Autonomous Robotic Strategies for Urban Search and Rescue

Kunjin Ryu

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mechanical Engineering

Tomonari Furukawa, Committee Chair
John Ferris
Dennis Hong
Andrew Kurdila
Craig Woolsey

October 12, 2012
Blacksburg, VA

Keywords: Simultaneous Localization and Mapping, Semi-autonomous Robot Navigation, Cooperative Search and Tracking

Autonomous Robotic Strategies for Urban Search and Rescue

Kunjin Ryu

ABSTRACT

This dissertation proposes autonomous robotic strategies for urban search and rescue (USAR) which are map-based semi-autonomous robot navigation and fully-autonomous robotic search, tracking, localization and mapping (STLAM) using a team of robots. Since the prerequisite for these solutions is accurate robot localization in the environment, this dissertation first presents a novel grid-based scan-to-map matching technique for accurate simultaneous localization and mapping (SLAM). At every acquisition of a new scan and estimation of the robot pose, the proposed technique corrects the estimation error by matching the new scan to the globally defined grid map. To improve the accuracy of the correction, each grid cell of the map is represented by multiple normal distributions (NDs). The new scan to be matched to the map is also represented by NDs, which achieves the scan-to-map matching by the ND-to-ND matching. In the map-based semi-autonomous robot navigation strategy, a robot placed in an environment creates the map of the environment and sends it to the human operator at a distant location. The human operator then makes decisions based on the map and controls the robot via tele-operation. In case of communication loss, the robot semi-autonomously returns to the home position by inversely tracking its trajectory with additional optimal path planning. In the fully-autonomous robotic solution to USAR, multiple robots communicate one another while operating together as a team. The base station collects information from each robot and assigns tasks to the robots. Unlike the semi-autonomous strategy there is no control from the human operator. To further enhance the efficiency of their cooperation each member of the team specifically works on its own task.

A series of numerical and experimental studies were conducted to demonstrate the applicability of the proposed solutions to USAR scenarios. The effectiveness of the scan-to-map matching with the multi-ND representation was confirmed by analyzing the error accumulation and by comparing with the single-ND representation. The applicability of the scan-to-map matching to the real SLAM problem was also verified in three different real environments. The results of the map-based semi-autonomous robot navigation showed the effectiveness of the approach as an immediately usable solution to USAR. The effectiveness of the proposed fully-autonomous solution was first confirmed by two real robots in a real environment. The cooperative performance of the strategy was further investigated using the developed platform- and hardware-in-the-loop simulator. The results showed significant potential as the future solution to USAR.

Acknowledgement

To begin with, I would like to thank my advisor Professor Tomonari Furukawa for his guidance and encouragement during my PhD study. He has been a supportive advisor who has provided me with a great deal of academic advice and shared his ideas and philosophy as a researcher. Professor Furukawa, I have learned a lot from you and cannot say thank you enough. I also would like to express my gratitude to every committee member, Professors John Ferris, Dennis Hong, Andy Kurdila and Craig Woolsey who have showed their interests in my research work with valuable feedbacks.

I am indebted to Professors Kenzo Nonami and Wenwei Yu for their time and advice during my visit to Chiba University for collaborative research. Thanks also to Professor Mark Haley who arranged the visit to Chiba University and other meetings with a few research institutes and companies. I am thankful to get to know all the current CMS lab members and some of CMS alumni. It was very enjoyable to have interesting discussions on different topics from which I sometimes came up with good ideas. Especially I thank Xianqiao Tong and Dr. Lin Chi Mak who I worked with for the international competition, MAGIC 2010, and Dr. Jan Wei Pan for helpful advice when I was preparing the preliminary exam.

I must thank many people who made my time enjoyable and rewarding. Thanks to friends who shared good memories with me in Blacksburg since I first came to Virginia Tech. Special thanks to Jiyoun for consistent support and encouragement. I also thank ME graduate students who worked at the Institute for Advanced Learning and Research in Danville and the Furukawa family for their warmth and kindness.

I owe a lot to my parents and brother; they have always been there and have supported in my decisions with patience. Thank you.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The development of autonomous robotics has dramatically extended the applicable scope of robots in the past thirty plus years. In early years, robots were not given a high level of autonomy, but instead they were usually controlled by human operators. As technology improved, robots have come to be able to perform much more complicated jobs, both while stationary and while moving. One example of the latter is rescue in urban disaster areas. In such environments, robots require a high level of autonomy in order to operate properly, since they usually have to complete multiple tasks at the same time, such as understanding and exploring environments, detecting and avoiding danger, and identifying and retrieving victims. These challenging problems have received much attention leading to research focused on applicable solutions to rescue in urban disasters.

This dissertation presents autonomous robotic strategies for urban search and rescue (USAR) scenarios. For the purpose of this dissertation, the term autonomous robotics is defined as the capability to work without human control. Although a variety of subproblems exist in USAR, this dissertation focuses on possible solutions to searching for and tracking objects of interests and their prerequisite technique of robot localization in the environment. In this Chapter, the background leading up to the recent interest in robotic applications in USAR is briefly explained, followed by the objective of this dissertation. The

approach to achieve the objective is then presented, and the original contributions are summarized. Finally, the contents of the remaining chapters of this dissertation are outlined.

1.1 Urban Search and Rescue

Every year there are a lot of disasters all over the world which could cause widespread destruction and seriously damage human health. They can be either natural or man-made disasters, and to some degree some of them are sometimes predictable. However, it is still nearly impossible to perfectly avoid these disasters, which results in urban search and rescue (USAR) scenarios. In general, when rescuers are committed to a USAR situation, there are lots of limitations which can reduce the efficiency of rescue. In some cases, a wide area needs to be searched and a part of the disaster area may not be accessible. Additionally, communication issues and other technical problems can arise after the disasters, which can also make the problem of search and rescue more difficult. Most importantly, the rescuers have to handle collapsed buildings, toxic chemicals, or any sort of explosive materials, which means they are seriously exposed to danger.

In order to more effectively deal with these problems, robotic technologies have been recently introduced to USAR scenarios. Although current rescue robots are nothing more than assistants and cannot completely substitute human rescuers, there is no doubt that robotic applications to USAR will be used more actively in the near future. One of major benefits of using robots in such scenarios is to prevent unnecessary sacrifice of lives of rescuers. Other advantages including reduced personnel requirements and reduced fatigue can dramatically improve the efficiency in search and rescue. Since robots can move much faster than human beings, the area that the robots can explore is much larger. Moreover, the robots carry a lot of different sensors which allow accurate search for possible victims: a microphone can hear sounds of human presence in the ruins; a thermal camera can detect body heat; and a vision camera can search for colors distinctive from the gray dust on the ruins.

Despite the potential of robotic technologies in USAR, actual applications are limited. This is mainly because technology is still not mature. However, there have been continuous efforts on robotic search and

rescue and some of them were fairly successful. The recent series of disasters has increased awareness of the possibilities of robots for assistance, and it is expected that robotic solutions play more important roles in future USAR.

1.2 Objective

The objective of this dissertation is to develop autonomous robotic strategies which can be applicable to search and rescue in urban disaster areas and to demonstrate the functionality of these solutions with experimentation both in virtual and real environments.

1.3 Approach

When developing and applying a new robotic solution to a real problem, there is always a dilemma between the need for proven technologies and the need for advanced but immature technologies. Both are important in that proven technologies guarantee robust behaviors of robots whereas the advanced technologies may solve problems which cannot be solved by proven technologies. In consideration of both needs, two autonomous strategies are proposed for USAR in this dissertation. The first strategy, map-based semi-autonomous robot navigation via tele-operation, is an immediately usable solution focusing more on the robustness of the solution. On the other hand, the second strategy, fully-autonomous search and tracking, is a future solution that emphasizes its significant potential in USAR.

In order to achieve this objective, it is obvious that robots to be used for the two strategies are mobile, and that an accurate solution to robot localization in the environment therefore is necessary. The simultaneous localization and mapping (SLAM) technique, the state-of-the-art proven technique, can be the most suitable solution to robot localization, and the development of the two strategies starts from the improvement of SLAM technique. The improvement is achieved by using an accurate range sensor and matching each new scan image to a globally defined map. The map is represented as normal distributions (NDs) on a grid space and accurately updated with time. Due to this global correction, each new scan image can be accurately mapped onto the grid space and again contributes to update the grid map after it

is converted to a set of NDs. Since the grid space is globally fixed and NDs are registered to grid cells of the map, an incremental map can be easily built regardless of the size of the environment the robot explores.

