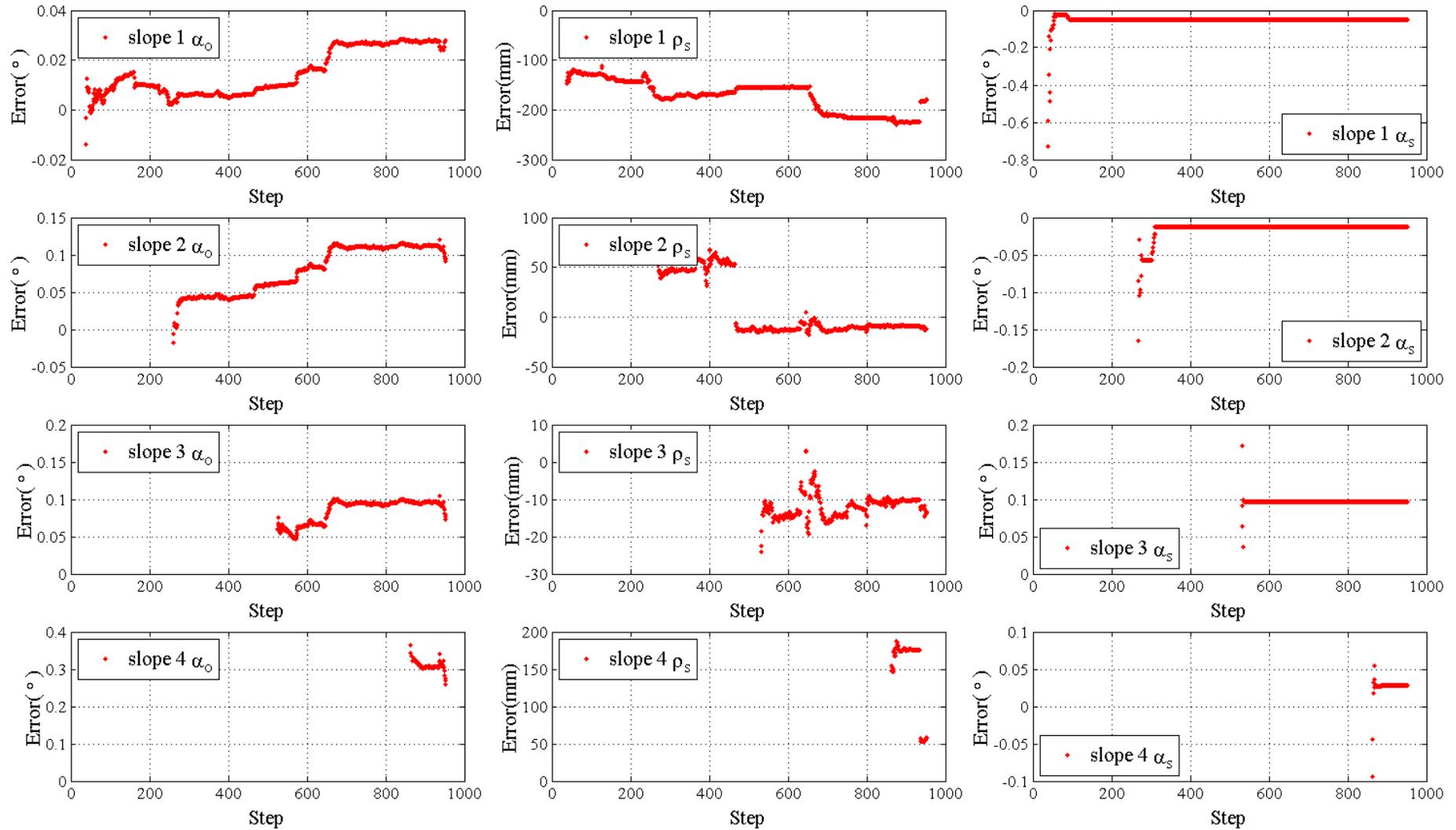


Figure 4-23 The errors of slope related estimation results

4.7 Experiments and Discussion

The experiment was conducted in a real environment that consists of crossroad and corridors, as shown in Figure 4-24. Three boxes (marked as Box 1, 2, 3) are intentionally placed along the long and straight corridor to add more detectable features to the environment. There are two slopes and one edge in the experimental environment. Slope 1 is a downslope accompanied with an edge, while slope 2 is an upslope next to the stairs. Figure 4-25 gives a detailed view of these two slopes. The robot is controlled to move roughly along the green dashed line (sketch, not the exact route) and finally come back to the initial position as shown in Figure 4-24. There are a total of 700 steps conducted in this experiment. For each step, incremental encoder data, a horizontal scan, and a vertical scan are recorded.

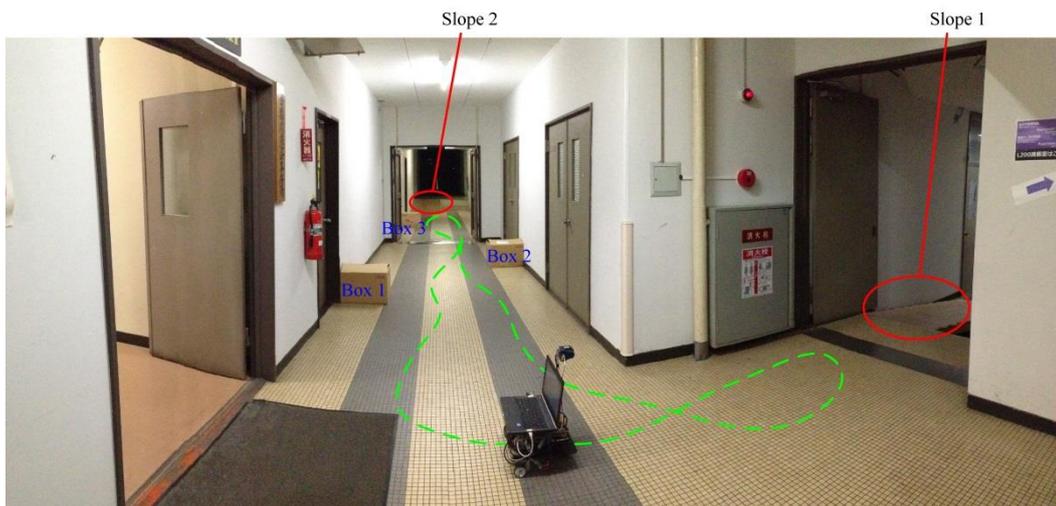


Figure 4-24 Experimental environment



(a) Downslope and edge

(b) Upslope and Stairs

Figure 4-25 One downslope, one edge and one upslope

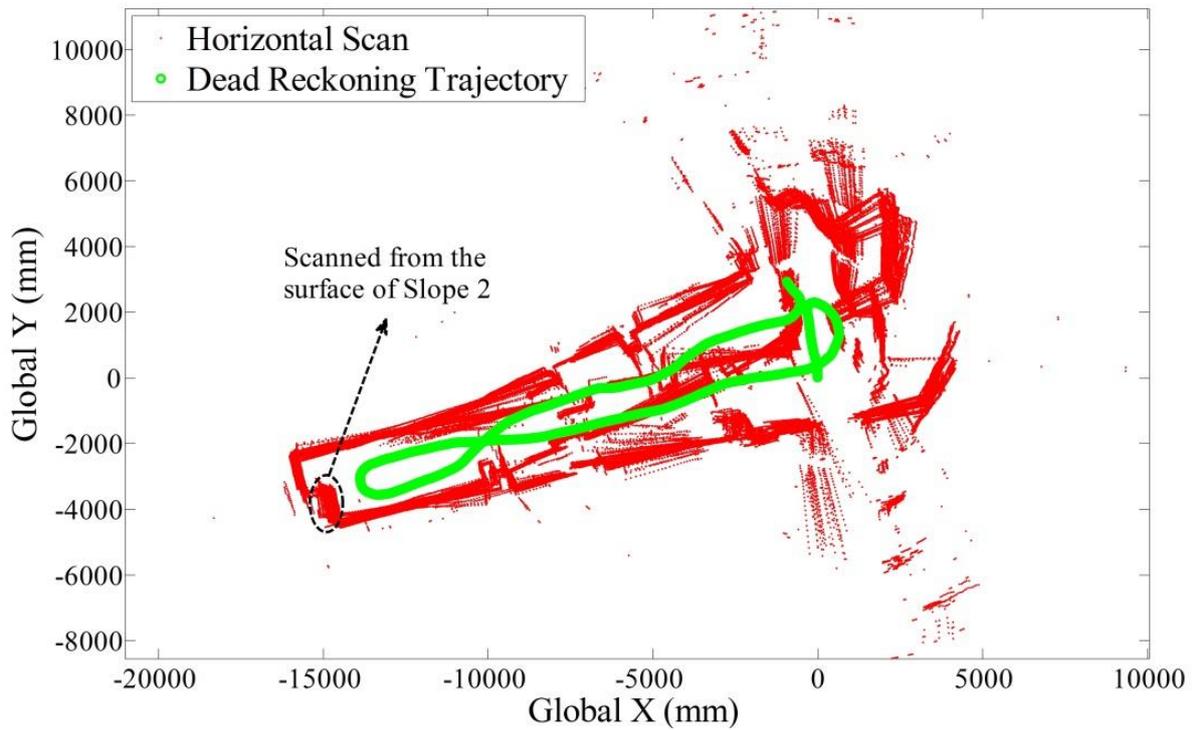


Figure 4-26 Horizontal scan registered by dead reckoning trajectory

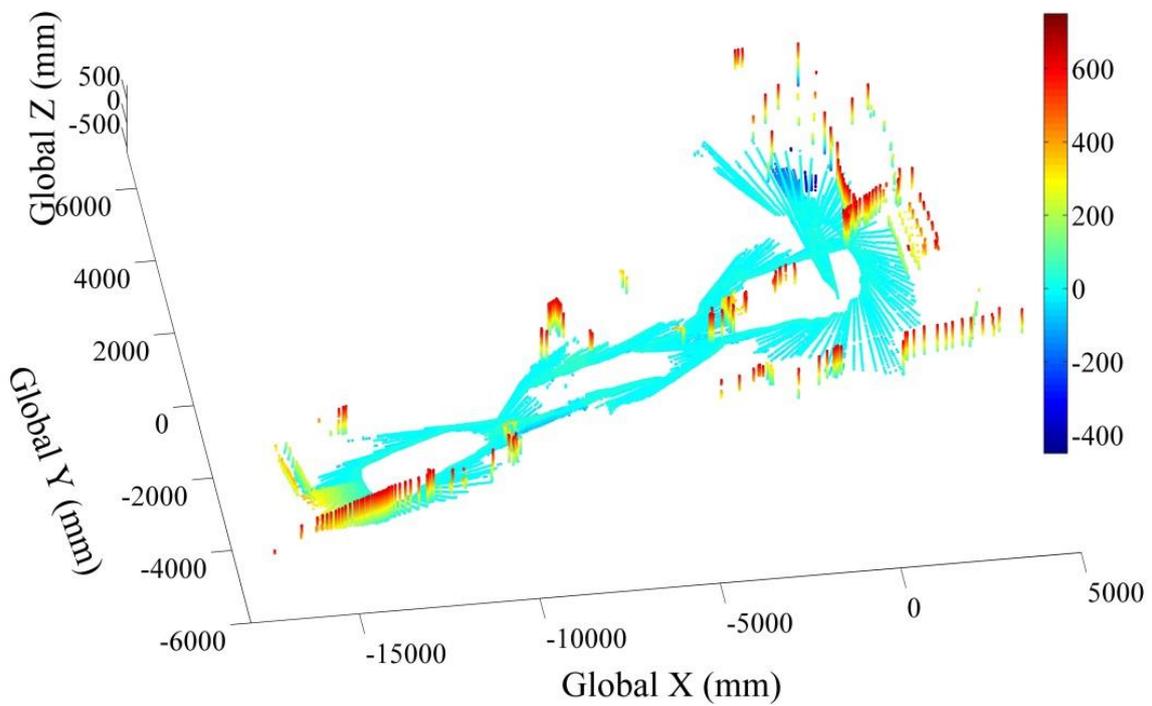


Figure 4-27 Vertical scan registered by dead reckoning trajectory

4.7.1 Experimental Raw Data

Figure 4-26 is plotted by utilizing the raw horizontal scanned data that is registered by the dead reckoning robot trajectory. The scanned data is severely corrupted because of the unbounded accumulating error of the dead reckoning pose. Notice that the data scanned from slope 2 is very noisy considering the small shift of the robot during the scanning on the slope 2. It is mainly due to inclined surface of slope 2 and rough ground in front of it. Figure 4-27 shows similar results of registered raw vertical scanned data. The area of slope 2 can be easily found out while the area of slope 1 is covered by the data scanned from the ground because of the huge reckoning error when robot returned to the initial position.

4.7.2 EKF-SLAM Results

Figure 4-28 shows the EKF-SLAM result when the robot finished the vertical scanning on slope 2 and turned back toward to the original position. Although the localization error has been obviously corrected comparing with the trajectory reckoned by odometer, it still have a certain amount of residual error which can be observed through the latest detected features in the left part of the figure. The final 2D EKF-SLAM result is shown in Figure 4-29. The features in the left part of the figure have been well corrected after the robot returned to the original position and re-observed the previous features. Overall, this 2D map gives a good top view of the environment.

However, as shown in Figure 4-29, an unexpected small crack can be found between the start line of slope 1 and the edge. Since the slope start line is extracted from the intersection points of the line-segment scanned from the slope and the ground line, this error is supposed to be mainly caused by the discontinuity between the slope 1 and the ground, as shown in Figure 4-30. As for the slope 2, a noisy horizontal line-segment scanned from the slope surface is recorded; this is mainly due to the imperfect flat ground. When the LRF1 scans on the inclined surface of the slope 2, the noise can easily be aggravated when the robot moves on the tile covered ground and vibrates slightly; some indication of this can be found in the scanned data pictured in Figure 4-26. The minor overestimated width of the slope 2 can be explained by the same reason.

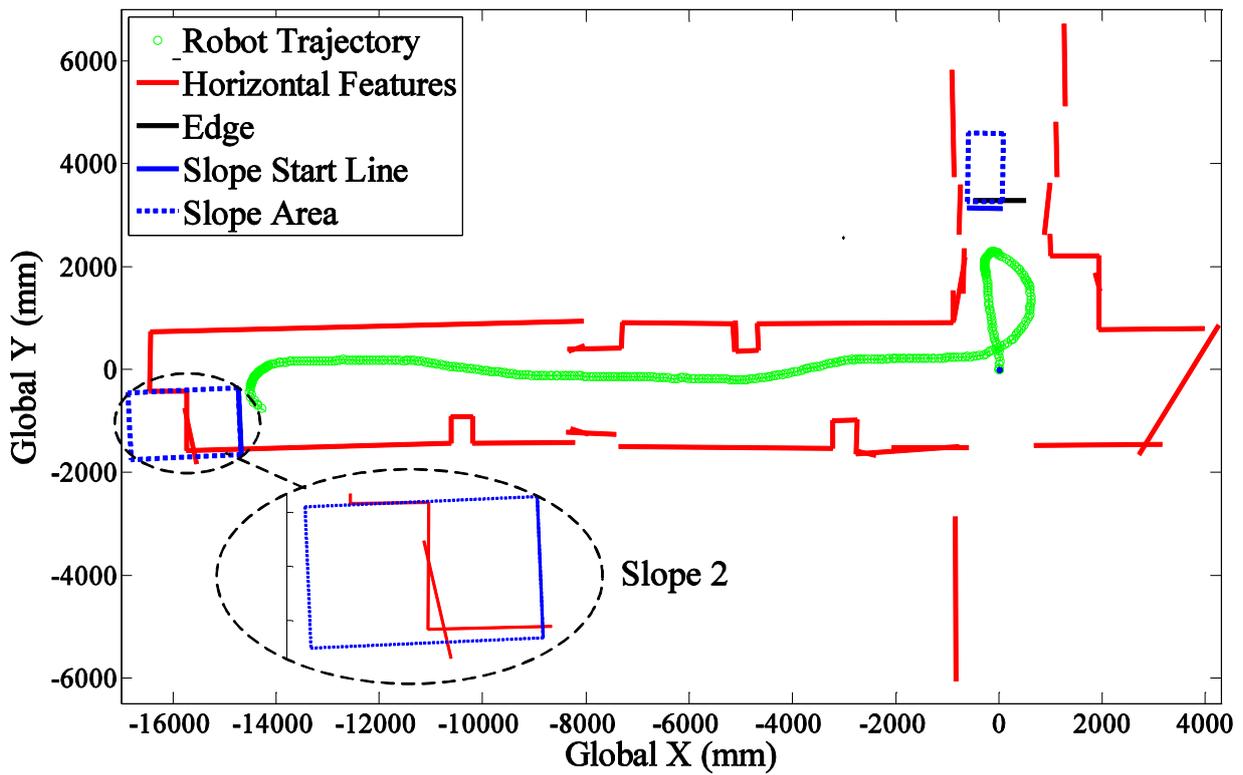


Figure 4-28 EKF-SLAM result in the halfway of the whole experiment

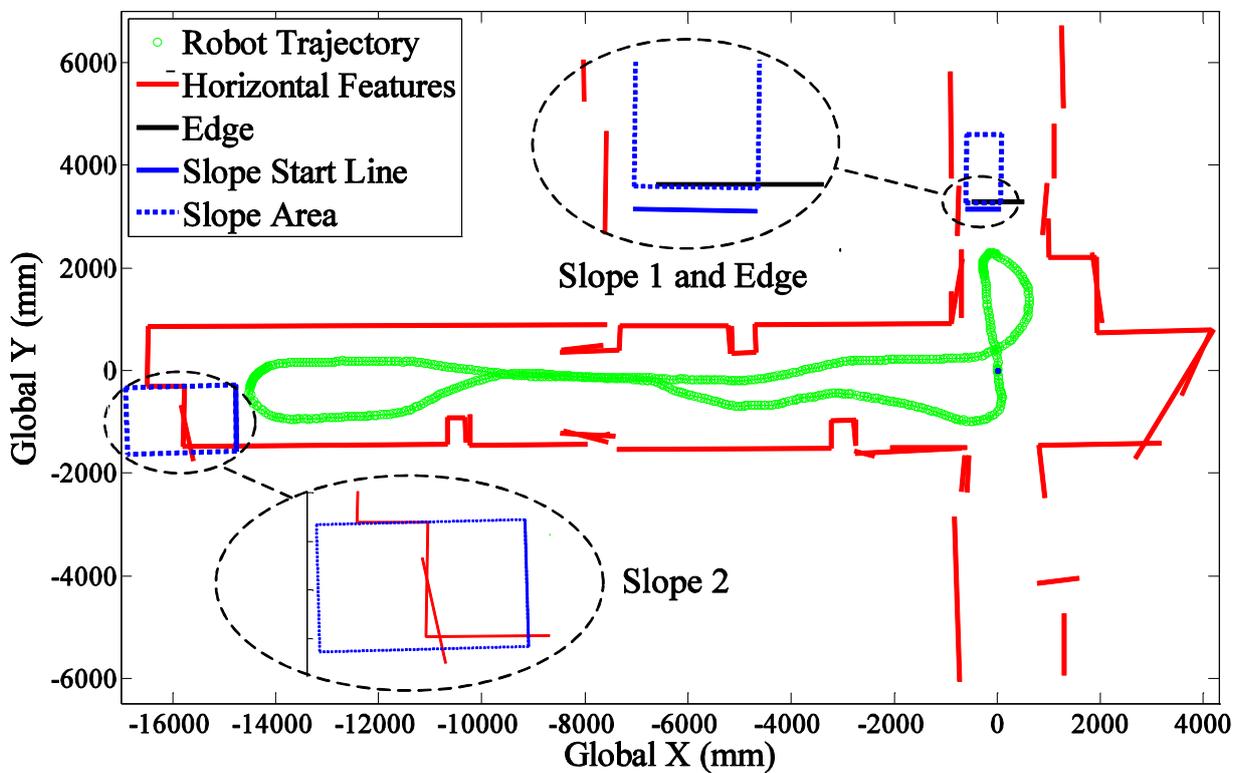


Figure 4-29 Final EKF-SLAM result with detected edge and slopes

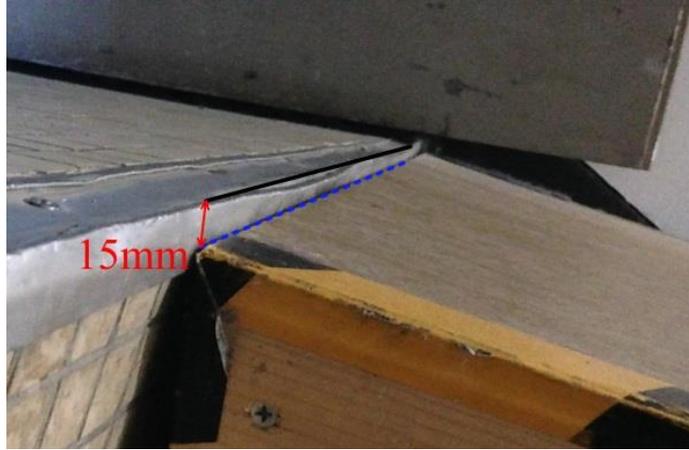


Figure 4-30 Discontinuity of the slope 1 with the ground

To verify the validity of the proposed algorithm, all the vital parameters of edge and slope that are estimated by the EKF-SLAM are compared with the reference value to calculate the error as follows:

$$Error = EKF_{result} - Reference. \quad (4-57)$$

Noted that the reference values cannot be exactly true since they are measured by author. The error of reference may be bigger than expected especially the reference of slope distance parameter $\rho_{s(1)}$ which is an indirect measurement due to the discontinuity as shown in Figure 4-30.

The changes in all the errors with respect to the time (step) are shown in Figure 4-31, Figure 4-32 and Figure 4-33, respectively. The errors of the orthogonal angles are shown in Figure 4-31. Angle $\alpha_{O(1)}$ indicates the orientation of the edge line and start line of the slope 1 while $\alpha_{O(2)}$ only represents the angular parameter of the start line of slope 2. Both angles are varying and corrected during the robot motion. Figure 4-32 shows the errors of distance parameters ρ_e , $\rho_{s(1)}$, and $\rho_{s(2)}$, which correspond to the edge, slope 1 and slope 2, respectively. The errors of $\alpha_{O(2)}$ and $\rho_{s(2)}$ diminish after step 460 when the robot moves towards the original position and re-observes the previous features. As for the gradient of the slopes, each of them is updated by an independent EKF so that they are corrected only when the corresponding slope is under observation, as shown in Figure 4-33.

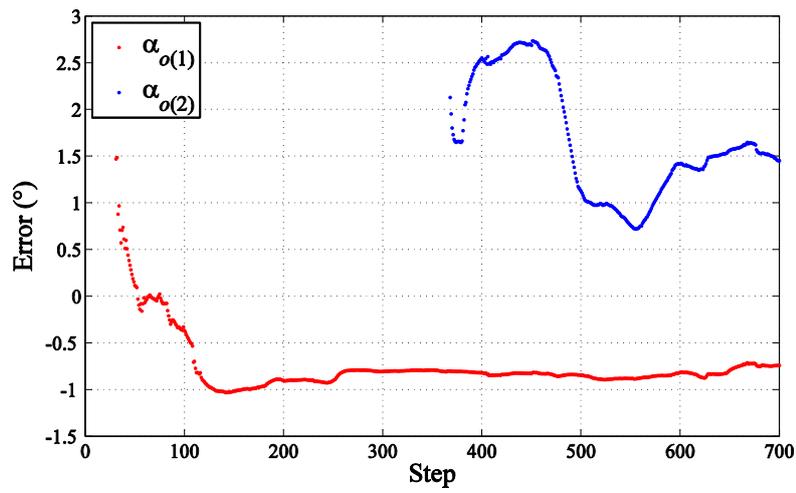


Figure 4-31 Error of orthogonal angles

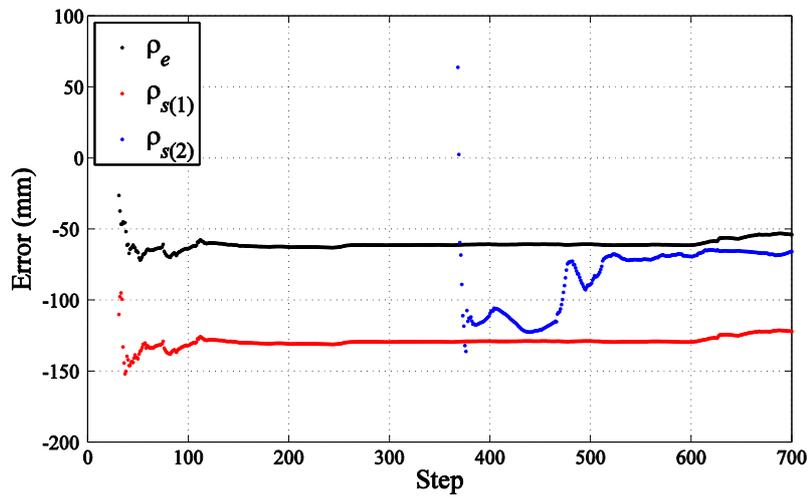


Figure 4-32 Error of distances parameters

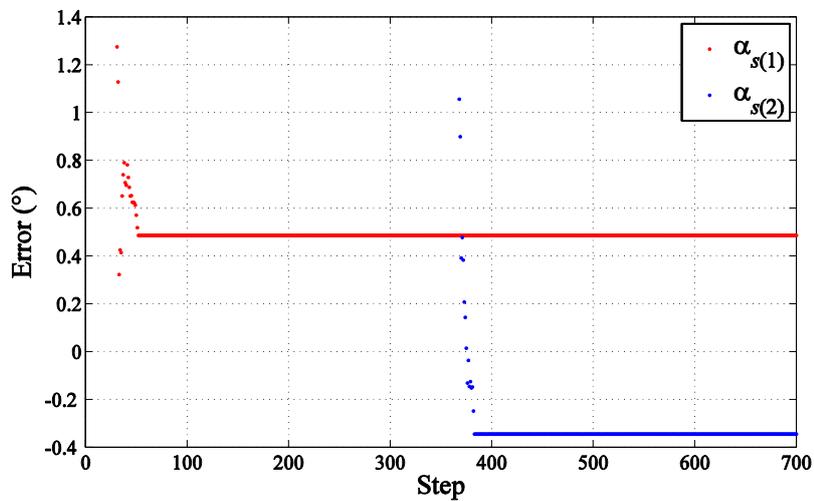


Figure 4-33 Error of slope angle

As the final SLAM result, the residual error of the orthogonal angle is mainly introduced by the error of local orthogonal angle estimation. The residual error is small but still slightly affects the orientation of the slope and edge. It can be found in the left bottom inset of Figure 4-29. The accuracy of distance parameter of slope and edge is mainly determined by the accuracy of orthogonal angle and the scanned slope line-segment that were extracted in the vertical scan. Considering the size of the explored environment, the residual errors of distance parameters are acceptable. Based on the distance parameter and corresponding orthogonal angle, the location of slope and edge can be determined in 2D map and easy for consultation. The error of the slope gradient is free from the localization error and is mostly introduced by the error of scanned slope line-segment and local orthogonal angle estimation. The estimated slope gradients are accurate enough for robot to find whether the slope is traversable.

4.7.3 Optional Prediction Results

To verify the efficiency of the optional prediction model, the error of the predicted transformation of the robot at each step should be presented. However, it is difficult to find the exact value of the robot pose transformation between two successive steps. Thus the robot trajectory of the EKF-SLAM is adopted as the reference route and the transformation between the two steps can be calculated. The error can then be calculated as follows:

$$Error = Prediction_Trans. - EKF_Trans. \quad (4-58)$$

The optional prediction is executed in this EKF-SLAM. And out of a total of 700 steps, the ICP based prediction is adopted in 174 steps while the remaining 526 steps adopt dead reckoning based prediction.

The pose transformation is composed of translation and rotation. The translational error is usually small since the robot cannot move at a high speed due to security considerations. Figure 4-34 and Figure 4-35 show plots of translational errors along X_R direction and Y_R direction at each step, respectively. Overall, the error along the Y_R is bigger than errors along X_R since the robot was moving forward instead of turning most of the time. However, no big difference can be found between the errors of two prediction models. Compared with the translational error, the rotational error is more critical because

it normally leads to a very big error when the range of scanned data increases, which may lead to the failure of features association. The fatal rotational errors are substantially reduced by optional prediction, as shown in Figure 4-36.

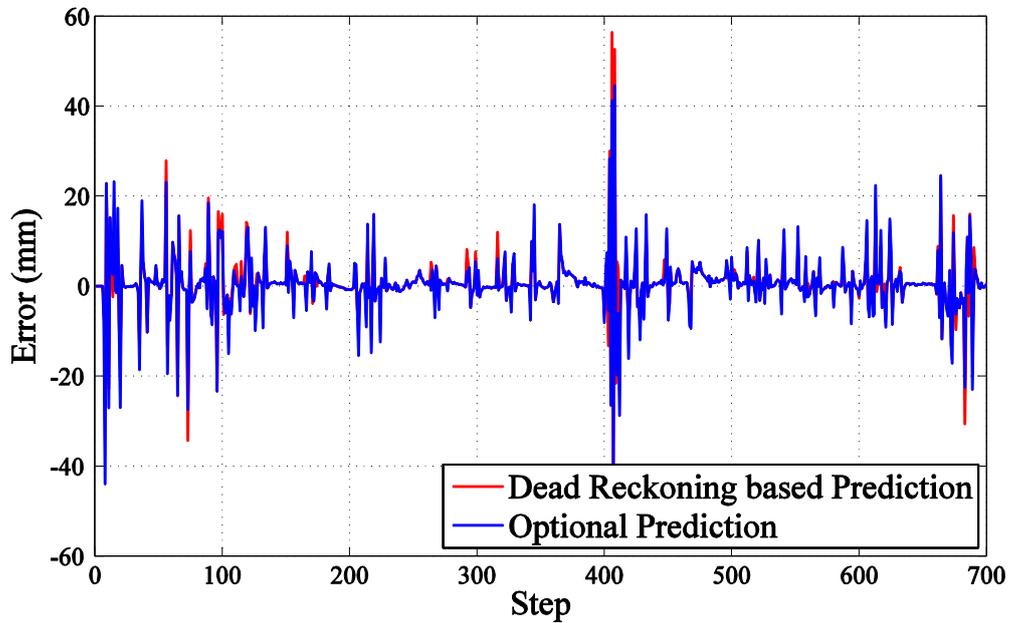


Figure 4-34 Comparison of translational error along X_R

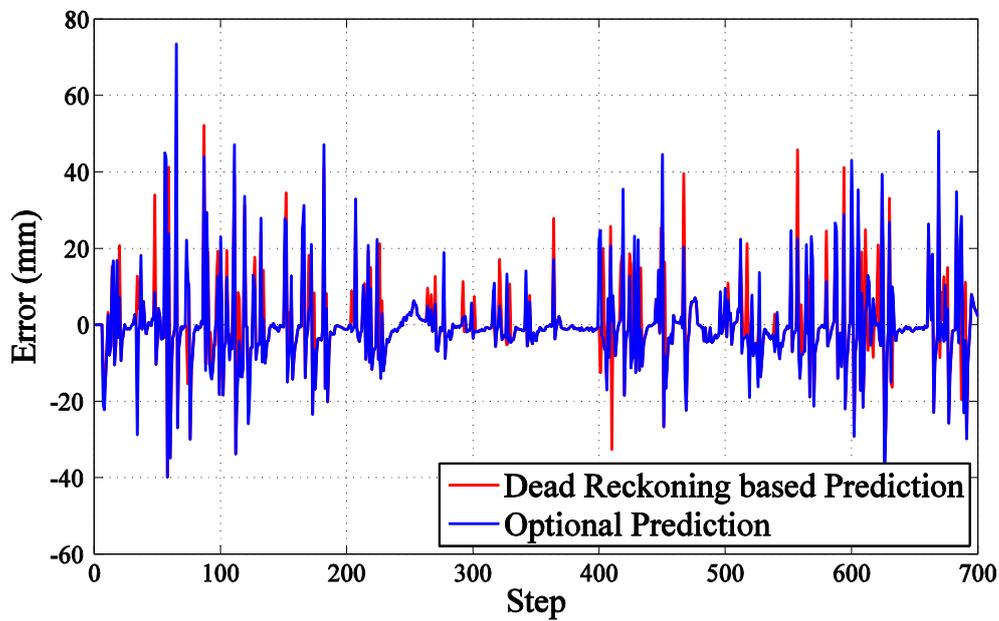


Figure 4-35 Comparison of translational error along Y_R

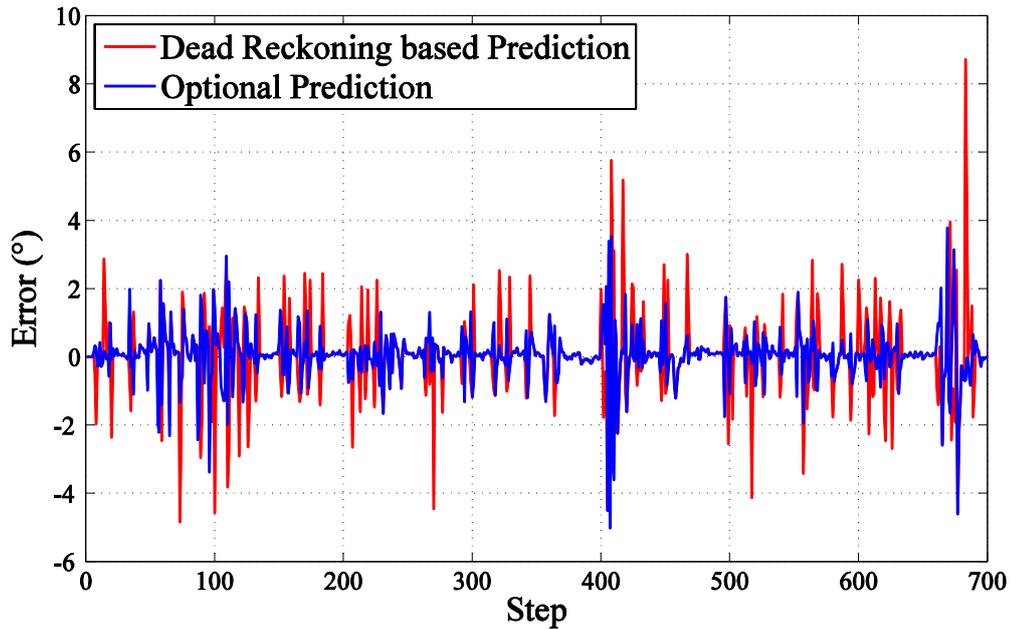


Figure 4-36 Comparison of rotational errors

The statistical results of the error properties are listed in Table 4-1 and Table 4-2. All the means of errors listed in Table 4-1 are close to zero. Table 4-2 lists out the standard deviation of errors which can act as the judgment of performance between two prediction models. Optional prediction shows slightly bad results in translation. It is usually due to the scan error caused by LRF itself and slight vibration when the robot moving on the tiled floor. However, the translational distance between each step is usually over 50mm, thus these differences between two prediction models can be neglected. On the contrary, optional prediction gives a superior performance in the rotational error correction by reducing the rotational error over 32.7%. And it makes the prediction more accurate, therefore the feature association of EKF-SLAM will be more robust.