The semi-autonomous robot navigation strategy heavily relies on the proven SLAM technique. In this approach, the level of autonomy of the on-site robot is limited and the human operator at a distant location controls the robot given the map from the robot. Since the human operator makes decisions, the strategy provides a robust but possibly limited solution to USAR. For the enhancement of the limitation, other functionalities such as computer-aided waypoint search, which computes the next waypoint based on the trajectory of the robot, and semi-autonomous return, which enable the robot to safely return to home position in case of communication loss, are applied to the strategy. Additionally, a graphical user interface (GUI) is designed where the human operator can control the robot by a simple mouse clicking.

The fully-autonomous search, tracking, localization and mapping (STLAM) introduces a team of robots which cooperatively search for and track objects of interest (OOIs) while each robot autonomously performs SLAM and explores the environment. Each robot is wirelessly connected to other robots as well as the base station, and they all share information. Search for the OOIs in an unknown environment is achieved by the area coverage method, and the efficiency of the search method is improved by frontier-based exploration. When the OOI is detected, the robot keeps tracking the position of the OOI, so that everyone can realize the existence of the OOI. In case that the OOI is mobile, the robot computes the position of the OOI by prediction and correction using the EKF.

1.4 Original Contribution

The principal contributions of this dissertation are enumerated as follows:

- The development of a unique grid-based scan-to-map matching technique for the SLAM problem that corrects the estimation error by matching scan images from a laser range finder (LRF) to the globally maintained grid map.

- The development of a multi-ND representation of the global map and the achievement of the scan-to-map matching throughout the novel ND to ND matching.
- The development of map-based semi-autonomous robot navigation using tele-operation which allows three-dimensional environment monitoring and autonomous return of the on-site robot to the home position as needed.
- The development of fully-autonomous and cooperative STLAM using multiple robots, where the robots cooperatively explore the environment, search for and track OOIs.
- The development of a simulator, the so-called Platform- and Hardware-in-the-loop Simulator (PHILS), that allows the evaluation of cooperative performance of a team of robots.

1.6 Publications

To date, components of the dissertation have been presented in the following publications:

[1] **Kunjin Ryu**, Tomonari Furukawa and Gamini Dissanayake, "Grid-based Scan-to-Map Matching for Accurate Simultaneous Localization and Mapping –Theory and Preliminary Numerical Study–," *2013 IEEE International Conference on Robotics and Automation (ICRA)*,Karlsruhe, Germany, submitted

[2] **Kunjin Ryu**, Tomonari Furukawa and Gamini Dissanayake, "Grid-based Scan-to-Map Matching for Accurate Simultaneous Localization and Mapping," *Autonomous Robots*, submitted

[3] **Kunjin Ryu**, Tomonari Furukawa, Jaime Valls Miro and Gamini Dissanayake, "Map-based Semi-Autonomous Strategy for Urban Search and Rescue," *International Journal of Intelligent Unmanned Systems*, accepted

[4] **Kunjin Ryu**, Tomonari Furukawa, "Virtual Field Testing for Performance Evaluation of Cooperative Multiple Robots," *The International Conference on Intelligent Robotics and Applications*, Montreal, Canada, Oct. 2012

[5] **Kunjin Ryu**, Tomonari Furukawa, "A LRF-based Teleoperated Navigation Method," *The International Conference on Intelligent Unmanned Systems*, Chiba, Japan, 2011

[7] **Kunjin Ryu**, Xianqiao Tong, Tomonari Furukawa, "The Platform- and Hardware-in- the-loop Simulator for Multi-Robot Cooperation," A Workshop on Frontiers of Real-World Multi-Robot Systems, Durham, NC, USA, 2011

[8] Tomonari Furukawa, Lin Chi Mak, **Kunjin Ryu**, Xianqiao Tong and Gamini Dissanayake, "Bayesian Search, Tracking, Localization and Mapping: A Unified Strategy for Multi-task Mission," *INFORMS 2011 Annual Meeting*, November 13-16, 2011, Charlotte, USA, 2011

[9] Tomonari Furukawa, Lin Chi Mak, **Kunjin Ryu**, Xianqiao Tong, "The Platform- and Hardware-in- the-loop Simulator for Multi-Robot Cooperation," *Proceedings of the 2010 Performance Metrics for Intelligent Systems (PerMIS'10) Workshop*, Baltimore, USA, 2010

1.7 **Outline of the Dissertation**

This dissertation is organized as follows:

- Chapter 2 reviews previous efforts on SLAM as a basic technique for developing robotics solutions to USAR. SLAM approaches are classified in terms of map representations and estimation methods. Advantages and disadvantages for each approach are briefly explained and further discussions based on the important issues provided in this introductory chapter are presented
- Chapter 3 describes an overview of the scan matching as one of the most relied-upon technique for the SLAM problem. Two specific scan matching techniques, which are the most associated with the scan-to-map matching, are formulated, and dead reckoning results by these techniques are presented.
- Chapter 4 presents the unique grid-based scan-to-map matching technique which achieves accurate SLAM. The multi-ND representation of the grid map is first described. The scan-to-map matching via the ND-to-ND matching and the update of the grid map are then presented. A number of experimental results in simulated and real environments are presented to investigate the performance of the scan-to-map matching and to demonstrate the applicability of the technique in real SLAM scenarios.
- Chapter 5 presents the semi-autonomous robotic solution for USAR based on the map accurately created by the scan-to-map matching technique. The concept of the semi-autonomous strategy throughout tele-operation together with the computer-aided waypoint search is proposed. The

design of the GUI which is used for tele-operated navigation and semi-autonomous return in case of communication loss are also described.

- Chapter 6 presents the fully-autonomous and cooperative search, tracking, localization and maping solution for USAR scenarios. The concept and theoretical formulations of search and tracking using multiple robots as a team are explained. The solution is validated by integrating it into two real robots and testing it in a real environment. Further evaluations on cooperation between the robots are investigated within the developed platform- and hardware-in-the-loop simulator.
- Chapter 7 summarizes the original contributions of the research presented in this dissertation and discusses areas for potential future work.

Chapter 2

Simultaneous Localization and Mapping

A robot commences its action by first understanding its own location and surroundings when it is placed in a totally unknown environment. Simultaneous localization and mapping (SLAM) is a problem of building a map of the environment while at the same time localizing the robot in the map. A solution to this problem is given by iteratively observing the surrounding environment and associating a new observation containing some objects to the previous observation containing the same objects. Since the solution does not rely on the global positioning system (GPS) for robot localization, SLAM techniques allow the robot to work in GPS-denied environments. The SLAM problem became even more important when the robot needs to autonomously explore the environment, and it is obvious that the ability of the robot is extremely limited without an accurate solution. [1, 2, 3, 4].

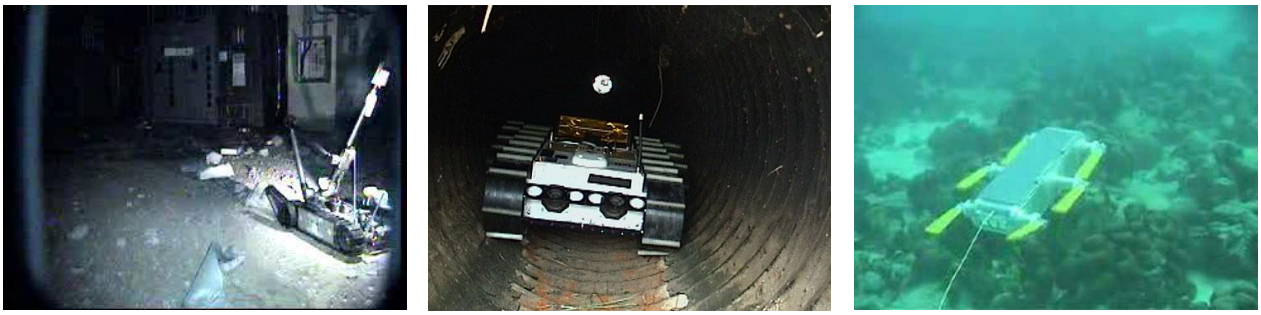
In the SLAM problem, a map is defined as a visual representation of an environment and used as a reference in order for the robot to determine its position within the environment. The map can be defined in different ways and the data association method is heavily related to the type of the map. This chapter reviews the past contributions concerned with the SLAM problem in terms of the type of the map and underlying estimation methods. Section 2.1 describes the motivation of the need for SLAM as an

alternative for the traditional robot localization techniques. Section 2.2 categorizes maps into four types and the SLAM techniques are further summarized according to the estimation methods in Section 2.3.

2.1 Robot Localization

Consider a scenario either of a natural disaster or a man-made disaster where the site is highly unstructured and dangerous. There are limitations on what human rescuers can do, thus, the use of one or multiple robots onsite could accelerate the efficiency of rescue. Now the question is, "What is the most fundamental and important task for mobile robots to operate effectively and intelligently in this scenario?" For mobile robots understanding surroundings and their own locations is a prerequisite condition to explore environments where the robots are. The solution to this problem is known as robot localization, which truly enables mobile robots to explore the environments and complete duties without getting lost. This robot localization usually comes with the need for a map of the environment since otherwise robot localization might not be a complete solution for the exploration. In other words, if robot localization and mapping are not taken into account, the ability of the mobile robot becomes extremely limited.

It is possible that the robot can explore an environment without the ability of creating a map of the environment if the *a priori* map exists. In such a case, the robot only needs to detect known landmarks to localize itself in the environment. However, in most cases, *a prior* maps are not available, robots thus have to construct maps by themselves in order to work properly within the given environments. The global positioning system (GPS) is an option for robot localization, however, the accuracy is not good enough for certain scenarios and it cannot be useful for indoor, underground, underwater environments (Figure 1). Another option for robot localization is dead reckoning. It computes the pose of the robot from the previously determined robot pose and the robot motion. The motion of the robot can be directly estimated by sensors such as an odometer, inertial sensors, or it can be predicted by computing the relative position between two consecutive observations of the environment. Since robot localization by



(a) Indoor (b) Underground (c) Underwater

Figure 1. Examples of GPS-denied environments (source: google images, under fair use, 2012)

dead reckoning is a sequential process of estimating the robot motion between a time period, errors which may come from the sensors, the imperfect derivation of motion model of the robot, and bad estimations of the robot motion from observations, can be accumulated with time. Once these errors are accumulated, there is no way of correcting the errors without a post processing which generally requires a heavy computational load and avoids real time robot exploration.

Simultaneous localization and mapping (SLAM) is a problem of creating a map of the environment and simultaneously localizing the robot in the map. The solution to the SLAM problem, one of the most widely investigated subfield in robotics, is regarded as a better approach to deal with robot localization than other approaches using the GPS and dead reckoning. This is because the robot solving the SLAM problem keeps track of its current pose and builds the map at the same time, which enables the robot to correct its pose from the map and vice versa. It is therefore obvious that accurate mapping can be achieved only when robot localization is correct, and that the quality of the estimation of the robot pose is also interactively linked to the map accuracy.

Since SLAM is an iterative process utilizing one or more sensors, there are several important issues underlying it. Once the robot obtains sensor readings at any position in the map, a way of correlating the current observation to any past observations has to be defined. It is called data association and plays an important role because the robot pose can be well estimated only when data association is correct. For successful data association, a large number of techniques have been proposed, and the type of data

extracted from the observation to be used for data association can vary by the data association techniques. One common type of data is a set of features, or landmarks. Some features, especially when the observation is obtained by a camera, can also be distinguished by their colors. Another common type of data is an unprocessed scan image obtained from range sensors such as an ultrasonic sensor and a laser range finder (LRF). This type of data is generally composed of a set of points that describe relative positions from the sensor to detected objects. Any specific data association technique might not perform well in some conditions, while the other techniques can be good solutions to data association. In this sense, the SLAM capability can be improved by using multiple data association techniques together.

Computational efficiency and noises are also important issues that need to be considered for solving the SLAM problem. A mobile robot is equipped with multiple sensors for SLAM and the net amount of data and computations are thus huge. If the environment is very large, an efficient way of handling such big data is a key to real-time processing which is necessary to the mobile robot. Moreover, computational efficiency is important since it can improve the accuracy of SLAM by having more computations given a computational ability. Meanwhile, the accuracy of SLAM can significantly drop when there exist large noises. Possible sources of the noises are hardware such as sensors and the environmental conditions including the reflection of light. Although it is not easy to identify and estimate the noise in many cases, the noise needs to be removed, or at least reduced, in order to achieve certain level of accuracy in solving the SLAM problem.

Other challenges include dynamic environments and objects, and closing the loop when the robot comes back to the previously explored area. Since there are so many related, SLAM approaches do not always focus on every problem at the same time, but they address their own priorities in some aspects.

2.2 Maps

This section reviews fundamentals of most common ways of representing the environment, an occupancy grid map, a topological map, a feature-based map, and a scan-based map. This includes definitions and properties of the maps and description on localization using the maps is also presented.

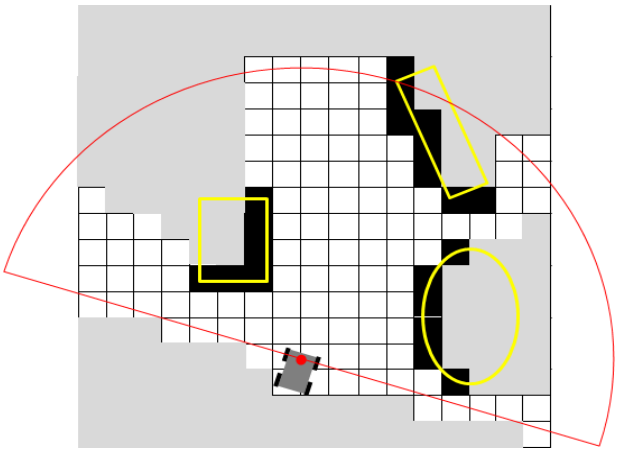


Figure 2. An example of occupancy grid map

2.2.1 Occupancy Grid Map

An occupancy grid map (Figure 2) is one of the most common ways of creating a metric map in robotic mapping [5, 6, 7, 8]. It is a probabilistic representation of the environment by grid cells with binary random variables. Due to Bayes theorem underlying the algorithm, occupancy mapping can efficiently reconstruct an environment from noisy and uncertain sensor readings in real time. In occupancy grid map, the mapping space is evenly divided into either two-dimensional or three-dimensional grid cells each of which has the probability indicating if it is occupied, open, or not explored. On this space, occupancy mapping needs to predict the posterior probability of the map given the history of the sensor measurements and that of the robot pose, where the robot pose is assumed to be known. At the initial state, since there is no prior information on the environment, every grid cell over the entire space remains as being unexplored. As the mapping space gets large and the number of grid cells increases, the dimensionality of this problem becomes extremely high. To avoid this computational issue, the problem is separated into small problems that deal only with a single cell without loss of generality.

One of the major benefits of the occupancy grid representation is that it can be very effectively used for robot navigation [9, 10, 11] including path planning [12] and collision avoidance [13] due to its simple and clear classification in defining the environment. The occupancy grid map has been built by different

sensors such as sonar sensors [5, 14], LRFs [15], and stereo vision [16], and created in three-dimensioanl space [17, 18]. In addition, the occupancy grid representation can make the map have multiple different resolutions even on the same environment. In other words, the resolution of the map is tunable and the resolution of the grid map can be dependent on the complexity of the environment [19].

On the other hand, one of drawbacks is that computational efficiency significantly drops when there is a need for an accurate mapping. More importantly, the map does not have a proper representation of uncertainties of the sensor and the vehicle, therefore, the map sometimes leads to divergence in robot localization. The occupancy grid map can be easily built, but at time same time the update is not as easy as the creation of the map. Although occupancy grid mapping has been used in changing environments [20], it is not generally suitable for such environments.

2.2.2 Topological Map

Contrary to the occupancy grid map, a topological map shown in Figure 3 does not depend on metric measurements. The map is mainly composed of nodes and edges which maintain the relationships between nodes. Nodes are abstracted models which are extracted from environmental entities. While the nodes refer to specific locations in the environment, the edges provide information on connections, or paths, between nodes. Since the topological map is a conceptual image with lack of scale, distances and directions describing the map are different from those in real environment. In the topological map, it is assumed that each node has to be somehow recognizable and unique from other nodes so that the robot is able to distinguish it from other nodes. This is the most important and difficult process, since the robot gets lost if it fails either to recognize a place or to match the place to the correct node in the map. Once the robot gets the location, the identification of the place in the map can be done by associating the observation taken at the location with node descriptions. In order for the location to be recognizable and distinguishable in the map, vision-based techniques [21, 22] as well as LRF-based methods [23] are used.