Table 4-1 Means of translational and rotational errors

	X(mm)	Y(mm)	Angle(°)
Dead Reckoning	0.6751	-0.2111	0.0459
Optional	0.3692	-0.3479	0.0337

Table 4-2 Standard deviations of translational and rotational errors

	X(mm)	Y(mm)	Angle(°)
Dead Reckoning	6.0679	10.5175	1.1434
Optional	6.6889	11.2768	0.7687

4.7.4 Computational Cost

It is well-known that the computational cost of EKF-SLAM is $O(n^2)$ per step where n is the number of stored feature [89]. Since 3D feature has at least 3 parameters while 2D feature only has 2 parameters, 3D EKF-SLAM theoretically requires 9/4 times computational cost per step as much as 2D EKF-SLAM in the case of they have the same feature number. Another extra computational cost of 3D SLAM comes from the feature extraction. To represent structured indoor environment, constrained plane and line-segment are the preferred features of 3D SLAM [90] and 2D SLAM, respectively. The extraction of constrained plane from spatial point cloud requires much more computational cost than line-segment which is extracted from fewer planar points.

Figure 4-37 shows the computational time of the proposed 2D algorithm at each step during the experiment. Overall, the time cost of each step is low. But it keeps increasing along the experiment process because more and more features are detected and added into EKF-SLAM. In general, the step that adopts ICP based prediction model requires higher computational cost than the step that adopts dead reckoning based prediction. Obviously, the application of optional prediction successfully reduces the computational cost by selecting the appropriate prediction model. A few of heavy costs, which require much higher computational time than the neighboring steps, can be found. This is because the bad dead reckoning results require more iterations of ICP to compensate the errors, especially when the rotational error is huge.

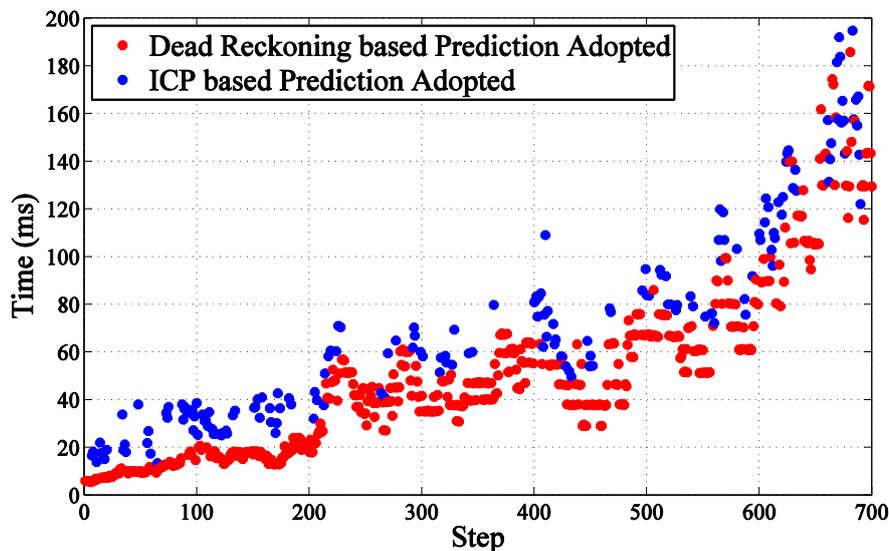


Figure 4-37 Computational time of each step

4.8 Conclusion

In this chapter, a computationally inexpensive algorithm has been proposed to detect the slope and edge in a structured indoor environment. This algorithm is mainly composed of 2D line-segments based EKF-SLAM and orthogonal assumption. Instead of introducing an additional mechanism to rotate LRF to get 3D points clouds, a vertically fixed LRF framework has been adopted to enable the robot to fulfill the horizontal scan and vertical scan. Also, an optional prediction model has been proposed to select the suitable model from dead reckoning based prediction and ICP based prediction, which saves the computational cost and ensures the robustness of the prediction result. In the simulation, SLAM results obtained by proposed algorithm are compared with the ground truth to estimate the errors. The negligible errors verified the efficiency of the proposed algorithm. The following experiment was conducted in a real indoor environment where there are edge, downslope, and upslope features. Although the SLAM results are still satisfactory, they are not as accurate as what obtained in the simulation. The errors have been analyzed and main reasons have been explained.

Chapter 5 IMU Calibration with Scan Matching

5.1 Introduction

Although some high grade MEMS-IMUs have been precisely calibrated and have no requirements for further calibration, most of the consumer grade MEMS-IMUs have been poorly calibrated for saving the cost on calibration. Calibration is defined as the process of comparing instrument outputs with known reference information, and determining coefficients that force the outputs to agree with the reference over a range of output values [91]. Local gravity is a good reference for accelerometer calibration since it is constant, stable and obviously over the resolution of sensor. In the case of gyroscope, the naturally existing Earth's rate is inapplicable for MEMS-IMU because Earth rotation rate is under the resolution of this kind low cost sensor. Classical method for MEMS-IMU calibration is performed using precise biaxial or tri-axial turntable that rotates the MEMS-IMU to different orientations. The readings of accelerometer when MEMS-IMU is static are compared with local gravity, while measurements of gyroscope during the rotation are compared with known rotation rate of the turntable [92]. This method needs costly and high-precision equipment that may not be available to all researchers.

In this chapter, the configuration of the MEMS-IMU that is adopted in this study is firstly presented. Different types of sensor errors are introduced and static test conduct on this IMU sensor is presented. Based on the analyzed sensor errors, a uniform sensor error model of accelerometers and gyroscopes is adopted. Secondly, straightforward calibration process is proposed to compress the error of the MEMS-IMU, which adopts local gravity as the reference of accelerometer's calibration and scan matching based angular displacement as the reference of gyroscope's calibration, respectively. The comparison between uncalibrated and calibrated measurements is conducted. Finally, the IMU based gait tracking is applied on the IMU to verify the validity of proposed calibration work.

5.2 IMU and Error

5.2.1 Consumer Grade IMU



Figure 5-1 MEMS-IMU MPU6050 (Model GY521)

MPU6050 (Model GY521), which is a popular applied consumer grade MEMS-IMU, is utilized in this study, as shown in Figure 5-1. It is very cost-efficient given the fact that it combines both tri-axial accelerometer and tri-axial gyroscope. And it contains 16-bits analog to digital conversion hardware for each channel. Therefore it captures x, y, and z channel at the same time. The sensor uses the I2C-bus to interface with the Arduino. Table 5-1 gives the scale ranges and sensitivities of the merged gyroscope and accelerometer. There are four levels and the sensitivity decreases with the increment of scale range. This study is focused on the indoor mobile robot navigation, in which scenario the acceleration and rotation are limited. Thus the IMU is set up to the first level which limits the output of gyroscope up to 250 degree per second and the output of accelerometer up to 2 times gravity.

Table 5-1 Scale range and sensitivity of MPU6050

<i>Gyroscope</i> Full Scale Range (°/sec)	<i>Gyroscope</i> Sensitivity (LSB/°/sec)	<i>Accelerometer</i> Full Scale Range (g)	<i>Accelerometer</i> Sensitivity (LSB/g)
±250	131	±2	16384
±500	65.5	±4	8192
±1000	32.8	±8	4096
±2000	16.4	±16	2048

5.2.2 Survey on Sensor Errors

The performance characteristics of inertial sensors are affected by various errors [93]. Most of these errors can be classified into biases, scaling factors, axial non-orthogonality, frame misalignment and noise.

Biases and Scaling Factor

The average readings of IMU when there is no input of acceleration or rotation over a specified time are regarded as the biases of corresponding sensors. The biases of gyroscope are presented in degree per second and the biases of accelerometers are expressed in meter per square second. Bias consists of a stationary part named bias offset and a random varying part named as bias drift. The bias drift is random in nature and can only be modeled by stochastic process. And another characteristic of bias is called bias instability, which refers to the random variation in the bias as computed over specified finite sample time, and, averaging time intervals.

Scaling factor is the ratio between the change of measurement and the change of real input, as shown in Figure 5-2 (b). Scaling factor can be regarded as the slope of the straight line that projects the input data to output readings.

Bias and scaling factor can be asymmetric, which means that the scaling factor is different when the sensor is measuring with positive input and negative input. Furthermore, bias and scaling factor are usually sensitive to thermal conditions for the fact that they are prone to drift with the change of temperature [94].

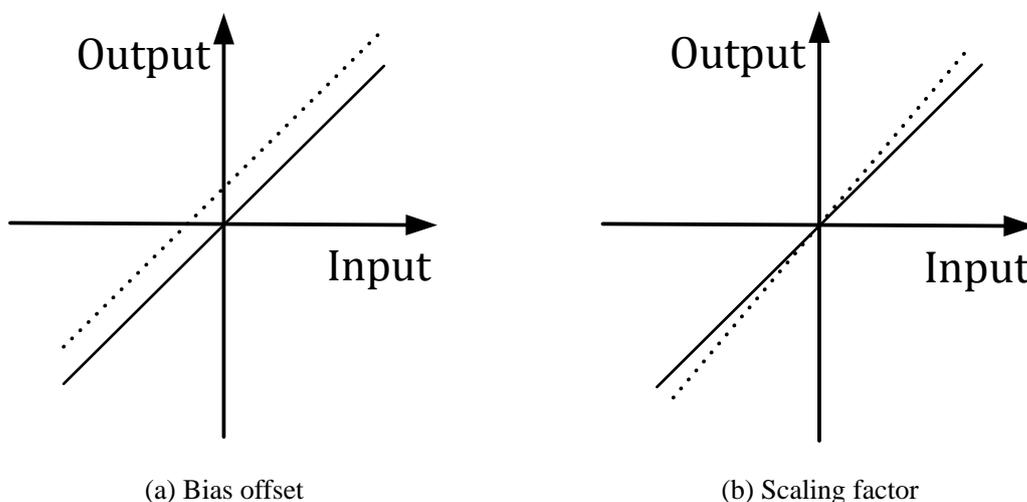


Figure 5-2 Bias offset and scaling factor

Non-orthogonality

Non-orthogonality is the error resulting from the procedure of imperfectly assembling individual sensors on the IMU chip. This phenomenon can be explained by using Figure 5-3 (a), the real axes of sensors are represented by using non-orthogonal frame $X_{S_r} Y_{S_r} Z_{S_r}$ while sensor frame $O-X_S Y_S Z_S$ is defined as an ideal orthogonal coordinate frame, and its X-axis is coincidence with the axis X_{S_r} . The small angular errors φ_{yx} , φ_{zx} , φ_{zy} have to be estimated so that the non-orthogonal tri-axial measurements can be correctly project to the orthogonal output readings with respect to the sensor frame.

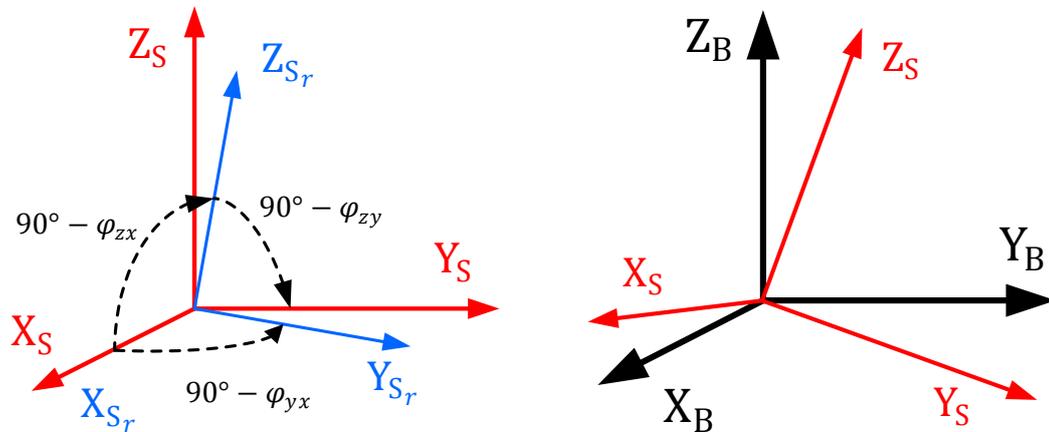
Misalignment

Misalignment describes the existence of angular error between sensor frame $O-X_S Y_S Z_S$ and the extrinsic carrying body frame $O-X_B Y_B Z_B$ such as the frame of IMU body or mobile robot, as shown in Figure 5-3 (b).

This error commonly exists and it generates from two main sources:

- a) Manufacture: Errors of packaging individual sensors on the IMU body.
- b) Installation: Errors of installing IMU chip on extrinsic carrying body.

Because of the existence of installation, misalignment error needs to be calibrated whenever the IMU sensor is installed into a new extrinsic body, namely, when the body frame has been changed. Since body frame and sensor frame are both regarded as ideal orthogonal, a small angle rotation matrix that can project readings from sensor frame to body frame is required to be estimated.



(a) Non-orthogonality inner sensor frame (b) Misalignment between sensor and body frame

Figure 5-3 Non-orthogonality and misalignment

Noise

Noise is the signal that carries no useful information and it is unavoidable for any sensors for they are random and unpredictable even if sensors are not interfering with any signals [95]. It is resulting from the sensor itself or other electronic interfacing equipment. Noise is non-systematic and therefore cannot be calibrated from the data using deterministic models. It can only be modeled by stochastic process.

5.2.3 Sensor Error Model

To keep the calibration procedure straightforward, the scaling factors and biases of gyroscope and accelerometer are treated as symmetric in this study. And the effect of temperature is neglected due to the fact that this calibrated MEMS-IMU is intended to be used on indoor mobile robot where temperature varies little. Besides, the thermal calibration usually needs thermal chamber to give accurate reference of temperature, which may not be available in all robotic research groups.

According to the errors described in the last sub-section, a uniform sensor error model (SEM) for gyroscope and accelerometer calibration is defined as

$$\mathbf{v}_B = \mathbf{MNS}(\mathbf{u} - \mathbf{b}), \quad (5-1)$$

where \mathbf{u} is the vector of sensors raw tri-axial readings and \mathbf{b} is the vector of corresponding biases

$$\mathbf{u} - \mathbf{b} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} - \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}. \quad (5-2)$$

\mathbf{S} stands for the scaling factors matrix

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}. \quad (5-3)$$

\mathbf{N} is the transformation matrix to transform triads' readings from non-orthogonal frame to orthogonal sensor frame defined in Figure 5-3 (a). And

$$\mathbf{N} = \begin{bmatrix} 1 & 0 & 0 \\ \sin\varphi_{yx} & \cos\varphi_{yx} & 0 \\ \sin\varphi_{zx} & \sin\varphi_{zy}\cos\varphi_{zx} & \cos\varphi_{zx}\cos\varphi_{zy} \end{bmatrix}, \quad (5-4)$$

where φ_{zx} , φ_{zy} and φ_{zy} are the small error angles. The detail procedure of deduction from three angles to transformation matrix can be found in [96]. By using small angle approximation, \mathbf{N} can be simplified to

$$\mathbf{N} = \begin{bmatrix} 1 & 0 & 0 \\ \varphi_{yx} & 1 & 0 \\ \varphi_{zx} & \varphi_{zy} & 1 \end{bmatrix}. \quad (5-5)$$

\mathbf{M} presents the rotation matrix to compensate misalignment between sensor frame $O-X_S Y_S Z_S$ and body frame $O-X_B Y_B Z_B$, as shown in Figure 5-3 (b).

$$\mathbf{M} = \begin{bmatrix} c_\beta c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ c_\beta s_\gamma & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma \\ -s_\beta & s_\alpha c_\beta & c_\alpha c_\beta \end{bmatrix}, \quad (5-6)$$

where α , β , γ are roll, pitch, yaw angle respectively to describe the orientation of sensor frame with respect to body frame, and $c_\alpha = \cos\alpha$, $s_\alpha = \sin\alpha$, etc. With small angle approximation, the rotation matrix \mathbf{M} can be simplified to

$$\mathbf{M} = \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix}. \quad (5-7)$$

And last, \mathbf{v}_B is the vector of calibrated readings with regard to body frame

$$\mathbf{v}_B = \begin{bmatrix} v_{x_B} \\ v_{y_B} \\ v_{z_B} \end{bmatrix}. \quad (5-8)$$

The SEM described in Equation (5-1) now can be detailed as

$$\begin{bmatrix} v_{x_B} \\ v_{y_B} \\ v_{z_B} \end{bmatrix} = \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \varphi_{yx} & 1 & 0 \\ \varphi_{zx} & \varphi_{zy} & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \left(\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} - \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \right). \quad (5-9)$$

Therefore there are in total 12 calibration parameters that need to be estimated for both gyroscopes and accelerometers. They are biases b_x , b_y , b_z , scaling factors s_x , s_y , s_z , non-orthogonal error angles φ_{yx} , φ_{zx} , φ_{zy} and misalignment angles α , β , γ .