As an *a prior* map, a topological map can be used for localizing the robot. However, there exist limitations on doing this, because of the difficulty of place recognition mentioned above and the fact that

Figure 3. An example of a topological map (source: google images, under fair use, 2012)

the map does not contain metric parameters which precisely describe the environment. In this sense, the localization based on the topological map is not regarded as SLAM. On the other hand, it is useful for robot navigation where a series of nodes functions as waypoints that the robot needs to sequentially visit [24].

2.2.3 Feature-based Map

A feature-based map, or simply a feature map, describes the environment by a collection of features which can be different types of geometric models such as points, lines, curvatures, and any arbitrary shapes (Figure 4). Under the assumption that the robot can perfectly recognize features from the environment and their positions are known, the feature map can be efficiently used for the robot localization problem. Given information on all features in the map, the robot can calculate its current pose by obtaining a set of observations. Throughout the process of feature extraction, the SLAM techniques based on the feature map [25, 26] recognizes detectable features in its field of view and associates these features with features in the map. The observed features are added to the map using the pose of the robot which is at the same time estimated from the feature map. Similar to the occupancy grid map, the feature map is also a subdivision of metric maps where features are in the two-dimensional or three-dimensional Cartesian coordinate system. However, compared to the occupancy grid map, the feature map manages

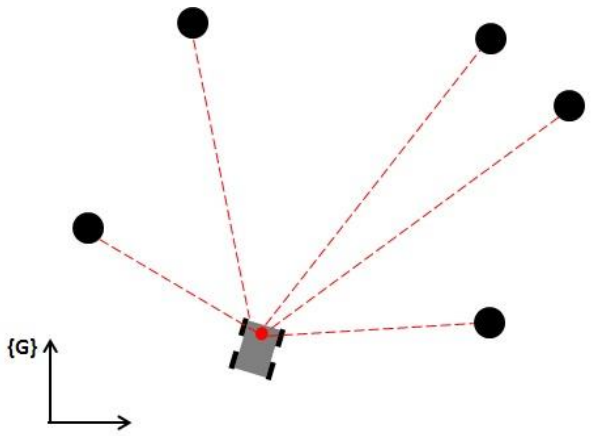


Figure 4. An example of feature-based map

data representing the environment more efficiently because it only stores features. This advantage over the occupancy grid representation can be more emphasized when the environment are very large, but can be represented by limited number of features.

A classical technique for associating features was to use the gated nearest neighbor (NN) algorithms [27, 28, 29]. These approaches compute the distance of a feature of the new observation to every existing feature and select the existing feature of the minimum distance as the corresponding feature. The approaches introduced a breakthrough to the feature-based SLAM, but since each new feature in the NN algorithms corresponds to a single existing feature, incorrect data association might be caused by spurious features. Other data association techniques include signature string matching [30, 31], and batch correspondence methods [32, 4, 33], which handle spurious features more robustly by adding search algorithms. Since the feature map only considers the extracted data, existing filters such as the EKF and the Rao-Blackwellized particle filter have been applied to maintain the feature map, which will be detailed in the following section. For instance, FastSLAM [34] maintained multiple candidates of existing features for each new feature using the Rao-Blackwellized particle filter and demonstrated its effectiveness in several real environments [35, 36].

Although the feature map has been widely used for the SLAM problem, there exist some drawbacks due to the inherent problem of the feature-based representation. The feature map does not contain any geometric information on the area that is not represented by features, so it cannot be suitable for robot navigation, path planning, and obstacle avoidance. In real environments, if they are not well defined by a set of features, the feature map cannot be used for solving the SLAM problem. This issue is related not only to environmental conditions but also to feature extraction techniques. To enhance the ability of modeling accurate features, a number of algorithms such as RANdom SAmple Consensus (RANSAC) [37], iterative end point fit (IEPF) technique [38] and split and merge [39, 40] have been introduced. Even though the environment can be easily described by features, the feature map has a problem in data association. This is more important issue, since successful SLAM is heavily influenced by the successful association of the new observation to the map. The false associations always cause wrong pose estimation of the robot and accordingly the accuracy of the feature map is also degraded. The increase of uncertainties of both robot and the map becomes exponentially large as data association keeps failing.

The feature-based data association has been widely used for the SLAM problem and demonstrated its effectiveness in several real environments. However, its capability highly relies on the success of the feature extraction from the observation. Moreover, even if features are well modeled and extracted, associating the exact same features is not an easy process. The chance of inappropriate data association increases when different features in the environment look similar.

2.2.4 Scan-based Map

Due to the superiority of the laser sensors in accuracy over other vision sensors, the scan-based map has come to be popular in SLAM. Similar to the occupancy grid map and the feature-based map, the scan-based map is also a metric map, but it is usually composed of a collection of raw scan images each with its own pose of the robot as shown in Figure 5. It can be a specific type of the feature-based map, where the scan images are considered as the features. Unlike the feature-based map, the scan-based map does

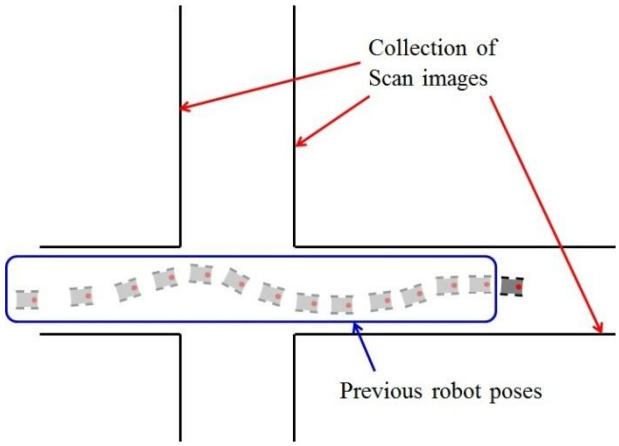


Figure 5. Scan based map composed of raw scan images and robot poses

not require additional feature extraction processes which can improve both computational efficiency and data association accuracy. Since raw scan images are directly used as observations, there is no loss of data, which enables the scan-based map to fully describe the environment regardless of the existence of features. However, when the environment is large and a lot of scan images are thus required to be stored. This results in the increase of memory consumption and a sacrifice in computational efficiency.

In comparison to the feature map, the scan-based map is relatively free from data association problem since there needs no explicit feature model to define. In scan-based SLAM, data association is commonly achieved by a technique called scan matching, or scan-to-scan matching. In order to match the new scan to the past scans using the scan-to-scan matching techniques for achieving SLAM, a number of approaches have been proposed with the development of additional strategies. Early efforts include the work of Lu and Milios [41] which performed the matching of the new scan to the previous scan and further matched all the scans by storing the past scans. This globally consistent matching approach has been successfully implemented together with different scan matching techniques [42, 43, 44]. Thrun et al. [45] used the expectation maximization (EM) algorithm that finds the best matching past scan to the new scan from all the past scans. The matching of the new scan to all the past scans is then achieved by the scan-to-scan matching of the new scan to this best past scan. Although they have demonstrated

capabilities in accurate matching, the approaches could still see accuracy issues without a loop closure as they do not either implement a powerful scan-to-scan matching or utilize all the past scans. Due to the need for matching to all the past scans for the best accuracy. Bosse et al. [46] introduced a subspace-to-map matching technique where the new scan is matched to all the past scans of a subspace of concern with any scan-to-scan matching technique and the subspaces are subsequently associated to each other for global mapping. This technique achieves the matching of the new scan to all the past scans, but the accuracy could still drop since the new scan points not in the subspace are not matched to the past scans.

2.3 Estimation Method

This section briefly reviews SLAM techniques according to the estimation methods. The most popular approaches include using the EKF, the particle filter, and maximum likelihood which is separated according to whether a probabilistic method is implemented or not.

2.3.1 EKF SLAM

In EKF approach [26, 47, 48, 49, 50], the system noise is defined as the Gaussian distribution and non-linear models are linearized so that the Kalman filter can be used. The EKF can be applied to any sensor readings as long as features exist in the observation and they are recognizable. Fundamental formulation is presented in Appendix 1. Especially when the robot model is close to the linearity, the uncertainty model by the EKF can produce a solid map [26]. The robot pose estimation thus becomes reliable after observing features repeatedly since the positions of landmarks become more certain with multiple observations and they are correlated to the robot pose estimation. The uncertainty of each feature is also correlated to other features, which enables the robot to localize itself precisely within the environment.