5.3 IMU Calibration Approach

Because of the wide application of MEMS-IMU, various calibration methods that do not need precise turntable have been proposed [97][98]. The calibration algorithm applied in this study is an extension of the approach that is proposed in [99]. In the original approach, the authors firstly use multi-position method to calibrate accelerometer and then calibrate gyroscope by using a manually driven turntable with the angular reference obtained from Fiber Optic Gyroscope (FOG) based measurement system. Comparing with the original approach, four improvements can be found in our new approach and they are listed as follows:

- a) First, instead of using two different sensor error models for accelerometer and gyroscope, this approach adopts uniform sensor error model for calibration work.
- b) Second, recently popular 3D printers [100] are introduced to produce a satisfactory plastic cube which makes it possible to align the frame of gyroscope and accelerometer to the uniform body frame $O-X_B Y_B Z_B$, as shown in Figure 5-4.
- c) Third, instead of utilizing FOG based measurement system, the proposed scan matching method is applied to laser scans to provide accurate estimation on rotational angle for gyroscope calibration.
- d) Fourth, instead of applying specialized turntable, a commonly available swivel chair is reformed to be a simple turntable to perform vibration-free rotation.

These improvements make the method become more feasible for most robotic researchers since scanning sensor is the necessity of autonomous robot and swivel chair can be found anywhere.

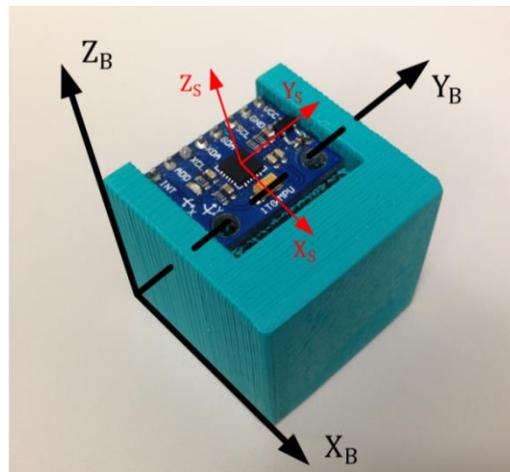


Figure 5-4 Body frame is aligned with the edges of the cube made by 3D printer

5.3.1 Static Raw Readings

Figure 5-5 and Figure 5-6 show the raw readings of IMU when it is statically placed on table. The readings of accelerometer plotted in Figure 5-5 are very noisy and probably biased. The gyroscope readings plotted in Figure 5-6 are also noisy and biased. The biases of gyroscope can be easily figured out since the ground truth of rotations should be all zero when the IMU is static. Comparing with the straightforward estimation on biases existence of gyroscopes' readings, it is more difficult to say that there are significant biases in the measurements of accelerations since the IMU is placed on the table surface that may not perfectly orthogonal to the direction of gravity.

Allan Variance

Allan variance, which is firstly proposed to investigate the frequency stability of precision oscillators, can be adapted to random drift characterization of IMU [101]. It can provide information on the types and magnitude of various noise terms by measuring the variance of the difference between consecutive interval averages. The Allan variance $\sigma_a^2(\tau)$ is defined as:

$$\sigma_a^2(\tau) = \frac{1}{2K} \sum_{k=1}^K (\bar{\mathbf{u}}(\tau, k+1) - \bar{\mathbf{u}}(\tau, k))^2 \quad (5-10)$$

where τ is the time span of one interval, K is the total number of independent intervals that segmented from the whole readings, $\bar{\mathbf{u}}(\tau, k)$ is the average of readings in k th interval. In this study, the time span τ is set up to vary from 1s to 1000s. As shown in Figure 5-7 and Figure 5-8, the effect of bias drift oscillation is converge after the average time interval is over 50s.

By dealing with the average of measurements over an interval of time determined using the Allan variance analysis above, the bias drift will be kept minimal during the successive time interval when the readings are collected for calibration. Thus, the only noise term need to be considered during the calibration process is white noise.

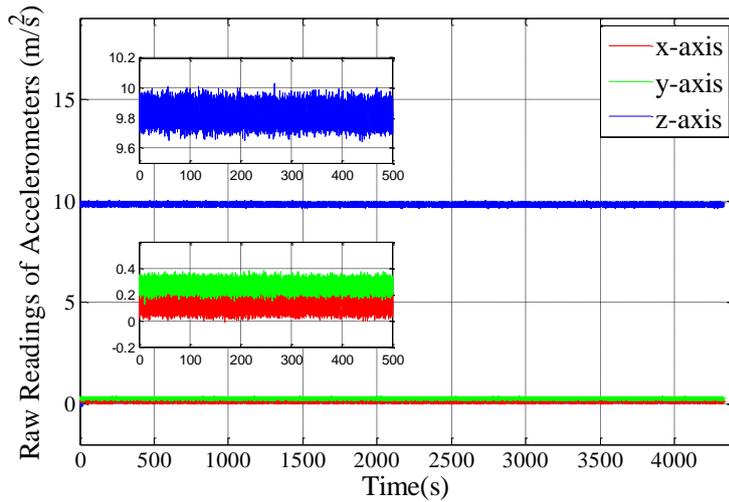


Figure 5-5 Accelerometer raw readings of statically placed MPU6050 IMU

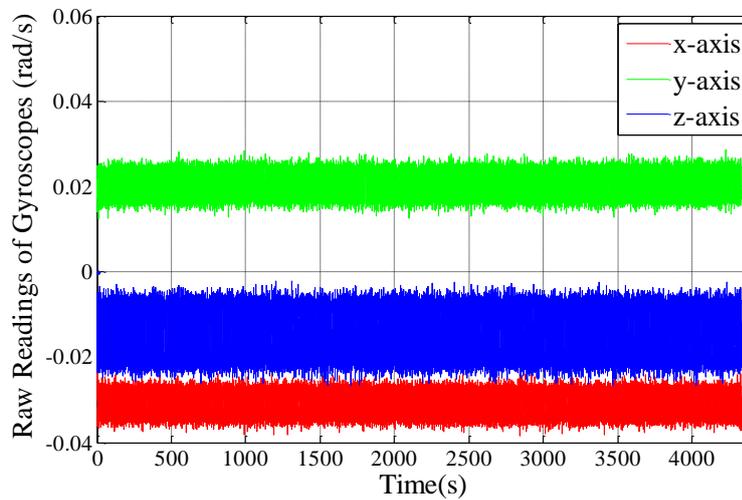


Figure 5-6 Gyroscope raw reading of statically placed MPU6050 IMU

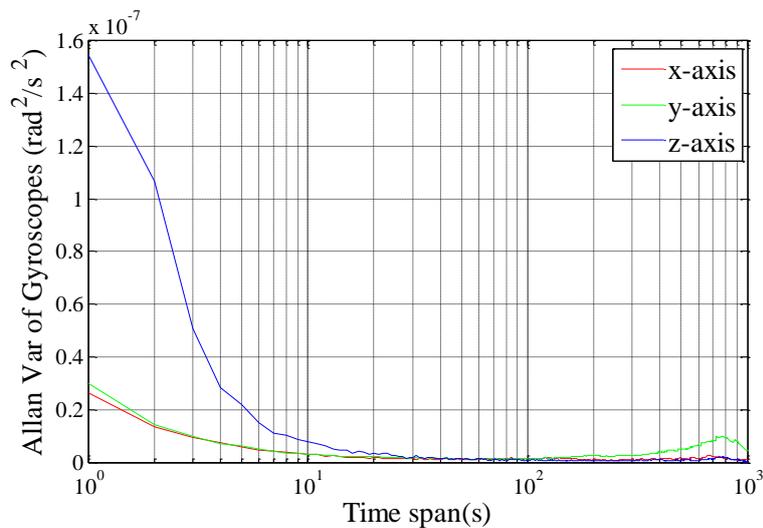


Figure 5-7 Allan Variance of Gyroscopes

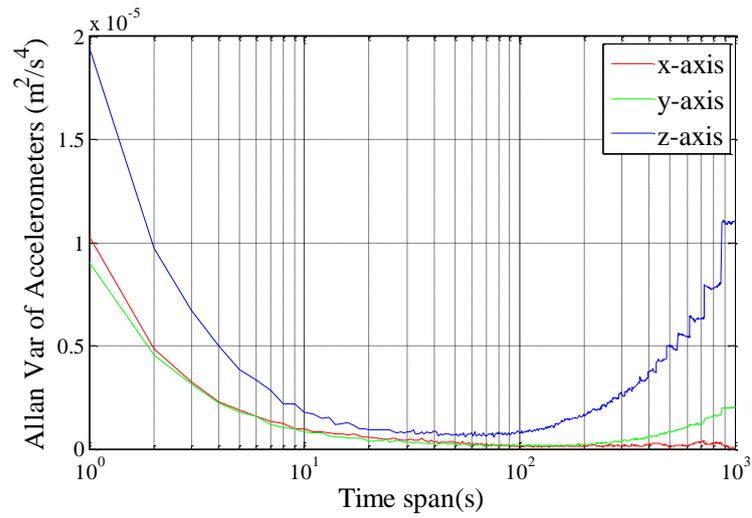


Figure 5-8 Allan Variance of Accelerometers

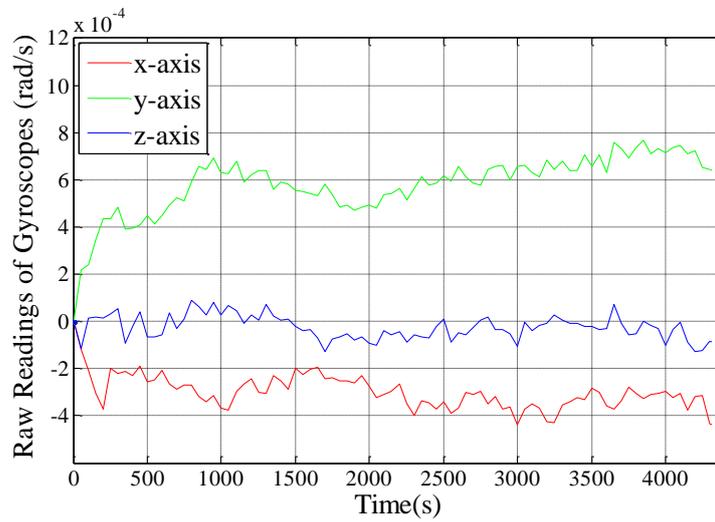


Figure 5-9 Biases Drift of Gyroscopes Over time

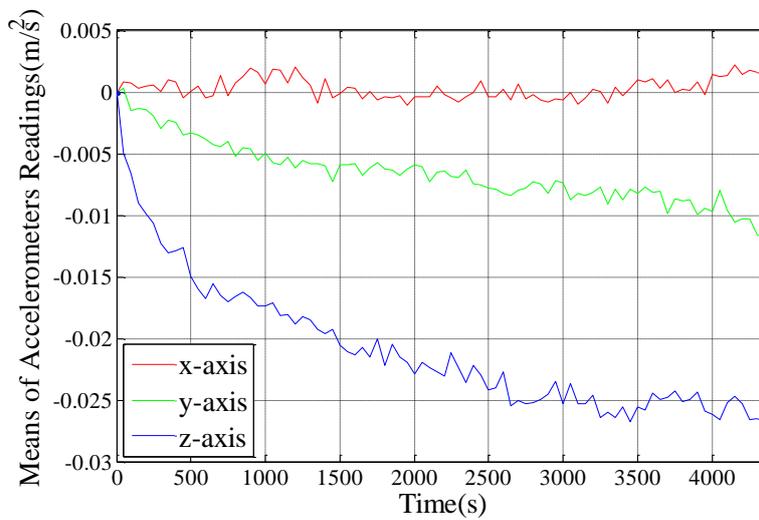


Figure 5-10 Means Drift of Accelerometers Over time

Figure 5-9 plots the biases drifting of gyroscope sensors when IMU is statically placed. The values of biases are obtained by average raw readings over time interval of 50s as implied by Allan variance. All the average value of the biases is then subtracted by corresponding average value of first interval so that the drift history can be clearly presented. Figure 5-10 shows the means of accelerometer readings. These two figures clearly demonstrate that the readings drift fast at the initial heat-up stage for the biases and scaling factors are affected by the increasing temperature.

5.3.2 Accelerometer Calibration

The accelerometer calibration in this study consists of two processes. The first part is multi-position based calibration which has been detailed in [99]. This process can estimate most of the calibration parameters except misalignment angle. The second part is our proposed misalignment correction method which is similar to first process.

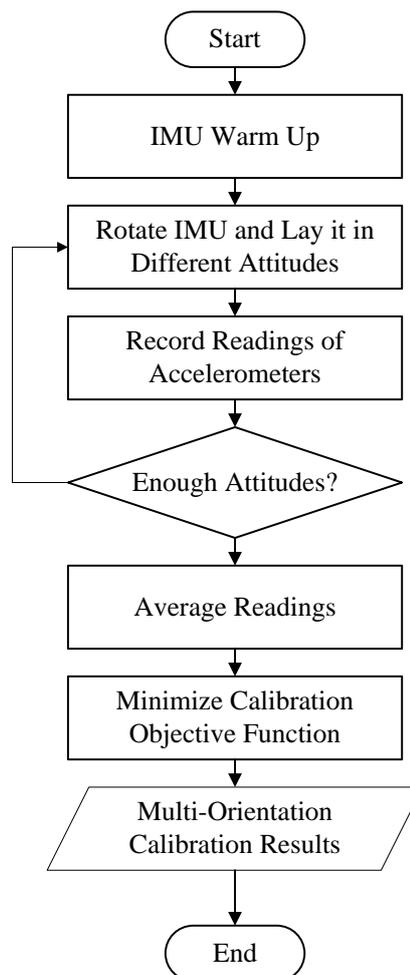


Figure 5-11 Flowchart of accelerometer Multi-position based Calibration

Multi-position based Calibration

The basic concept of this method is to place IMU in various orientations stationary so that local gravity can be treated as the only input. Then the readings of accelerometer in every orientation are recorded and the magnitudes of readings are compared with the magnitude of local gravity. The related calibration parameters are estimated by minimizing the magnitude error with quasi-newton nonlinear optimization algorithm.

The flowchart of calibration has been plotted in Figure 5-11. The detail practical operation is summarized as follows:

- a) Connect IMU to Arduino board with power access to get warm up. According to the mean drift plotted in Figure 5-10, the warm up should keep more than at least 10 minutes so that the swift bias drift introduced by thermal effect can be suppressed.
- b) Rotate the Cube and place it stationary, record the readings of accelerometers.
- c) Repeat last step to make sure that that the vector of gravity is approximately overspread the sphere that is centered in IMU. In our study, readings in 30 different orientations have been recorded. Total 30 Orientations include 8 orientation along each axis, as show in Figure 5-12, and 6 orientations that main face of cube towards down. No accurate knowledge about these particular orientations is required since the error angles of these orientations have no effect on the magnitude of measurements, which makes this step simple and feasible without precise turntable.
- d) Average the tri-axial readings of accelerometer in each orientation.
- e) Estimate calibration parameters by solving minimization problem as follows

$$(\hat{\mathbf{N}}^{acc}, \hat{\mathbf{S}}^{acc}, \hat{\mathbf{b}}^{acc}) = \operatorname{argmin} \sum (|\mathbf{G}| - |\bar{\mathbf{v}}_S^{acc}(i)|)^2, \quad (5-11)$$

where $|\mathbf{G}|$ is the magnitude of local gravity. $\bar{\mathbf{v}}_S^{acc}(i)$ is the average measurements w.r.t sensor frame and is obtained from i -th orientation and corrected by related calibration parameters as

$$\bar{\mathbf{v}}_S^{acc}(i) = \mathbf{N}^{acc} \mathbf{S}^{acc} (\bar{\mathbf{u}}^{acc}(i) - \mathbf{b}^{acc}), \quad (5-12)$$

where $\bar{\mathbf{u}}^{acc}(i)$ is the average of raw accelerometer readings that is obtained from i -th orientation.

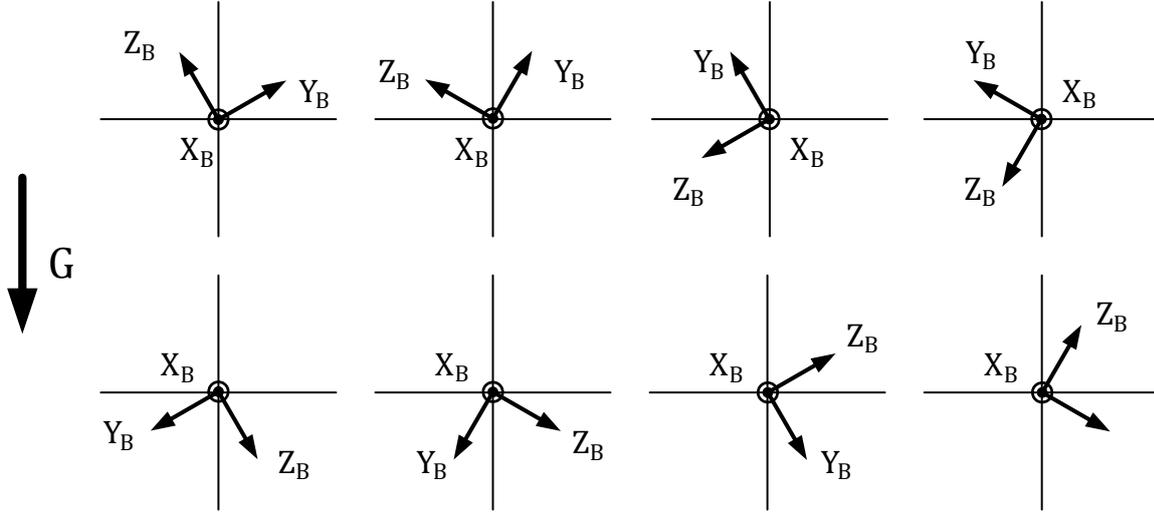


Figure 5-12 Rotated along X_B , 8 orientations with respect to local Gravity

In real application, this minimization problem is solved by using Matlab Optimization Toolbox™ with the initial guess of calibration parameters as follows

$$\varphi_{yx}^{acc} = 0, \varphi_{zx}^{acc} = 0, \varphi_{zy}^{acc} = 0;$$

$$s_x^{acc} = 1, s_y^{acc} = 1, s_z^{acc} = 1;$$

$$b_x^{acc} = 0, b_y^{acc} = 0, b_z^{acc} = 0. \quad (5-13)$$

By using multi-position based calibration approach, bias, scaling factor and non-orthogonal angles can be estimated while misalignment error cannot be eliminated since it is a rotation matrix and is independent from the magnitude of measurements, as explained in following equations

$$\mathbf{v}_B^{acc} = \mathbf{M}^{acc} \mathbf{v}_S^{acc}, \quad |\mathbf{v}_B^{acc}| = |\mathbf{v}_S^{acc}|. \quad (5-14)$$

Misalignment Correction

Though multi-position based calibration approach is straightforward and efficient, it cannot eliminate the misalignment error between sensor frame and body frame. Benefiting from the introduction of the cube made by 3D printer, an easy approach to estimate misalignment error is proposed in this study. The flowchart of this approach is plotted in Figure 5-13.