A major problem underlying the EKF SLAM is that the linearization of inherent nonlinearities of both the vehicle motion and the observation models causes an inconsistent performance. In such a scenario that the true uncertainty of the robot exceeds a limit, a large error in the map results in inconsistency in mapping. There is higher possibility that this happens in large-scale environments and

inconsistency of mapping algorithm in the environments is unavoidable [51]. The unscented Kalman filter deals better with nonlinearities in the motion model of the robot [52]. A mapping algorithm, *Robocentric Map Joining*, that limits the level of uncertainty in the incremental map has been introduced to improves consistency of the EKF SLAM [53]. Based on split covariance intersection (SCI), Julier and Uhlmann [54] developed consistent, constant time algorithm maintaining an extremely large map in the global frame.

Despite extraordinary efforts in the EKF SLAM, the performance of the EKF is still heavily affected by how to define and extract features from the observations. In order to successfully identify features for the EKF SLAM, the features are frequently enforced to be sparse [55, 56], which allows the positive feature identification. The sparse features are usually defined when the features are extracted from cameras rather than the LRF, which can be effective in both successful identification of features and computational efficiency. However these sparsely distributed features cannot accurately describe the environment, furthermore, they can be more sensitive to misassociation between features since the number of features is relatively limited.

The EKF estimation for the SLAM problem is theoretically proven techniques and has shown its applicability to the SLAM problem. However, the underlying properties of the EKF such as the linearization error and the Gaussian assumptions for the errors are not always valid in real SLAM scenarios.

2.3.2 Particle Filter SLAM

As an alternative solution to the EKF SLAM, efficient approaches based on particle filtering have been introduced [57, 58, 59, 60]. In these approaches, each particle in the RBPF represents a possible robot trajectory and a map. To learn accurate grid maps Eliazar and Parr [61] and Hähnel et al. [62] utilized the RBPF with additional approaches. In the first work, Eliazar and Parr described a new map representation called distributed particle (DP) mapping, which enables maintaining and updating hundreds of candidate maps and robot pose efficiently. Unlike other methods that require feature extraction and data association

process, this approach does not rely on the presence of landmarks, but uses the LRF only. In the second work, an improved motion model was presented, which reduces the number of required particles. Base on the work of Hähnel et al., Howard presented an approach to learn grid maps with multiple robots [63]. The focus of this work lies in how to merge the information obtained by the individual robots and not in how to compute better proposal distributions.

The common problem of using the RBPF is its computational complexity to achieve high accuracy in SLAM. In the context of the feature based SLAM, Montemerlo et al. [34] presented a RBPF that uses a Gaussian approximation of the improved proposal. This Gaussian is computed for each particle using the Kalman filter that estimates the pose of the robot. Each particle possesses N low-dimensional EKFs, one for each of the N landmarks. In this case, the computational complexity is $\mathrm{O}(NM)$ where M is the number of particles in the particle filter. Updating this filter requires $\mathrm{O}(M \log N)$ times, with or without knowledge of the data associations. However, this approach can be used only when the map is represented by a set of features and when the error is assumed to be Gaussian. To improve the computational efficiency, Grisetti et al. [64] proposed an approach to reduce the number of particle by considering not only the robot movement but also the most recent observation.

Other contributions include using the RBPF in combination with the camera-based vision SLAM .Elinas et al. [65] presented a stereo vision SLAM using RBPF that landmark estimation are derived from stereo vision and motion estimates are based on sparse optical flow. Hu et al. [66] also presented a vision-based SLAM with implementation of RBPF, which is able to track artificial landmarks such as multi-colored cylinders.

The RBPFs have been introduced as another effective way of estimation methods in the SLAM problem. Unlike the EKF SLAM, the particle filter based SLAM techniques do not suffer from the linearization error or the Gaussian assumptions. However, in order to attain a certain level of accuracy in the estimation process, they have to maintain a large number of particles.

2.3.3 Maximum Likelihood SLAM

Maximum Likelihood SLAM approaches compute most likely scan or map given the history of sensor readings by optimizing the objective function which can vary by the approach [67, 68, 70]. In order to do this, the robot poses are regarded as nodes and relations between nodes, or the network, are required to be constructed. Lu and Milios [67] applied the least mean square error together with the creation of the network between robot poses. In this approach, when the robot returns to a previously explored region, all the networks are globally built at the same time. Gutmann et al. [68] proposed an effective way for constructing such a network and for detecting loop closures, while running an incremental maximum likelihood algorithm. When a loop closure is detected, a global optimization on the network of relation is performed. Hähnel et al. [69], proposed an approach which is able to track several map hypotheses using an association tree. Since these approaches correct the robot pose at once, the estimation of the robot pose is not accurate until the nodes are connected to one another.

Olsen et al. [70], on the other hand, proposed a graph-based approach that updates the network locally by applying stochastic gradient descent to minimize the error. By doing so, the estimation of the robot can be accurately maintained without the global correction. Grisetti et al. [71] extended the work of Olsen by introducing a tree structure, which accelerates the speed of convergence. Kaess [72] also proposed the incremental smoothing and mapping which allows real-time SLAM in large environments. This approach utilized a QR decomposition to correct the poses of the nodes in the network can be efficiently corrected by back substitution. Another real-time SLAM approach is using the so-called Treemap algorithm which ignores the weak correlations between distant locations [73] .

Maximum likelihood SLAM techniques can provide an accurate solution to the SLAM problem. However, since they iteratively perform their estimations, low computational efficiency is generally their weakness.

2.4 Summary

This chapter has reviewed the early efforts on the SLAM problem in terms of the type of the map and the underlying estimation methods. The map can be roughly classified into the occupancy grid map, the topological map, the feature-based map, and the scan-based map. The occupancy grid map represents the environment by grid cells each of which has the probability indicating if it is occupied, open, or not explored. Since the occupancy grid map contains the global information on the environment, it is useful for robot navigation. The topological map is not a metric map and composed of nodes and edges. The map focuses on the special relations between nodes. The feature-based map maintains uniquely defined features in the Cartesian coordinate system. A popular method to maintain the features is to use the EKF, and the performance of the feature-based SLAM heavily relies on the feature extraction capability. The scan-based map is a collection of unprocessed scan images with the corresponding locations from which the scan have taken. Since the scan-based map does not require any feature extraction, data association is relatively straight forward.

Past efforts on the SLAM problem can be also classified by the estimation method. The EKF SLAM estimates the robot pose and update the map in the EKF framework. This has proven its capability in certain situations, however, the inherent EKF linearization error can degrades the result. The particle filter SLAM is another popular approach that uses RBPF to estimate the robot pose and to create the map. It might provide an accurate solution to the SLAM problem, but the accuracy significantly drops if the number of particles is not enough. The maximum likelihood SLAM iteratively estimates the robot pose by computing the most likely scan or map. Although a number of approaches have proved real-time SLAM performance, there is still a chance that the accuracy can be an issue when accelerating the speed of convergence.

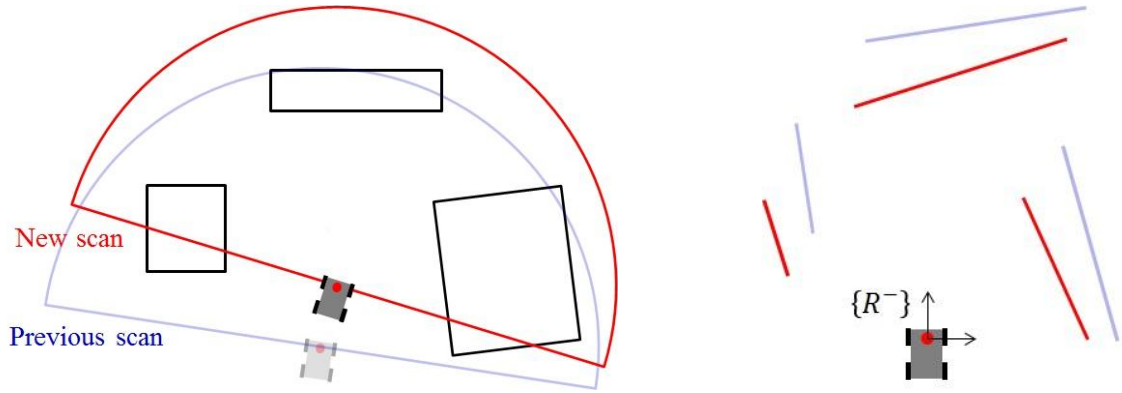
Chapter 3

Scan Matching

3.1 Introduction

Figure 6 illustratively shows the robot with a range sensor observes its surrounding environment at different time steps while it is moving. Having the two different scans, they are registered and matched to each other on the same coordinate system by a technique called the scan matching to find the rigid body transformation between the positions from which the two scans are taken. For the mobile robot to localize itself the scan matching technique is useful, since sensor readings for the scan matching do not require feature extracting processes, which prevents inappropriate data association caused by the feature extraction when matching one scan to another. In addition, the sensors used for the scan matching is more accurate and much more robust than those who directly capture the dynamics of the robot. In recently years, the scan matching plays a very important role in solving the SLAM problem due to its good performance as well as its robustness. As a result, a large number of scan matching techniques have been proposed and utilized for achieving the robotic localization and mapping with the development of additional strategies.

One of the most popular scan matching techniques for the SLAM problem is based on the iterative closest point (ICP) technique [74], which allows the point-to-point matching between two scans by



(a) (b)

Figure 6. Previous scan (blue) and new scan (red)

minimizing the total distance between them. Despite the popularity of the technique, the point-to-point correspondence may yield inappropriate data association since two corresponding points are not actually on the same position in the environment. This point-wise correspondence also makes the technique sensitive to the false detection. In order to avoid the inherent drawbacks of the ICP technique and to enhance its performance, a lot of variants have been proposed. Zhang [75] added a robust outlier rejection method to the ICP technique when selecting the correspondence. The k-d tree, data structure for storing a finite set of points from a k-dimensional space, was additionally implemented to accelerate the search for the point-wise correspondence [76, 77]. Conventional ICP techniques uses the Euclidean distance to compute the distances between scan points, and the least square sums as maximum likelihood estimator. However, they do not provide a good estimation when the robot rotates. To overcome this problem, Iterative Dual Correspondence (IDC) [78] establishes two sets of correspondences, one dealing with the translation using the Euclidean distance and the other with the rotation by means of an angular distance. Metric-Based Iterative Closest Point (MBICP) [79] defines a new distance measure that simultaneously accounted for translation and rotation errors.

Instead of using the point-to-point correspondence, point-to-line based [80, 81] and point-to-plane [82] techniques were introduced where a point corresponds to a line and a plane, respectively. These

approaches reduced the effects of exact correspondence, but they required the feature extraction. Biber and Straßer [42] proposed another scan matching technique by representing a subdivided grid space and collectively describing a scan within each grid cell by a ND. This grid-based technique, the so-called Normal Distribution Transform (NDT), spatially associates every point of the new scan in a grid cell to the ND in the cell. The NDT requires neither the point-to-point correspondence nor feature extraction as points with no feature are collectively handled. However, the scan matching performance of the NDT relies on the size of the grid cell and outliers from the false detection. Moreover, the NDT scan matching might fail when the initial guess that matches two scans is not good. Inspired by the NDT scan matching, Takubo et al. [83] implemented the ICP technique as the initial guess for the NDT scan matching and further proposed a technique to eliminate outliers. Takeuchi and Tsubouchi [84] proposed the extension of the two-dimensional NDT scan matching to the three-dimensional scan matching. In their approach, a scan is divided into voxels and the ND of each cell is approximated by scan points in the cell. In order to match three-dimensional scans using the two-dimensional NDT method, Ripperda and Brenner [85] applied an algorithm to cut a slice which is parallel to the ground out of three-dimensional data. Additionally, for consistent convergence, coarse-to-fine strategy that changes the cell size was implemented.

Other past efforts include scan matching techniques that incorporate appropriate sensor uncertainty models. Pfister et al. [86] presented a method that weights the contribution of each scan point according to its uncertainty, and Montesano et al. [87] introduced probabilistic computation of the correspondences between the scans.

In the next sections, the scan matching is technically reviewed. Section 3.2 defines the previous and new scan and describes the process of the scan matching. Section 3.3 and 3.4 present the fundamental formulations of the ICP and the NDT scan matching techniques since these are the most associated with

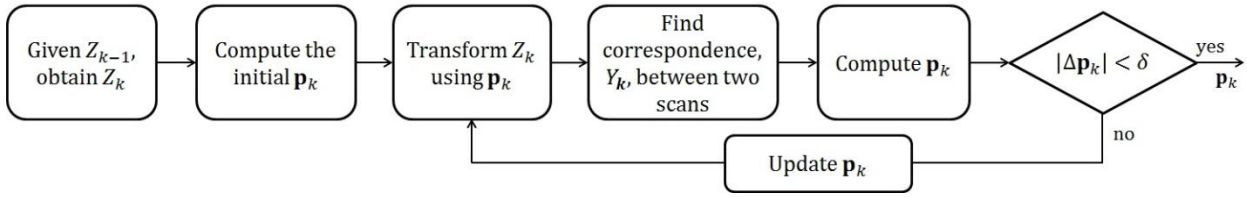


Figure 7. Scan-to-scan matching process

the scan-to-map matching technique which will be detailed in the next chapter. In Section 3.5 an experimental result of the dead reckoning based on the scan matching techniques is presented as an application of the scan matching.

3.2 Scan Matching Techniques

Figure 7 shows the schematic diagram of the general scan-to-scan matching technique. When scans are taken by a range sensor on a moving robot, they are sequentially obtained with respect to different robot coordinate systems. Let ${}^{\{R^-\}}Z_{k-1} = \left\{ {}^{\{R^-\}}\mathbf{z}_{k-1}^i | \forall i \in \{1, \cdots, m\} \right\}$ be the previous scan in the previous robot coordinate system, and ${}^{\{R\}}Z_k = \left\{ {}^{\{R\}}\mathbf{z}_k^i | \forall i \in \{1, \cdots, m\} \right\}$ be the new scan in the new robot coordinate system, where k is the time step, m is the number of points in the scan. $\{R^-\}$ and $\{R\}$ denote the previous robot coordinate system and the new robot coordinate system.

Given the two scans, a scan-to-scan matching technique iteratively finds relative transformation parameters, ${}^{\{R^-\}}_{\{R\}}\mathbf{p}_k = \left[t_k^x, t_k^y, \phi_k\right]^t$, composed of a translation, $\left[t_k^x, t_k^y\right]^t$, and a rotation, ϕ_k, between the two coordinate systems by locally matching the two scans. The first step is to transform the new scan in the new robot coordinate system to that in the previous coordinate system using the currently guessed transformation parameters. Note that the initial transformation parameters can be estimated from readings of other sensors such as an odometer, or can be set as zeros assuming that the two scans are close enough. Mathematically, the transformation

of a point of the new scan in the new robot coordinate system to that in the previous robot coordinate system is performed as

$$ {}^{\{R^-\}}\mathbf{z}_k^i \left({}^{\{R^-\}}_{\{R\}}\mathbf{p}_k \right) = \mathbf{R}(\phi_k) {}^{\{R\}}\mathbf{z}_k^i + \mathbf{t}_k = \begin{bmatrix} \cos\phi_k & -\sin\phi_k \\ \sin\phi_k & \cos\phi_k \end{bmatrix} \begin{bmatrix} {}^{\{R\}}\mathrm{z}_k^{x_i} \\ {}^{\{R\}}\mathrm{z}_k^{y_i} \end{bmatrix} + \begin{bmatrix} t_k^x \\ t_k^y \end{bmatrix} \quad 3.1 $$

where $\mathbf{t}_k = \left[t_k^x,\ t_k^y\right]^t$, and ${}^{\{R\}}\mathbf{z}_k^i = \left[{}^{\{R\}}\mathrm{z}_k^{x_i},\ {}^{\{R\}}\mathrm{z}_k^{y_i}\right]^t$.