Assume a square tray that is made by 3D printer, which can perfectly hold the cube, is fixed on the table, as shown in Figure 5-14. The surface of the table can be regarded as a plane but is not perfectly orthogonal with the orientation of local gravity. As a consequence, the Z_T axis of tray frame $O-X_T Y_T Z_T$ should not perfectly align with the direction of local gravity as well. Let local gravity $\mathbf{G} = [0 \ 0 \ g]^T$ in a horizontal frame $O-X_H Y_H Z_H$ whose Z_H axis is perfectly aligned with local gravity and origin is coincident with the origin of frame $O-X_T Y_T Z_T$. Furthermore, these two frames are roughly aligned which makes the rotation matrix \mathbf{R}_T^H between them to be approximated as

$$\mathbf{R}_T^H = \begin{bmatrix} 1 & -\gamma_T & \beta_T \\ \gamma_T & 1 & -\alpha_T \\ -\beta_T & \alpha_T & 1 \end{bmatrix}, \quad (5-15)$$

in which Euler angles α_T , β_T , γ_T describe the orientation of square tray frame with respect to horizontal frame. Thus, the vector of local gravity in tray frame is estimated as

$$\mathbf{G}_T = \mathbf{R}_T^H \mathbf{G} = \begin{bmatrix} 1 & -\gamma_T & \beta_T \\ \gamma_T & 1 & -\alpha_T \\ -\beta_T & \alpha_T & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} \beta_T g \\ -\alpha_T g \\ g \end{bmatrix} \quad (5-16)$$

With this approximation, another multi-position based process has been applied. Different from previous calibration process, the multi-position is performed by rotating the cube and then embedding it into the tray. Therefore the rotation matrix of body frame with respect to tray frame is known in every position. Then the misalignment is estimated by minimizing the error between the local gravity vector in tray frame, and, the accelerometer tri-axial readings that have been transformed to tray frame by calibrating parameters, and, corresponding rotation matrix.

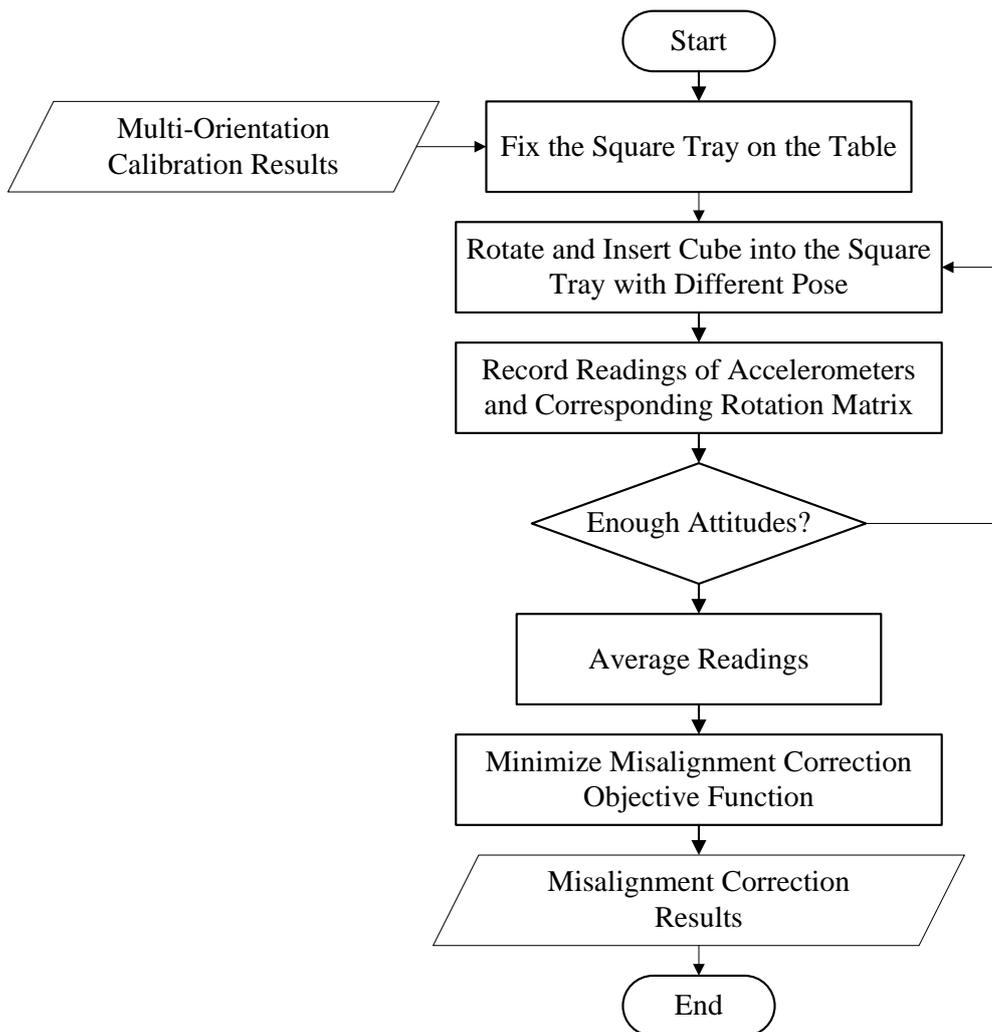


Figure 5-13 Flowchart of accelerometer Misalignment Correction

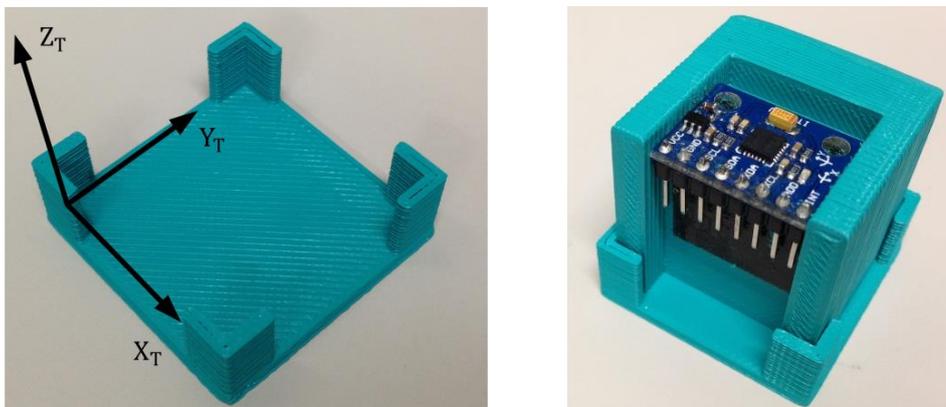


Figure 5-14 Square tray, tray frame and cube

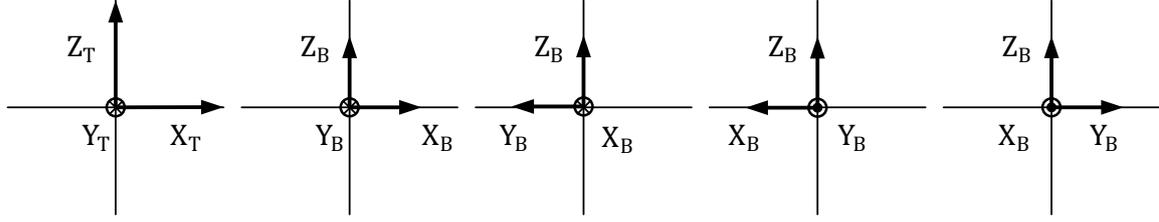


Figure 5-15 Rotate cube 4 time along axis Z_B

The detailed practical operation is summarized as follows:

- Fix the tray on a roughly horizontal plane of a stationary table, embed the cube into the tray and make body frame $O-X_B Y_B Z_B$ align with tray frame $O-X_T Y_T Z_T$.
- Rotate the cube and embed it into the tray, record the readings of accelerometer and also the rotation matrix of body frame $O-X_B Y_B Z_B$ with respect to tray frame $O-X_T Y_T Z_T$.
- Repeat last step until readings and rotation matrix of 24 orientations (6 faces of cube can be align with plane $O-X_T Y_T$ and 4 orientations of cube can arrived for each face in this condition, as shown in Figure 5-15) have been recorded.
- Average the tri-axial readings of accelerometer in each orientation.
- Estimate misalignment by solving minimization problem

$$(\hat{\alpha}_T, \hat{\beta}_T, \hat{\mathbf{M}}^{acc}) = \operatorname{argmin} \sum (\mathbf{G}_T - \mathbf{R}_T^B(i) \mathbf{M}^{acc} \bar{\mathbf{v}}_S^{acc}(i))^2 \quad (5-17)$$

where \mathbf{G}_T is vector of local gravity in tray frame, $\mathbf{R}_T^B(i)$ is the rotation matrix between body frame and tray frame of i -th orientation. And $\bar{\mathbf{v}}_S^{acc}(i)$ is the accelerometers' average measurements w.r.t sensor frame, which is obtained in i -th orientation and calibrated, as revealed in Equation (5-12).

The initial guess of minimization parameters is set as follows

$$\begin{aligned} \alpha_T &= 0, \beta_T = 0; \\ \alpha^{acc} &= 0, \beta^{acc} = 0, \gamma^{acc} = 0. \end{aligned} \quad (5-18)$$

Theoretically, this minimization approach not only estimates the misalignment between sensor fame and body frame but also finds the roll and pitch angles of the tray frame. However, the estimation of roll and pitch angles of tray frame is out of this study and it is difficult to obtain ground truth for their examination.

5.3.3 Gyroscope Calibration

This study aims to calibrate gyroscope without precise turntable, in which case accurate reference of angular velocity is difficult to obtain. Therefore, the angle domain calibration method has been applied which does not depend on knowledge of measured angular velocity but on the rotational angle obtained by numerical integration of gyroscopes' readings.

The calibration procedure of gyroscope is summarized as follows:

- a) The first step is similar to the one expressed in the calibration procedure of accelerometer. Connecting IMU to Arduino board with power access to warm up at least 10 minutes to suppress thermal effect.
- b) Then, the IMU is placed stationary over 50 seconds. The readings of the triads are recorded and finally averaged as the biases of gyroscope.
- c) Third, the cube is rotated along one axis of body frame $O-X_B Y_B Z_B$, in another word, rotate along one of its edges. The readings of triads are recorded and the reference of rotational angle is estimated. This step is detailed in the next part of this sub-section.
- d) Forth, last step is repeated until rotations along all three axes of body frame have been conducted and all the related data is recorded.
- e) Finally, the angle domain calibration method, which is explained in the last part of this sub-section, is applied to estimate the residual calibration parameters of gyroscope.



(a) Manually rotate



(b) Rotate by robot



(c) Rotate with swivel chair

Figure 5-16 Three methods to estimate angular displacement

Accurate rotational angle estimation

In order to obtain satisfactory result by utilizing angle domain calibration method, accurate reference of rotational angle of each rotation along axis of body frame is required. Furthermore, smooth and stable rotation is preferable in consideration of the accuracy of rotational angle estimated by gyroscope, with discrete time angular velocity integration.

In this study, three methods have been tried and corresponding readings from gyroscope have been recorded and analyzed to find out the best solution.

- 1) The first solution is to rotate the cube manually. Take advantage of the sharp of cube, rotate the cube against the wall and ground can obtain accurate rotational angle equal to times of 90° , as shown in Figure 5-16 (a). Theoretically, this should be the most promising method for its easy operation and high accuracy. However, in the real application, this method sometimes turns out to be difficult to conduct. There are two reasons: the first one is the interruption of the connecting wire makes the cube unstable; the second reason is that manually rotation can easily introduce some vibrations which may affect the accuracy of the final integrated angular displacement, as shown in Figure 5-17 (a).
- 2) After first trail, the cube is fixed on the top plate where there is LRF, as shown in Figure 5-16 (b). Then the rotation is introduced by robot and the rotational angle is estimated by using incremental laser scan matching algorithm proposed in Chapter 3. This solution is rejected soon because of huge vibration observed during the robot motion, as shown in Figure 5-17 (b).
- 3) The finally adopted solution is to rotate the cube by using an easy turntable. The turntable is made by removing seat pad of a normal swivel chair. The top plate of robot is taken and then fixed on the top of chair's spin axis, as showing in Figure 5-16 (c). The plate is adjusted to be orthogonal to the spin axis by rotating chair to different direction and examining readings of the calibrated accelerometers. Rotate with this easy turntable, the readings of gyroscopes are very smooth which is desired to obtain accurate estimation of rotational angle by numerical integrations, as shown in Figure 5-17 (c). The rotational angle is obtained by using laser scan matching algorithm. And the scan matching result is plotted in Figure 5-17 (d).

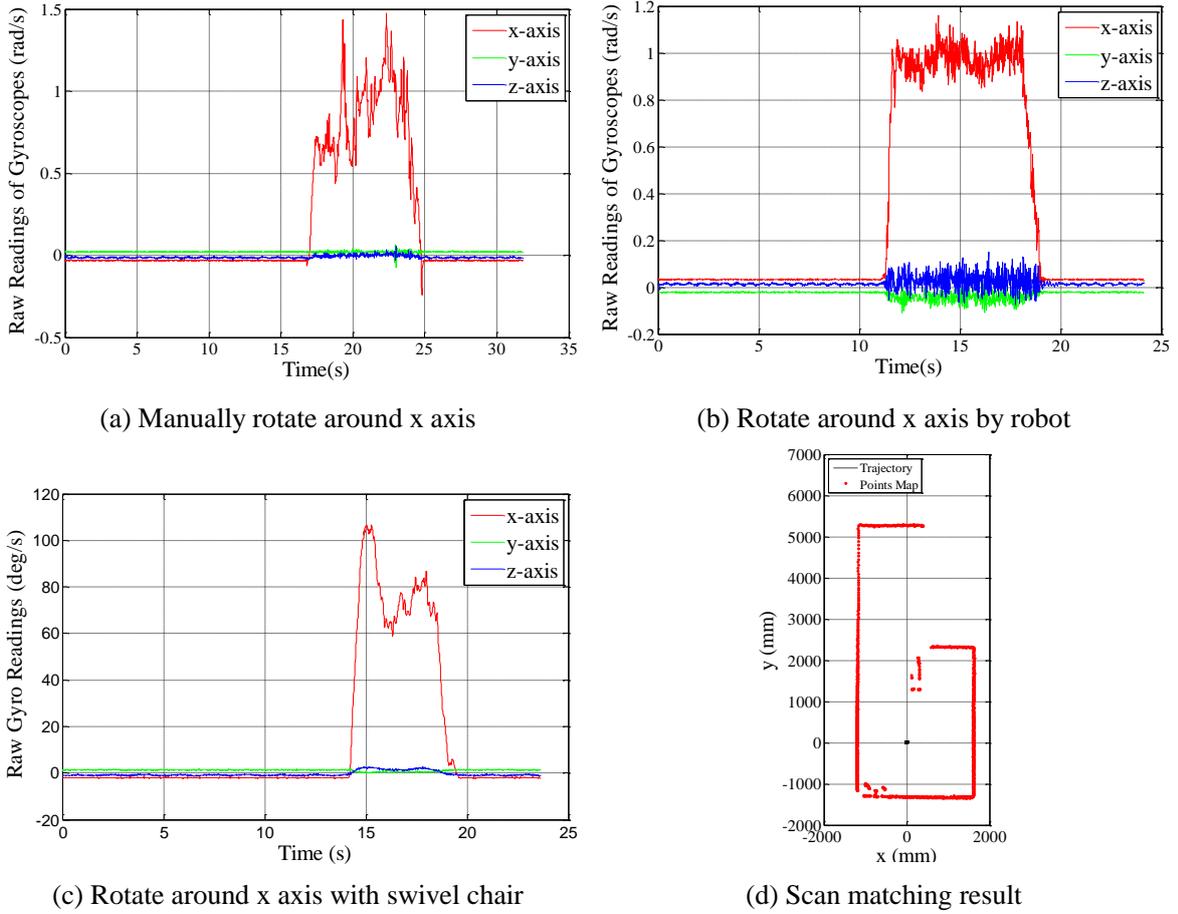


Figure 5-17 Gyroscope raw readings with three rotation methods

Angle Domain Calibration

The vector of rotational angle estimated by gyroscope when IMU is rotated along the X_B axis of body frame can be calculated by numerical integration as

$$\mathbf{U}_x^{gyr} = \begin{bmatrix} \theta_{xx}^{gyr} \\ \theta_{yx}^{gyr} \\ \theta_{zx}^{gyr} \end{bmatrix} = \sum_{i=0}^n (\mathbf{u}^{gyr}(i) - \mathbf{b}^{gyr})/f \quad (5-19)$$

where $\mathbf{u}^{gyr}(i)$ is the i^{th} readings vector of the tri-axial gyroscope, \mathbf{b}^{gyr} is the vector of biases, f is the frequency of gyroscope, n is the number of gyroscope readings in the period of rotation. The first letter of subscript of the angle in the vector indicates which gyroscope axis it reads from and the second letter indicates which axis of body frame the IMU is rotating along. For instance, θ_{yx}^{gyr} is the integrated angle of gyroscope Y axis readings when the IMU is rotated along X_B .