Then each point of the new scan ${}^{\{R^-\}}Z_k$ is associated with ${}^{\{R^-\}}Z_{k-1}$ and finds the correspondence set, ${}^{\{R^-\}}Y_k = \left\{ {}^{\{R^-\}}Y_k^i \mid \forall i \in \{1, \cdots, m\} \right\}$, to which the new scan is to be compared. Note that the number of corresponding elements may be less than m if any new scan point does not find a corresponding element. The new transformation parameters are finally computed by minimizing the error metric between the new scan and the correspondence, or equivalently maximizing the score function indicating how good the scan-to-scan matching is. The way of finding ${}^{\{R^-\}}Y_k$ and of computing the transformation parameters varies by scan-to-scan matching techniques, which will be detailed in the following subsections. The iterative identification of the transformation parameters stops when the absolute value of the increment of computation is lower than the specified threshold value:

$$ \left| \Delta_{\{R\}}^{\{R^-\}}\mathbf{p}_k \right| < \delta \quad 3.2 $$

3.3 ICP

When the new scan is transformed to the coordinate system of the previous scan, the ICP scan-to-scan matching technique calculates the distance to all previous scan points from each new scan point and finds the corresponding point, i.e. ${}^{\{R^-\}}Y_k^i = {}^{\{R^-\}}\mathbf{y}_k^i$, that has the minimum distance (Figure 8). The corresponding point has the shortest distance to the new scan point:

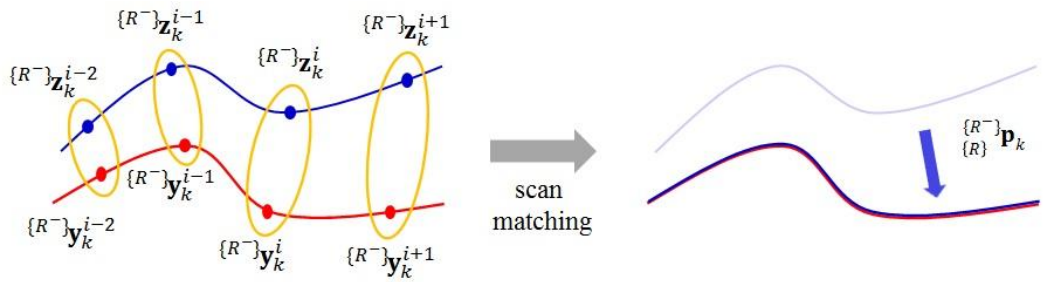


Figure 8. Point-wise correspondence and the matching of two scans

$$d\left({}^{\{R^-\}}\mathbf{z}_k^i, {}^{\{R^-\}}\mathbf{y}_k^i\right) = \min\left\{d\left({}^{\{R^-\}}\mathbf{z}_k^i, {}^{\{R^-\}}\mathbf{z}_{k-1}^j\right) | \forall j \in \{1, \cdots, m\}\right\} \tag{3.3}$$

where $d(\cdot,\cdot)$ denotes a distance between two points. Given the correspondence the derivation of ${}^{\{R^-\}}_{\{R\}}\mathbf{p}_k$ is equivalent to solving the minimization problem of the error metric:

$$e\left({}^{\{R^-\}}_{\{R\}}\mathbf{p}_k\right) = \sum_i \left\| {}^{\{R^-\}}\mathbf{y}_k^i - \left(\mathbf{R}(\phi_k){}^{\{R\}}\mathbf{z}_k^i + \mathbf{t}_k\right)\right\|^2 \rightarrow \min_{{}^{\{R^-\}}_{\{R\}}\mathbf{p}_k} \tag{3.4}$$

The ICP technique solves the minimization problem using the singular value decomposition (SVD) [88]. The means of the new scan and its corresponding point set are first computed as:

$${}^{\{R\}}\bar{\mathbf{z}}_k = \frac{1}{m}\sum_i {}^{\{R\}}\mathbf{z}_k^i, \; {}^{\{R^-\}}\bar{\mathbf{y}}_k = \frac{1}{m}\sum_i {}^{\{R^-\}}\mathbf{y}_k^i \tag{3.5}$$

Defining ${}^{\{R\}}\mathbf{a}_k^i = {}^{\{R\}}\mathbf{z}_k^i - {}^{\{R\}}\bar{\mathbf{z}}_k$ and ${}^{\{R^-\}}\mathbf{b}_k^i = {}^{\{R^-\}}\mathbf{y}_k^i - {}^{\{R^-\}}\bar{\mathbf{y}}_k$, the error metric, $e\left({}^{\{R^-\}}_{\{R\}}\mathbf{p}_k\right)$, in Equation 3.4 can be rewritten as

$$e\left({}^{\{R^-\}}_{\{R\}}\mathbf{p}_k\right) = \sum_{i=1} \left\| {}^{\{R^-\}}\mathbf{b}_k^i - \mathbf{R}(\phi_k){}^{\{R\}}\mathbf{a}_k^i + \left({}^{\{R^-\}}\mathbf{y}_k^i - \mathbf{R}(\phi_k){}^{\{R\}}\bar{\mathbf{z}}_k - \mathbf{t}_k\right)\right\|^2 \tag{3.6}$$

Decoupling the rotation and the translation, the substitution of ${}^{\{R^-\}}\mathbf{y}_k^i - \mathbf{R}(\phi_k){}^{\{R\}}\bar{\mathbf{z}}_k - \mathbf{t}_k = 0$ into Equation 3.6 yields

$$e\left({}^{\{R^-\}}_{\{R\}}\mathbf{p}_k\right) = \sum_i \left\| {}^{\{R^-\}}\mathbf{b}_k^i - \mathbf{R}(\phi_k)\,{}^{\{R\}}\mathbf{a}_k^i \right\|^2$$

$$= \sum_i \left({}^{\{R^-\}}\mathbf{b}_k^i - \mathbf{R}(\phi_k)\,{}^{\{R\}}\mathbf{a}_k^i\right)\left({}^{\{R^-\}}\mathbf{b}_k^i - \mathbf{R}(\phi_k)\,{}^{\{R\}}\mathbf{a}_k^i\right)^t \qquad 3.7$$

$$= \sum_{i=1}^{n_c}\left\|\mathbf{b}_k^i\right\|^2 + \sum_{i=1}^{n_c}\left\|\mathbf{a}_k^i\right\|^2 - \mathrm{tr}\big(\mathbf{R}(\phi_k)\big)\mathbf{N}_k$$

where $\mathbf{N}_k = \sum_i {}^{\{R\}}\mathbf{a}_k^i\,{}^{\{R^-\}}\mathbf{b}_k^i$. In the above equation, the error metric is minimized when $\mathrm{tr}(\mathbf{R}(\phi_k)\mathbf{N}_k)$ is maximized. Decomposing $\mathbf{N}_k$ by the SVD into $\mathbf{N}_k = \mathbf{U}_k\mathbf{D}_k\mathbf{V}_k^t$, the transformation matrix, $\mathbf{R}(\phi_k)$, and $\mathbf{t}_k$ are finally given by

$$\mathbf{R}(\phi_k) = \mathbf{V}_k\mathbf{U}_k^t, \quad \mathbf{t}_k = {}^{\{R^-\}}\mathbf{y}_k^i - \mathbf{R}(\phi_k)\,{}^{\{R\}}\bar{\mathbf{z}}_k \qquad 3.8$$

where $\mathbf{U}_k$ and $\mathbf{V}_k$ are real or complex unitary matrices, and $\mathbf{D}_k$ is a rectangular diagonal matrix with nonnegative real number entries [88]. From $\mathbf{R}(\phi_k)$ the orientational transformation parameter, ϕ_k, can be derived as

$$\phi_k = \mathrm{atan2}(R_{21}, R_{11}) \qquad 3.9$$

where R_{ij} is the entry of $\mathbf{R}$ in the *i*th row and the *j*th column.

3.4 Normal Distribution Transform

Unlike the ICP technique the NDT scan-to-scan matching technique compares each new scan point to a ND since the NDT technique maps ${}^{\{R^-\}}\mathbf{z}_k^i$ onto a grid space having cells each represented with a ND. The NDT technique first defines a grid space with respect to the previous robot coordinate system and derives a ND for each grid cell after identifying ${}^{\{R^-\}}Z_{k-1}$ on the space as shown in Figure 9. For the *j*th cell, the mean and covariance matrix are computed by

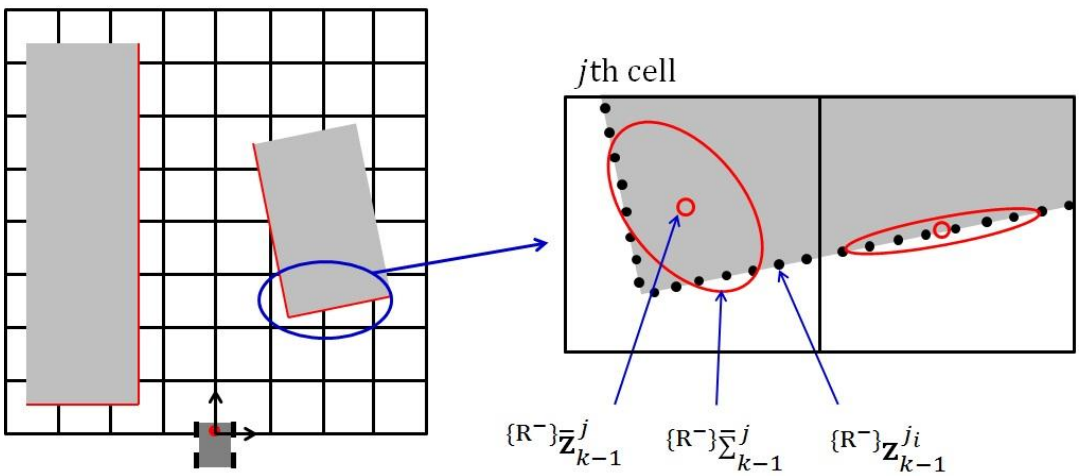


Figure 9. Two-dimentional grid space and normal distributions created by scan points