With the same method, the vectors of rotational angle estimated by gyroscope when rotation is along Y_B and Z_B can be calculated. Three vectors form the rotational angle matrix estimated by gyroscope

$$\mathbf{U}^{gyr} = [\mathbf{U}_x^{gyr} \quad \mathbf{U}_y^{gyr} \quad \mathbf{U}_z^{gyr}] = \begin{bmatrix} \theta_{xx}^{gyr} & \theta_{xy}^{gyr} & \theta_{xz}^{gyr} \\ \theta_{yx}^{gyr} & \theta_{yy}^{gyr} & \theta_{yz}^{gyr} \\ \theta_{zx}^{gyr} & \theta_{zy}^{gyr} & \theta_{zz}^{gyr} \end{bmatrix} \quad (5-20)$$

Meanwhile, laser scan matching method provides accurate reference of rotational angle matrix. Since the rotations are conducted by align each axis of body frame along with the spin axis of turntable. The reference matrix is diagonal as follows

$$\mathbf{V}^{ref} = \begin{bmatrix} \theta_{xx}^{ref} & 0 & 0 \\ 0 & \theta_{yy}^{ref} & 0 \\ 0 & 0 & \theta_{zz}^{ref} \end{bmatrix}. \quad (5-21)$$

Therefore, Equation (5-1) can be rewritten to integrated form:

$$\mathbf{V}^{ref} = \mathbf{M}^{gyr} \mathbf{N}^{gyr} \mathbf{S}^{gyr} \mathbf{U}^{gyr}. \quad (5-22)$$

\mathbf{U}^{gyr} is constructed of main rotational components in diagonal and small off-diagonal noises brought by misalignment and non-orthogonality. Thus it must be an invertible matrix. Then the Equation (5-22) can be modified to

$$\mathbf{V}^{ref} (\mathbf{U}^{gyr})^{-1} = \mathbf{M}^{gyr} \mathbf{N}^{gyr} \mathbf{S}^{gyr}. \quad (5-23)$$

Symmetrical matrix can be obtained in both side by multiplying corresponding transport matrix as

$$(\mathbf{V}^{ref} (\mathbf{U}^{gyr})^{-1})^T (\mathbf{V}^{ref} (\mathbf{U}^{gyr})^{-1}) = \mathbf{S}^{gyr T} \mathbf{N}^{gyr T} \mathbf{M}^{gyr T} \mathbf{M}^{gyr} \mathbf{N}^{gyr} \mathbf{S}^{gyr} \quad (5-24)$$

Since \mathbf{M}^{gyr} stands for the rotation matrix between two orthogonal frame, which makes

$$\mathbf{M}^{gyr T} \mathbf{M}^{gyr} = \mathbf{I}. \quad (5-25)$$

where \mathbf{I} is the identity matrix. Then Equation (5-24) is simplified to:

$$(\mathbf{V}^{ref}(\mathbf{U}^{gyr})^{-1})^T(\mathbf{V}^{ref}(\mathbf{U}^{gyr})^{-1}) = \mathbf{S}^{gyrT}\mathbf{N}^{gyrT}\mathbf{N}^{gyr}\mathbf{S}^{gyr} \quad (5-26)$$

The matrix $\mathbf{N}^{gyr}\mathbf{S}^{gyr}$ can be estimated by using Cholesky decomposition which

$$\mathbf{N}^{gyr}\mathbf{S}^{gyr} = \text{chol}[(\mathbf{V}^{ref}(\mathbf{U}^{gyr})^{-1})^T(\mathbf{V}^{ref}(\mathbf{U}^{gyr})^{-1})]^T \quad (5-27)$$

Since scaling factor matrix \mathbf{S}^{gyr} is diagonal matrix and non-orthogonal matrix \mathbf{N}^{gyr} is a lower triangular matrix, they can be decomposed from the multiplication matrix by using LU factorization as:

$$[\mathbf{N}^{gyr}, \mathbf{S}^{gyr}] = \text{LU}[\mathbf{N}^{gyr}\mathbf{S}^{gyr}]. \quad (5-28)$$

Finally, the misalignment can be calculated as:

$$\mathbf{M}^{gyr} = \mathbf{V}^{ref}(\mathbf{U}^{gyr})^{-1}(\mathbf{S}^{gyr})^{-1}(\mathbf{N}^{gyr})^{-1} \quad (5-29)$$

The detailed algorithm is described in [99] and [102].

5.4 Experimental Results

The first part of this section gives the calibration results of accelerometer and gyroscope. The second part shows the performance of IMU based Gait tracking by using MEMS-IMU with/without calibration.

5.4.1 Calibration Results

Accelerometer

Table 5-2 lists out all of the calibrated parameters of accelerometer.

Table 5-2 Calibrated Parameters of Accelerometer

<i>Biases (m/s²)</i>	<i>Scaling Factors</i>	<i>Non-Orthogonality (deg)</i>	<i>Misalignment (deg)</i>
$b_x^{acc} = 0.2666$	$s_x^{acc} = 1.0030$	$\varphi_{yx}^{acc} = -0.0172$	$\alpha^{acc} = 0.5787$
$b_y^{acc} = -0.0684$	$s_y^{acc} = 1.0035$	$\varphi_{zx}^{acc} = 0.2521$	$\beta^{acc} = 0.8422$
$b_z^{acc} = -0.2048$	$s_z^{acc} = 0.9734$	$\varphi_{zy}^{acc} = -0.2922$	$\gamma^{acc} = 0.0630$

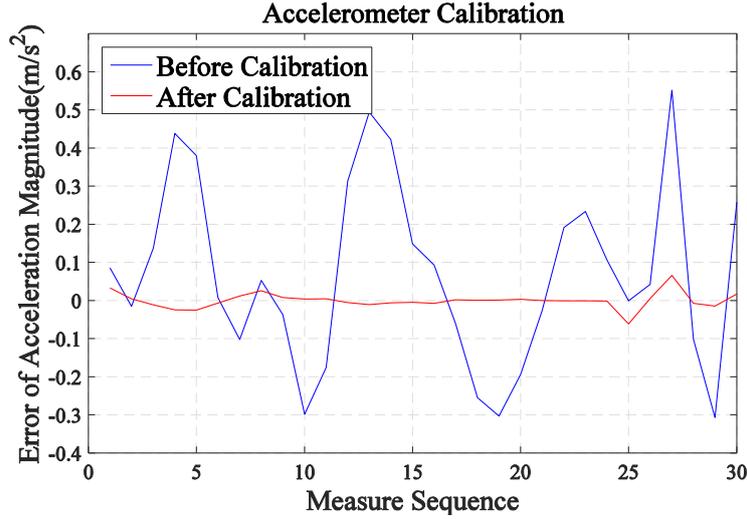


Figure 5-18 Magnitude error collapse after calibration

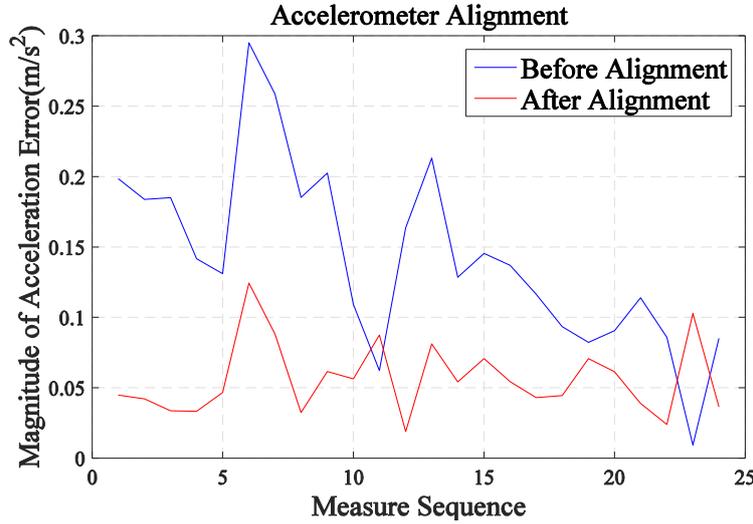


Figure 5-19 Error magnitude is diminished after alignment

The Figure 5-18 shows that the error between measured accelerations' magnitudes and its reference local gravity is efficiently suppressed by the calibration work. The initial error that is represented by blue line is calculated by comparing the magnitude of gravity and magnitude of raw triads' static readings as follows

$$\sigma_{before}^{calibration}(i) = |\mathbf{G}| - |\bar{\mathbf{u}}^{acc}(i)|, \quad i = 1, 2, \dots, 30. \quad (5-30)$$

The error after calibration, which is plotted by red line, is calculated by comparing the magnitude of gravity and magnitude of calibrated readings of accelerometer as follows

$$\sigma_{after}^{calibration}(i) = |\mathbf{G}| - |\mathbf{N}^{acc} \mathbf{S}^{acc} (\bar{\mathbf{u}}^{acc}(i) - \mathbf{b}^{acc})|, \quad i = 1, 2, \dots, 30. \quad (5-31)$$

The Root Mean Square Error (RMSE) of acceleration magnitude before calibration is 0.246864m/s². And after calibration, the RMSE reduces to 0.020344 m/s².

After the calibration have diminished the biases, scaling factors and non-orthogonality of accelerometer, the alignment between sensor frame and body fame is conducted. Figure 5-19 shows the magnitude of error vector of accelerations with/without alignment. By applying the proposed misalignment correction method, the vector of local gravity in tray frame \mathbf{G}_T is estimated along with the misalignment matrix \mathbf{M}^{acc} . Then, the average readings of accelerometer in each static orientation are transformed to tray frame with/without introducing misalignment matrix. And finally the transformed readings are compared with the local gravity in tray frame as follows

$$\sigma_{before}^{alignment}(i) = |\mathbf{G}_T - \mathbf{R}_T^B(i)\bar{\mathbf{v}}_S^{acc}(i)|, i = 1,2, \dots,30, \quad (5-32)$$

$$\sigma_{after}^{alignment}(i) = |\mathbf{G}_T - \mathbf{R}_T^B(i)\mathbf{M}^{acc}\bar{\mathbf{v}}_S^{acc}(i)|, i = 1,2, \dots,30. \quad (5-33)$$

The error vectors are calculated and their magnitudes are plotted in Figure 5-19.

The RMSE of the error before alignment is 0.155883m/s² while it is diminished to 0.061735m/s² after alignment, which shows the necessity of alignment.

Gyroscope

All of the 12 calibrated parameters of gyroscope are listed out in Table 5-3. The intuitive improvement brought by the calibration can be found in Table 5-4, Figure 5-20, Figure 5-21 and Figure 5-22. Table 5-4 lists the rotational angle integrated by gyroscope raw readings and reference rotational angle provided by laser scan matching. The error is obvious and it is efficiently eliminated by the calibration process.

Table 5-3 Calibrated Parameters of Gyroscope

Biases (deg/s)	Scaling Factors	Non-Orthogonality (deg)	Misalignment (deg)
$b_x^{gyr} = -2.0297$	$s_x^{gyr} = 1.0028$	$\varphi_{yx}^{gyr} = -0.9511$	$\alpha^{gyr} = 0.3610$
$b_y^{gyr} = 1.3546$	$s_y^{gyr} = 1.0021$	$\varphi_{zx}^{gyr} = -1.2834$	$\beta^{gyr} = 0.6245$
$b_z^{gyr} = -0.8975$	$s_z^{gyr} = 0.9992$	$\varphi_{zy}^{gyr} = 1.0141$	$\gamma^{gyr} = -0.3151$

Table 5-4 Comparison of Estimated Rotation Angles

<i>Gyroscope Integrated Rotational angle</i> <i>Matrix U^{gyr} (deg)</i>	<i>Scan Matching Reference Rotational</i> <i>angle Matrix V^{ref} (deg)</i>	<i>Angular Error After</i> <i>Calibration (deg)</i>
$\begin{bmatrix} 353.4059 & -2.1773 & -3.7280 \\ -3.9243 & 371.1555 & 2.3333 \\ 11.9167 & -9.0066 & 341.2379 \end{bmatrix}$	$\begin{bmatrix} 354.2717 & 0 & 0 \\ 0 & 371.9059 & 0 \\ 0 & 0 & 341.2675 \end{bmatrix}$	Error $< 10^{-5}$

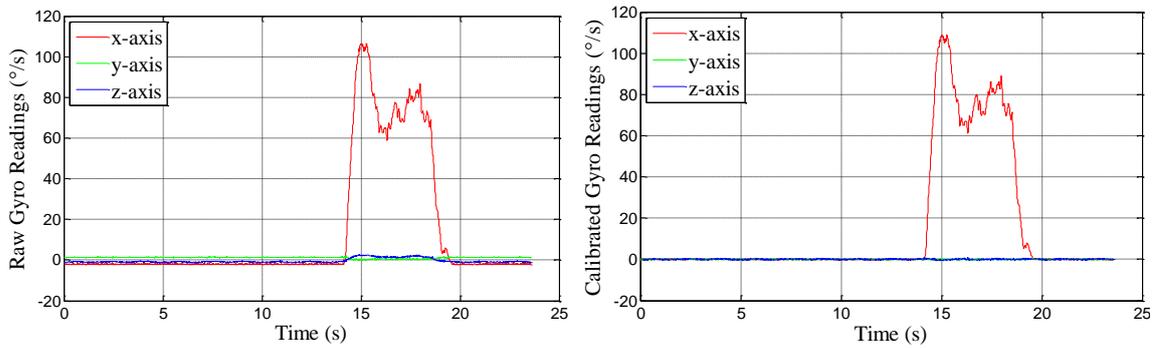


Figure 5-20 Raw and calibrated readings of gyroscope when IMU rotates along X_B axis

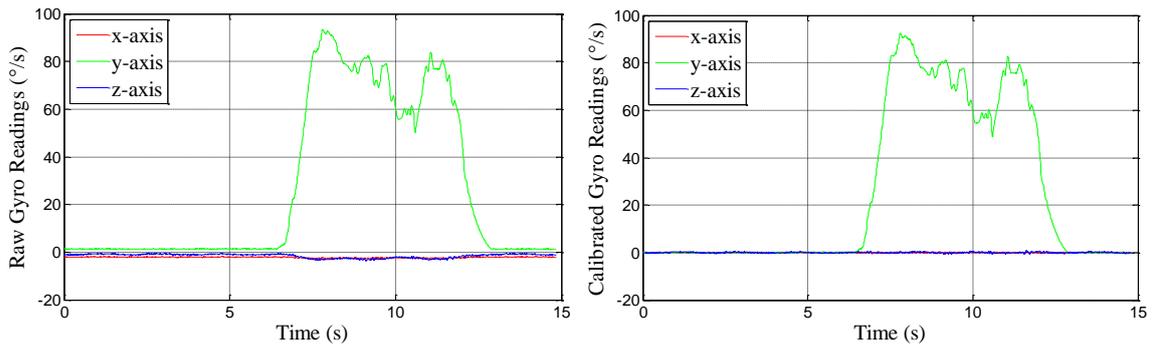


Figure 5-21 Raw and calibrated readings of gyroscope when IMU rotates along Y_B axis

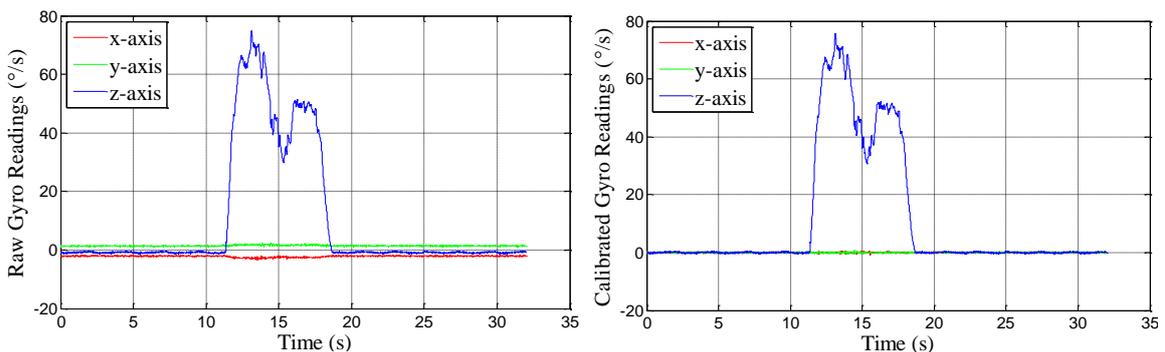


Figure 5-22 Raw and calibrated readings of gyroscope when IMU rotates along Z_B axis

Above figures show the raw and calibrated readings of gyroscope. The Error of biases, non-orthogonality and misalignment are obviously suppressed after calibration.

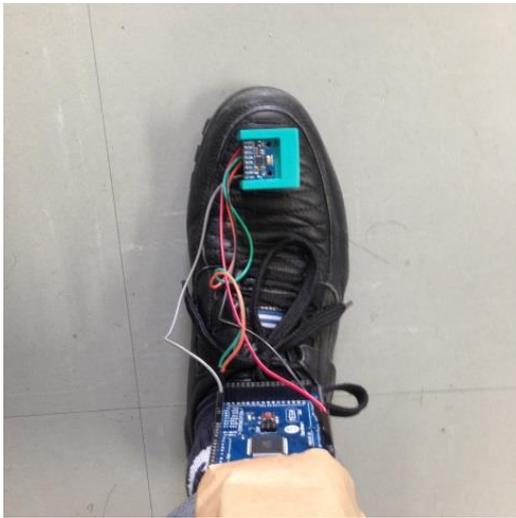
5.4.2 Application in IMU based Gait tracking

A further verification on the calibration work is done by applying IMU based gait tracking [103]. The measurements of gyroscopes are often used to be integrated to estimate the change of orientation. During the motion with small external acceleration, the measurements of accelerometers can be regarded as the acceleration of constant local gravity that is corrupted with noise. Therefore, the drifts of orientation components roll and pitch can be bounded by fusing measurements of gyroscopes and accelerometers [42][104]. In theory, the trajectory of IMU during the motion can be estimated by integrating the measurements of accelerometers twice with the estimated orientation. However, the trajectory estimation based on double integration without error bound can diverge in a short time period because of unavoidable stochastic noises and systematic errors. When a person walks naturally, the foot is periodically hitting the ground and keeps stationary for a short moment. By correcting the drift of integrated velocities during this stationary moment, IMU based gait tracking can fulfill accurate trajectory tracking for a long time.

The cube that carrying calibrated IMU is attached on researcher's right shoe and researcher walks up the stairs to the upper floor, as shown in Figure 5-23. Researcher starts to walk at the position that is one step from the stairs and walks along the middle line of the stairs to the upper floor. After arriving at the upper floor, researcher stops walking at the position which is similar with the start position. The gait tracking application adopted in this study is developed from the open source code published by S. Madgwick [105]. The modification is introduced to make the application suits to the configuration of the applied IMU.

Figure 5-24 shows the 3D trajectories estimated by integrating raw and calibrated IMU measurements. The trajectory estimated by using raw measurements is obviously twisted while the calibrated result matches the shape of stairs much better. Figure 5-25, Figure 5-26, Figure 5-27 give top view, front view, and lateral view of the trajectory, respectively. Among this view, the most severe distortion of the trajectories can be found in the top view. This is because the error and drift of orientation in yaw cannot be bounded by using local gravity. On the other hand, the trajectory estimated by applying calibrated measurements shows much better result especially in the top view, which

means the systematic errors have been well compressed. Therefore the validity of the previous calibration work has been further verified.

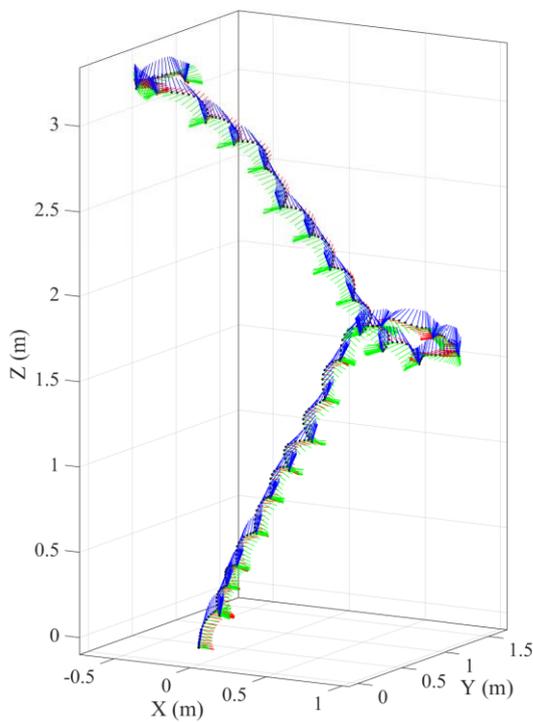


(a) IMU attached on foot

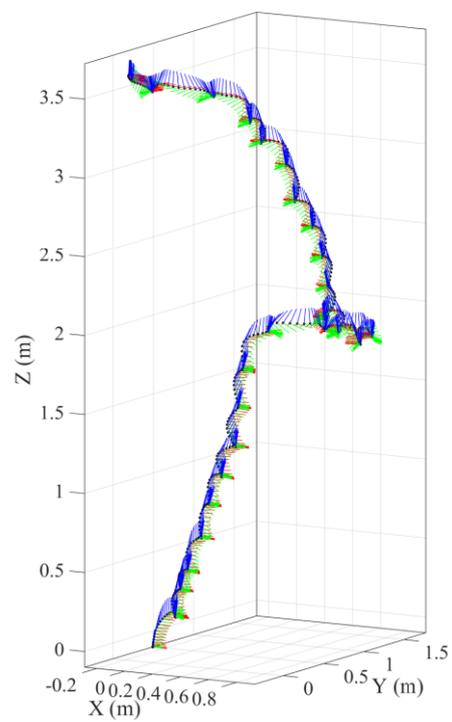


(b) Upstairs

Figure 5-23 IMU placement and gait tracking environment



(a) Using uncalibrated measurements



(b) Using calibrated measurements

Figure 5-24 Gait tracking trajectory results in 3D space

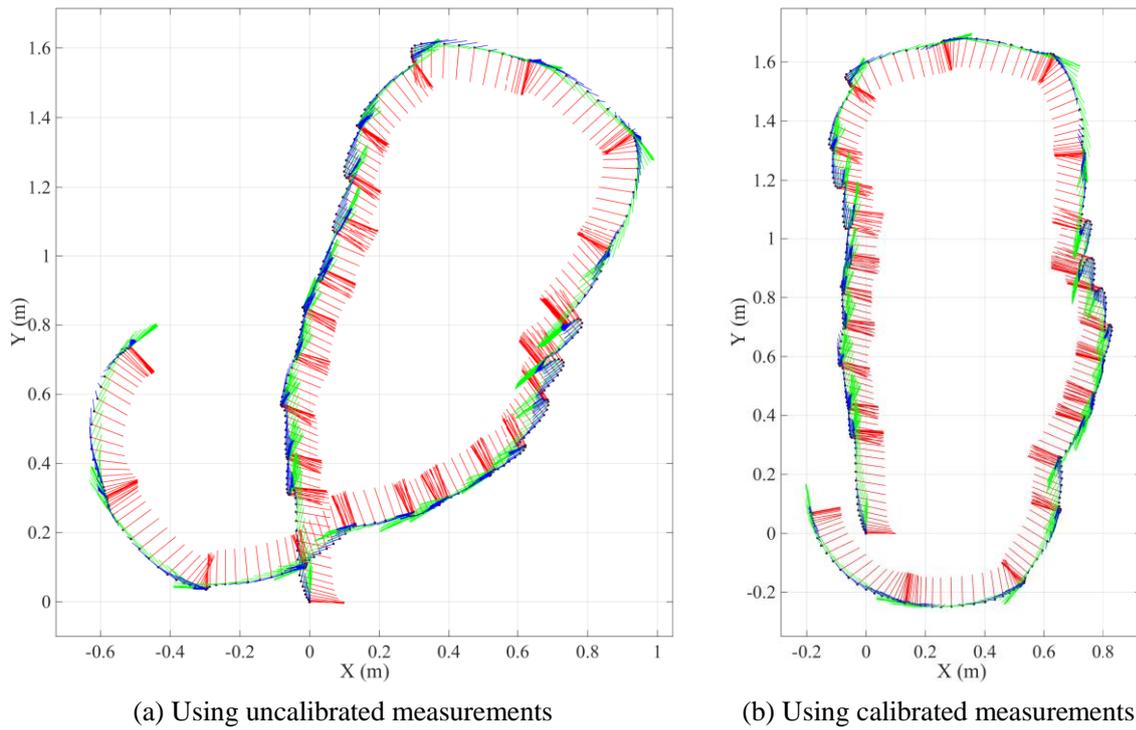


Figure 5-25 Gait tracking trajectory results in XY plane, top view

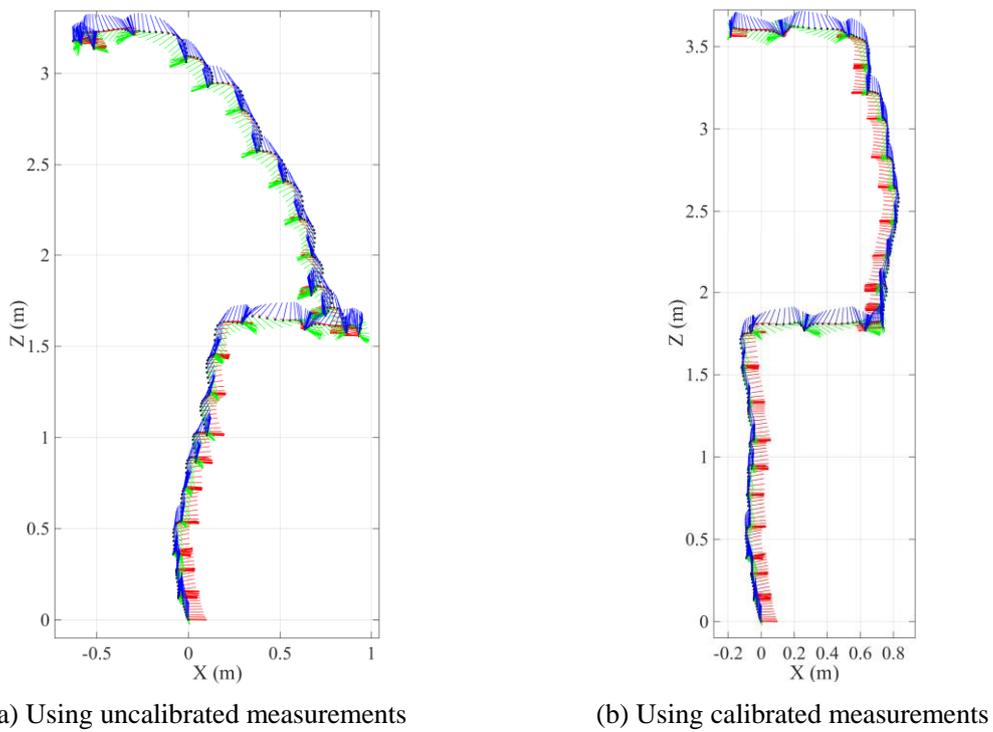


Figure 5-26 Gait tracking trajectory results in XZ plane, front view

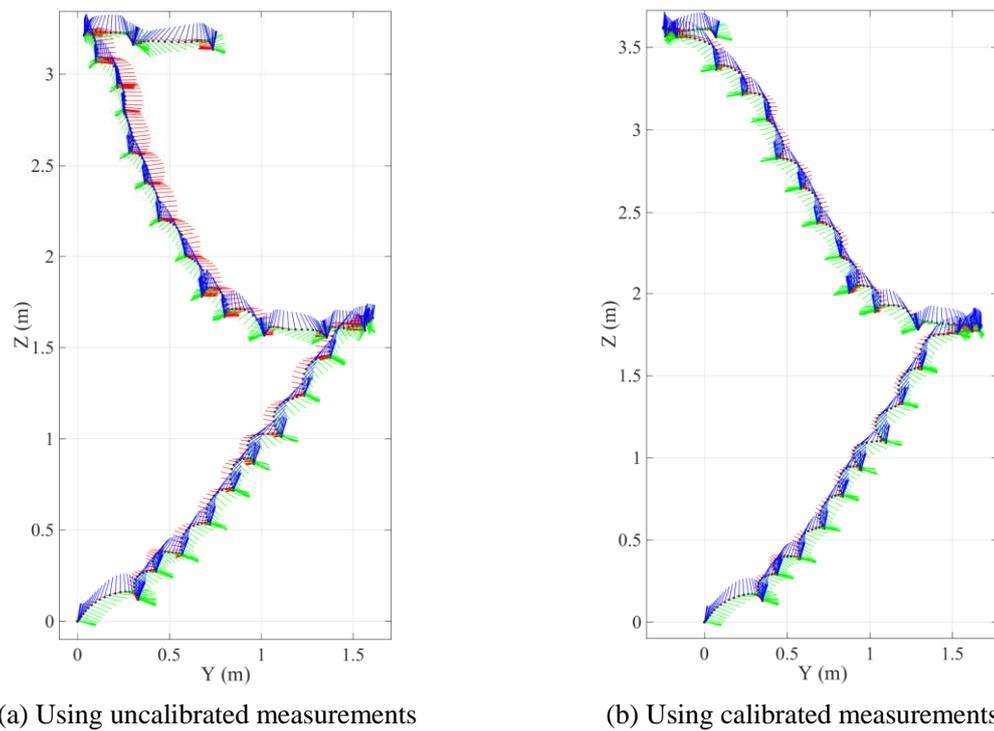


Figure 5-27 Gait tracking trajectory results in YZ plane, lateral view

5.5 Conclusion

In this chapter, a consumer grade MEMS-IMU that captures triads' gyroscope and triads' accelerometer has been investigated. Various types of sensor errors that affect the accuracy of measurements of low-cost MEMS-IMU have been firstly introduced. And a uniform sensor error model of accelerometers and gyroscopes has been adopted. Then, the static performance of MEMS-IMU is examined and a straightforward calibration process has been proposed to compress the error of the MEMS-IMU. Local gravity is utilized as the reference of accelerometers' measurements while scan matching method offers angular reference for gyroscopes' calibration. The uncalibrated/calibrated measurements of MEMS-IMU are compared with the references to verify the efficiency of the proposed calibration method. Furthermore, an IMU based gait tracking application is conducted by using the uncalibrated/calibrated measurements. The result clearly shows the improvement brought by the calibration work.

Chapter 6 Conclusion and Future work

6.1 Conclusion

This thesis is focusing on the study of indoor mobile robots navigation. SLAM is one of the primary techniques to enable mobile robot to realize autonomous navigation. A key methodology in SLAM, scan matching has been intensively investigated in this study. The contributions of this study can be summarized as follows

- 1. Orientation Estimator for Scan Matching:** A new orientation estimator ICN has been proposed and compared with another straightforward orientation estimator POE. ICN turns out to be more accurate and more robust against the effect of translation.
- 2. Pre-processor for Scan Matching:** Based on the analysis on the incident angle, a pre-processor for scan matching named RIA has been proposed. RIA can efficiently eliminate dominant component of the transformation between scans.
- 3. New association metric for Scan Matching:** Another application of incident angle is a new association metric named IAFM. Based on this new metric, closest point rule based association process becomes more efficient and robust on excluding outliers.
- 4. New framework for incremental scan matching:** To overcome the limitation of SPM-ICP, a sub-maps joining version of it, Split SPM-ICP has been presented in this study. This new framework can handle loop closure problem as well as reducing computational cost of scan matching.
- 5. Optional prediction model for EKF-SLAM:** Utilizing robust and efficient scan matching algorithm, a hybrid prediction model for EKF-SLAM has been proposed. This new prediction method can eliminate the effect brought by outlier readings from wheel odometer. And it is able to automatically switch prediction method

between odometer and scan matching, which saves the computational cost and makes the prediction result reliable.

- 6. Slope and edge detection by using line-segment based EKF-SLAM:** Based on the sound Orthogonal Assumption, the structured indoor slope and edge have been modelled into line-segment and merged into EKF-SLAM framework. This method is straightforward and does not need to extract features from 3D point clouds, which dramatically simplifies the calculation work.
- 7. Straightforward calibration method for MEMS-IMU:** By using scan matching method, accurate angular displacement estimation has been provided for gyroscope calibration. The accelerometer has been calibrated by utilizing local gravity as the reference vector. A plastic cube that is made by 3D printer has been introduced to eliminate the residual misalignment error.

6.2 Future work

There are quite a few future works can be done based on current achievements described in this thesis. They are summarized and list as follows:

- 1. Probability based Optimization:** ICP based scan matching is very accurate and robust especially in the manner of incremental scan matching. In this study, the loop closing method in incremental scan matching approach is not probability based solution. This method works satisfactory when the environmental constraints are sufficient and the accumulate error of scan matching is still in tolerance. On the other hand, for loops that contain under-constrain area for scans, probability based approach is necessary to conduct optimization solution so that the error of scan matching result in such area could be compressed.
- 2. EKF-SLAM Speed up:** The main limitation of EKF-SLAM based application is the huge computational cost which is increasing with the addition of features' number. Further assumption or compression method can be introduced to speed up the line-segments based EKF-SLAM that has been investigated in this study. For example, instead of taking every line-segment as an independent feature, the line-segments that have similar parameters and separate location can be maintained as one feature in EKF state space, which is a good simplification in structure indoors and can dramatically reduce the features' number.

- 3. Sensor Fusion:** In this study, LRF is the only exploring sensor that has been utilized. Although LRF has various merits, a single type of exploring sensor can never guarantee the robustness of sufficient exploring in all circumstances. Therefore, diverse sensors' data fusion is a necessary methodology to conduct safe autonomous navigation. Comparing with planar LRF, vision sensor can provide dense spatial information of the surrounding and usually needs lower cost in price, which makes vision sensor based SLAM solution is very popular in recent years. A fusion of LRF, vision sensor as well as calibrated IMU should be able to be a robust approach to solve SLAM problem.
- 4. Path planning:** The robot is controlled to move in indoors to conduct scan matching and SLAM work in this study. One of the immediate following works should be the combination of current achievement with some fast and robust path planning algorithm which is another vital technology in robot autonomous navigation task. There are famous path planning algorithms have been developed, such as typical Dijkstra algorithm, A-Star algorithm and Dynamic A-Star (D-Star) algorithm. The comparison and modification towards these algorithms might be necessary before applying.

Reference

- [1] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: Part I. *Robotics Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99-108, Jun. 2006.
- [2] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *Robotics Automation Magazine, IEEE*, vol. 13, no. 2, pp.108–117, Sep. 2006.
- [3] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agent)*. The MIT Press, 2005.
- [4] U. Frese. A discussion of simultaneous localization and mapping. *Autonomous Robots*, vol. 20, pp. 25-42, 2006.
- [5] S. Thrun. Learning occupancy grids with forward sensor models. *Autonomous Robots*, 2002.
- [6] S. Y. An, J. G. Kang, L. K. Lee, and S. Y. Oh. Line-segment-Based Indoor Mapping with Salient Line Feature Extraction. *Advanced Robotics*, vol. 26, pp. 437-460, 2012.
- [7] http://en.wikipedia.org/wiki/Kalman_filter
- [8] J. A. Castellanos. *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. PhD dissertation, Universidad de Zaragoza, Spain, 1998.
- [9] G. Dissanayake, P. Newman, H.F. Durrant-Whyte, S. Clark, and M. Csobra. A solution to the simultaneous localisation and mapping (SLAM) problem. *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp.229–241, 2001.
- [10] S. Huang and G. Dissanayake. Convergence analysis for extended Kalman filter based SLAM. In *IEEE International Conference on Robotics and Automation*. 2006.
- [11] A Doucet, N. de Freitas, K. Murphy, and S. Russell. RaoBlackwellised particle filtering for dynamic Bayesian networks. *Uncertainty in Artificial Intelligence*, pp. 176-183, 2000.

-
- [12] K. Murphy and S. Russell. Rao-blackwellized particle filtering for dynamic bayesian networks. In *Sequential Monte Carlo Methods in Practice*, Springer, 2001.
- [13] M. Montemerlo. Fastslam: A factored solution to the simultaneous localization and mapping problem with unknown data association. Ph.D. dissertation, 2003.
- [14] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.
- [15] A. Eliazar, and R. Parr. Dp-slam: Fast, robust simultaneous localization and mapping without predetermined landmarks. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 1135–1142, 2003.
- [16] S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research (IJRR)*, vol. 25, no. 5-6, pp. 403, 2006.
- [17] G. Grisetti, R. Kümmerle, C. Stachniss, W. Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31-43, 2010.
- [18] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
- [19] N. Sunderhauf and P. Protzel. Switchable Constraints for Robust Pose Graph SLAM. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [20] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2006.
- [21] T. Furukawa, L. Dantanarayana, J. Ziegler, R. Ranasinghe and G. Dissanayake. Fast Global Scan Matching for High-Speed Vehicle Navigation. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sep. 14-16, 2015.

-
- [22] Ryu, K.J. Autonomous Robotic Strategies for Urban Search and Rescue. Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, 2012.
- [23] Y. S. Chou, J. S. Liu. A Robotic Indoor 3D Mapping System Using a 2D Laser Range Finder Mounted on a Rotating Four-Bar Linkage of a Mobile Platform. *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, 2013.
- [24] T. Elseberg, D. Borrmann and A. Nüchter. 6DOF semi-rigid slam for mobile scanning. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [25] S. Thrun, W. Burgard and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2000.
- [26] L. Kurnianggoro, V.-D. Hoang, and K.-H. Jo. Calibration of a 2D Laser Scanner System and Rotating Platform using a Point-Plane Constraint. *Computer Science and Information Systems*, vol. 12, no. 1, pp. 307–322, 2015.
- [27] Q. Zhang, R. Pless. Extrinsic Calibration of a Camera and Laser Range Finder (improves camera calibration). *Proceedings Of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep.28 - Oct. 2, 2004.
- [28] A. Davison, I. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, no. 6, pp.1052-1067, 2007.
- [29] R. Mur-Artal, J. M. M. Montiel and Juan D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, Oct.2015.
- [30] J. Engel, T. Schöps, D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *European Conference on Computer Vision (ECCV)*, 2014.
- [31] T. Lemaire, C. Berger, I. Jung and S. Lacroix. Vision-Based SLAM: Stereo and Monocular Approaches. *International Journal of Computer Vision*, vol. 74, no. 3, pp. 343-364, 2007.
- [32] J. Engel, J. Stueckler, D. Cremers. Large-Scale Direct SLAM with Stereo Cameras. In *International Conference on Intelligent Robots and Systems (IROS)*, 2015.

-
- [33] D. Caruso, J. Engel, D. Cremers. Large-Scale Direct SLAM for Omnidirectional Cameras. International Conference on Intelligent Robots and Systems (IROS), 2015.
- [34] P. Sturm and S. Maybank. On plane-based camera calibration: a general algorithm, singularities, applications. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 432–437, Jun. 1999, USA.
- [35] Z. Zhang. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence. vol.22, no.11, pp. 1330–1334, 2000.
- [36] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, W. Burgard. An Evaluation of the RGB-D SLAM System. IEEE International Conference on Robotics and Automation (ICRA), 2012.
- [37] M. Bautista, A. Hernandez-Vela, V. Ponce, X. Perez-Sala, X. Bar, O. Pujol, C. Angulo, S. Escalera. Probability-based Dynamic Time Warping for Gesture Recognition on RGB-D data. International Workshop on Depth Image Analysis. 2012.
- [38] K. Khoshelham, S. Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. Sensors, vol. 12, no. 2, pp.1437-1454, 2012.
- [39] D. Herrera, J. Kannala, J. Heikkilä Joint depth and color camera calibration with distortion correction. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 34, no. 10, pp. 2058-2064, 2012.
- [40] B. Karan. Calibration of Kinect-type RGB-D sensors for robotic applications. FME Transactions, vol. 43, no. 1, pp. 47-54, 2015.
- [41] O. J. Woodman. An introduction to inertial navigation. Technical report, published by the University of Cambridge Computer Laboratory, 2007, <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.html>, available via Internet.
- [42] S. O. H. Madgwick, A. J. L. Harrison and R. Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. Proceedings of IEEE International Conference Rehabilitation Robotics, vol. 51, pp. 1-7, 2011.
- [43] G. Nutzi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of IMU and vision for absolute scale estimation in monocular SLAM. Journal of Intelligent and Robotic Systems, vol. 61, pp. 287–299, 2010.

-
- [44] A. Martinelli. Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44-60, 2012.
- [45] Y. Li, J. Georgy, X. Niu, Q. Li, N. El-Sheimy. Autonomous calibration of MEMS gyros in consumer portable devices. *IEEE Sensor Journal*, vol. 15, no. 7, pp. 4062–4072, 2015.
- [46] P. Biber, W. Strasser. The normal distributions transform: A new approach to laser scan matching. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2743–2748, 2003.
- [47] A. Diosi and L. Kleeman. Fast Laser Scan Matching using Polar Coordinates. *The International Journal of Robotics Research*, vol. 26, no. 10, pp. 1125, 2007.
- [48] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Maching Intelligence*, vol. 14, no. 2, pp. 239-256, 1992.
- [49] J. L. Martinez, J. Gonzalez, J. Morales, A. Window and J. Garcia-Cerezo. Genetic and ICP Laser Point Matching for 2D Mobile Robot Motion Estimation. *Journal of Field Robotics*, vol. 23, no. 1, pp. 21-34, 2006.
- [50] S.Druon, M. Aldon, A. Crosnier. Color constrained icp for registration of large unstructured 3d color data sets. *IEEE International Conference on Information Acquisition*, pp. 249–255 , Aug. 2006.
- [51] D. Chetverikov, D. Svirko, D. Stepanov, P. Krsek. The trimmed iterative closest point algorithm. *Proceedings of 16th International Conference on Pattern Recognition*, vol. 3, pp. 545–548, 2002.
- [52] F. Pomerleau, F. Colas, F. Ferland, F. Michaud. Relative Motion Threshold for Rejection in ICP Registration. *Proceedings of the 7th International Conference on Field and Service Robotics (FSR)*, Jul. 2009.
- [53] F. Lu, E. Milos. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. *Journal of Intelligent Robotics Systems*, vol. 18, pp. 249–275, 1997.
- [54] J. Minguez, F. Lamiroux, L. Montesano. Metric-Based Scan Matching Algorithms for Mobile Robot Displacement Estimation. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2005.
- [55] A. Censi. An ICP variant using a point-to-line metric. *Proceeding of the IEEE International Conference on Robotics & Automation (ICRA)*, 2008.

-
- [56] S. Steinmann. Analysis of Scan Matching Methods for Indoor 3D Mapping. Semester-Thesis, ETH, Spring Term, 2009.
- [57] J. Lv, K. Yukinori, A. A. Ravankar, A. Ravankar, T. Emaru. A Solution to Estimate Robot Motion with Large Rotation by Matching Laser Scans. 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), 2015.
- [58] G. Weiß, C. Wetzler, E. von Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. Proceeding of International Conference on Intelligent Robots and Systems. pp. 595–601, 1994.
- [59] D. Musser. Introspective Sorting and Selection Algorithms. Software Practice and Experience, vol.27, pp. 983-993, 1997.
- [60] N. Roussopoulos, S. Kelley, F. D. R. Vincent. Nearest neighbor queries. Proceedings of the ACM SIGMOD International Conference on Management of data, pp.71-79, 1995.
- [61] Y. Malkov, A. Ponomarenko, V. Krylov, A. Logvinov. Approximate nearest neighbor algorithm based on navigable small world graphs. Information Systems no. 45, pp. 61–68, 2014.
- [62] D. Yankov, E. J. Keogh, L. Wei, X. Xi, and W. L. Hodges. Fast Best-Match Shape Searching in Rotation-Invariant Metric Spaces. IEEE Transactions on Multimedia, vol. 10, no. 2, pp. 230-239, 2008.
- [63] J.S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), 1999.
- [64] J. BENTLEY. Multidimensional binary search trees used for associative searching. Communications of the ACM. vol. 18, no. 9, pp. 509–517, 1975.
- [65] R. A. Brown. Building k-d Tree in $O(kn \log n)$ Time. Journal of Computer Graphics Techniques, vol. 4, no. 1, 2015.
- [66] D. Holz and S. Behnke. Sancta Simplicitas – On the efficiency and achievable results of SLAM using ICP-Based Incremental Registration. Proceeding of the IEEE International Conference on Robotics and Automation (ICRA), pp. 1380–1387, 2010.

-
- [67] G. Grisetti, C. Stachniss, and W. Burgard. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.
- [68] G. H. Lee, F. Faundorfer, and M. Pollefeys. Robust pose-graph loop closures with expectation-maximization. *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.
- [69] P. M. Newman and K. Ho. SLAM Loop-Closing with Visually Salient Features. *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.
- [70] G. Dubbelman, I. Esteban and K. Schutte. Efficient trajectory bending with applications to loop closure. *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.
- [71] V. Kubelka, L. Oswald, F. Pomerleau, F. Colas, T. Svoboda, M. Reinstein. Robust Data Fusion of Multimodal Sensory Information for Mobile Robots. *Journal of Field Robot*, vol. 31, pp.1–27, 2014.
- [72] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots*, 2013
- [73] M. Ruhnke, R. Kummerle, G. Grisetti, and W. Burgard. Highly accurate 3D surface models by sparse surface adjustment. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2012.
- [74] J. Weingarten and R. Siegwart. EKF-based 3D SLAM for structured environment reconstruction. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005.
- [75] R. C. Luo and C. C. Lai. Enriched indoor map construction based on multisensor fusion approach for intelligent service robot. *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3135–3145, Aug. 2012.
- [76] B. Wei, J. Gao, K. Li and H. Chen. Navigation and Slope Detection System Design for Autonomous Mobile Robot. *The Ninth International Conference on Electronic Measurement & Instruments*, 2009.
- [77] A. Murarka and B. Kuipers. A Stereo Vision Based Mapping Algorithm for Detecting Inclines, Drop-offs, and Obstacles for Safe Local Navigation. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2009.
- [78] B. C. Akdeniz and H. I. Bozma. Local Terrain Mapping via 3D Laser Based Bubble Surfaces. *European Conference on Mobile Robots (ECMR)*, Sep. 2013.

-
- [79] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis, and R. Siegwart. A comparison of line extraction algorithms using 2d range data for indoor mobile robotics. *Autonomous Robots*, vol. 23, no.2, pp. 97-111, 2007.
- [80] C. Premebida and U. Nunes. Segmentation and geometric primitives extraction from 2d laser range data for mobile robot applications. *Robotica 2005 - Scientific meeting of the 5th National Robotics Festival, Coimbra, Portugal, Apr. 2005*.
- [81] G. A. Borges, and M.-J. Aldon. Line Extraction in 2D range images for Mobile Robotics. In: *Journal of Intelligent & Robotic Systems*, vol. 40, no. 3, pp.267-297. 2004.
- [82] S. Y. An, J. G. Kang, L.K. Lee, S.Y. Oh, Line-segment-Based Indoor Mapping with Salient Line Feature Extraction. *Advanced Robotics* vol.26, pp.437-460, 2012.
- [83] J. Lv, Y. Kobayashi, A. A. Ravankar, T. Emaru. Straight Line-segments Extraction and EKF-SLAM in Indoor Environment. *Journal of Automation and Control Engineering*, vol. 2, no. 3, pp. 270-276, 2014.
- [84] I. Markovsky and S. Van Huffel, Overview of total least squares methods. *Signal Processing*, vol. 87, pp. 2283-2302, 2007.
- [85] A. Garulli, A. Giannitrapani, A. Rossi, A. Vicino. Mobile robot SLAM for line-based environment representation. *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, 2005*.
- [86] S. T. Pfister, S. I. Roumeliotis, J. W. Burdick. Weighted Line Fitting Algorithms for Mobile Robot Map Building and Efficient Data Representation. *IEEE International Conference on Robotics and Automation (ICRA)*, pp.1304-1311, Sep. 2003.
- [87] A. Censi. An accurate closed-form estimate of ICP's covariance. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2007.
- [88] J. Lv, Y. Kobayashi, Y. Hoshino, T. Emaru. Slope Detection based on Orthogonal Assumption. *Proceedings of IEEE/SICE International Symposium on System Integration (SII)*, 2012.
- [89] L. Paz, J. Tardos, and J. Neira, Divide and conquer: EKF SLAM in $O(n)$. *IEEE Transactions on Robotics*, vol. 24, no. 5, 2008.
- [90] J. Weingarten and R. Siegwart. EKF-based 3D SLAM for structured environment reconstruction. *Proceedings of IEEE/RSJ International Conference on Intelligent*

-
- Robots and Systems (IROS), 2005.
- [91] P. Aggarwal, Z. Syed, X. Niu, and N. El-Sheimy. A Standard Testing and Calibration Procedure for Low Cost MEMS Inertial Sensors and Units. *The Journal of Navigation*, 61, pp. 323–336, 2008.
- [92] I. Skog and P. Händel. Calibration of a MEMS inertial measurement unit. *Proceeding of XVII IMEKO World Congress*, Sep.17-22, 2006.
- [93] H. Hou. Modeling inertial sensors errors using Allan variance. M.S. thesis, MMSS Res. Group, Dept. Geomatics Eng., Univ. Calgary, Calgary, AB, Canada, UCGE Rep. 20201, Sep. 2004.
- [94] B. Fang, W. Chou, and L Ding. An Optimal Calibration Method for a MEMS Inertial Measurement Unit. *International Journal of Advanced Robotic Systems*, 11:14, 2014.
- [95] [https://en.wikipedia.org/wiki/Noise_\(signal_processing\)](https://en.wikipedia.org/wiki/Noise_(signal_processing))
- [96] E-H. Shin and N. El-Sheimy. A New Calibration Method for Strapdown Inertial Navigation Systems. *Zeitschrift für Vermessungswesen*, vol. 127, pp. 1-10, 2002.
- [97] C. Ren, Q. Liu, and T. Fu. A Novel Self-Calibration Method for MIMU. *IEEE Sensors Journal*, vol. 15, no. 10, Oct. 2015.
- [98] D. Tedaldi, A. Pretto, and E. Menegatti. A Robust and Easy to Implement Method for IMU Calibration without External Equipments. *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [99] J. Rohac, M. Sipos, and J. Simanek. Calibration of Low-cost Triaxial Inertial Sensors. *IEEE Instrumentation & Measurement Magazine*. vol. 18, no. 6, pp. 32-38, Dec. 2015.
- [100] J.-O. Nilsson, I. Skog, and P. Händel. Aligning the Forces-Eliminating the Misalignments in IMU Arrays. *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 10, Oct. 2014.
- [101] N. El-Sheimy, H. Hou, and X. Niu. Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 1, pp.140–149, Jan. 2008.
- [102] D. Jurman, M. Jankovec, R. Kamnik, M. Topič. Calibration and data fusion solution for the miniature attitude and heading reference system. *Sensors and Actuators A: Physical*, vol. 138, no. 2, pp. 411-420, 2007.
- [103] X. Yun, E. R. Bachmann, H. Moore IV, and J. Calusdian. Self-contained Position Tracking of Human Movement Using Small Inertial/Magnetic Sensor Modules. *IEEE International Conference on Robotics and Automation (ICRA)*, 2007.
- [104] Y. S. Suh. Orientation Estimation Using a Quaternion-Based Indirect Kalman Filter With Adaptive Estimation of External Acceleration. *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 12, pp.3296-3305, 2010.
- [105] <https://github.com/xioTechnologies/Gait-Tracking-With-x-IMU>