$$
\begin{aligned}
{}^{\{R^-\}}\bar{\mathbf{z}}^{j}_{k-1} &= \frac{1}{m^{j}_{k-1}} \sum_{i=1}^{m^{j}_{k-1}} {}^{\{R^-\}}\mathbf{z}^{j_i}_{k-1} \\
{}^{\{R^-\}}\bar{\Sigma}^{j}_{k-1} &= \frac{1}{m^{j}_{k-1}} \sum_{i=1}^{m^{j}_{k-1}} \left({}^{\{R^-\}}\mathbf{z}^{j_i}_{k-1} - {}^{\{R^-\}}\bar{\mathbf{z}}^{j}_{k-1}\right)\left({}^{\{R^-\}}\mathbf{z}^{j_i}_{k-1} - {}^{\{R^-\}}\bar{\mathbf{z}}^{j}_{k-1}\right)^{t}
\end{aligned}
\qquad 3.10
$$

where ${}^{\{R^-\}}\mathbf{z}^{j_i}_{k-1}$ is the ith point of the previous scan in the jth cell, and m^{j}_{k-1} is the number of scan points in the cell.

After transforming every new scan point using the currently guessed transformation parameters, each point is located in some grid cell. If ${}^{\{R^-\}}\mathbf{z}^{j_i}_{k}$ sees a ND created by the previous scan in the cell, the correspondence or the properties of the ND, i.e. ${}^{\{R^-\}}Y^{i}_{k} = \left\{{}^{\{R^-\}}\hat{\mathbf{z}}^{j_i}_{k}, {}^{\{R^-\}}\hat{\Sigma}^{j_i}_{k}\right\}$, are those of the ND of the previous scan:

$$
{}^{\{R^-\}}\hat{\mathbf{z}}^{j_i}_{k} \leftarrow {}^{\{R^-\}}\bar{\mathbf{z}}^{j}_{k-1}, \quad {}^{\{R^-\}}\hat{\Sigma}^{i}_{k} \leftarrow {}^{\{R^-\}}\bar{\Sigma}^{j}_{k-1} \qquad 3.11
$$

Upon completion of the identification of the correspondence, the derivation of ${}^{\{R^-\}}_{\{R\}}\mathbf{p}_k$ is equivalent to solving the maximization problem of the score function, $s\left({}^{\{R^-\}}_{\{R\}}\mathbf{p}_k\right)$:

$$s\left({}^{\{R^-\}}_{\{R\}}\mathbf{p}_k\right) = \sum_j \sum_{i=1}^{m_k^j} \exp\left(\frac{-\left({}^{\{R^-\}}\mathbf{z}_k^{j_i} - {}^{\{R^-\}}\hat{\mathbf{z}}_k^{j_i}\right)^T \left({}^{\{R^-\}}\hat{\Sigma}_k^i\right)^{-1} \left({}^{\{R^-\}}\mathbf{z}_k^{j_i} - {}^{\{R^-\}}\hat{\mathbf{z}}_k^{j_i}\right)}{2}\right) \rightarrow \max_{{}^{\{R^-\}}_{\{R\}}\mathbf{p}_k} \qquad 3.12$$

Since the score function is the sum of piecewise smooth functions, a standard quadratic optimization method can be used. Applying Newton's method, ${}^{\{R^-\}}_{\{R\}}\mathbf{p}_k$ is iteratively computed by the increment $\Delta^{\{R^-\}}_{\{R\}}\mathbf{p}_k$:

$$\Delta^{\{R^-\}}_{\{R\}}\mathbf{p}_k = -\mathbf{H}_k^{-1}\mathbf{g}_k \qquad 3.13$$

where $\mathbf{H}_k$ and $\mathbf{g}_k$ are the sums of the Hessian, $\widetilde{\mathbf{H}}_k$, and the gradient, $\tilde{\mathbf{g}}_k$, of the objective function $f = -s\left({}^{\{R^-\}}_{\{R\}}\mathbf{p}_k\right)$:

$$\mathbf{H}_k = \sum_j \sum_{i=1}^{m_k^j} \widetilde{\mathbf{H}}_k^{j_i}, \quad \mathbf{g}_k = \sum_j \sum_{i=1}^{m_k^j} \tilde{\mathbf{g}}_k^{j_i} \qquad 3.14$$

Note that $\mathbf{H}_k$ has to be positive definite for the minimization problem to be solvable. If not, $\mathbf{H}_k$ is adjusted by adding $\lambda\mathbf{I}$ which makes it positive definite. For the *i*th new scan point ${}^{\{R^-\}}\mathbf{z}_k^{j_i}$ in the *j*th cell, the gradient vector $\tilde{\mathbf{g}}_k^{j_i}$ is given by

$$\tilde{\mathbf{g}}_k^{j_i} = \tilde{\mathbf{z}}_k^t \left({}^{\{R^-\}}\hat{\Sigma}_k^i\right)^{-1} \frac{\partial \tilde{\mathbf{z}}_k}{\partial\left(t_k^x, t_k^y, \phi_k\right)} \exp\frac{-\tilde{\mathbf{z}}_k^t \left({}^{\{R^-\}}\hat{\Sigma}_k^i\right)^{-1} \tilde{\mathbf{z}}_k}{2} \qquad 3.15$$

and the *m*,*n* entry of $\widetilde{\mathbf{H}}_k^{j_i}$ is computed by:

$$\tilde{H}_k^{j_i}[m,n] = -\exp\frac{-\tilde{\mathbf{z}}_k^t\left({}^{\{R^-\}}\hat{\Sigma}_k^i\right)^{-1}\tilde{\mathbf{z}}_k}{2}\left\{\left(-\tilde{\mathbf{z}}_k^t\left({}^{\{R^-\}}\hat{\Sigma}_k^i\right)^{-1}\frac{\partial\tilde{\mathbf{z}}_k}{\partial p_k[m]}\right)\left(-\tilde{\mathbf{z}}_k^t\left({}^{\{R^-\}}\hat{\Sigma}_k^i\right)^{-1}\frac{\partial\tilde{\mathbf{z}}_k}{\partial p_k[n]}\right)\right.$$
$$\left.+\left(-\tilde{\mathbf{z}}_k^t\left({}^{\{R^-\}}\hat{\Sigma}_k^i\right)^{-1}\frac{\partial^2\tilde{\mathbf{z}}_k}{\partial p_k[m]\partial p_k[n]}\right)+\left(\frac{\partial\tilde{\mathbf{z}}_k^t}{\partial p_k[n]}\left({}^{\{R^-\}}\hat{\Sigma}_k^i\right)^{-1}\frac{\partial\tilde{\mathbf{z}}_k}{\partial p_k[m]}\right)\right\} \quad 3.16$$

where $\tilde{\mathbf{z}}_k = {}^{\{R^-\}}\mathbf{z}_k^{j_i} - {}^{\{R^-\}}\hat{\mathbf{z}}_k^{j_i}$ and the first and second partial derivative of $\tilde{\mathbf{z}}_k$ can be derived as below:

$$\frac{\partial\tilde{\mathbf{z}}_k}{\partial\left(t_k^x, t_k^y, \phi_k\right)} = \left[\frac{\partial\tilde{\mathbf{z}}_k}{\partial p_k[1]}, \frac{\partial\tilde{\mathbf{z}}_k}{\partial p_k[2]}, \frac{\partial\tilde{\mathbf{z}}_k}{\partial p_k[3]}\right] = \begin{bmatrix} 1 & 0 & -{}^{\{R\}}z_k^{x_i}\sin\phi_k - {}^{\{R\}}z_k^{y_i}\cos\phi_k \\ 0 & 1 & {}^{\{R\}}z_k^{x_i}\cos\phi_k - {}^{\{R\}}z_k^{y_i}\sin\phi_k \end{bmatrix} \quad 3.17$$

$$\frac{\partial^2\tilde{\mathbf{z}}_k}{\partial p_k[m]\partial p_k[n]} = \begin{cases} \begin{bmatrix} -{}^{\{R\}}z_k^{x_i}\cos\phi_k + {}^{\{R\}}z_k^{y_i}\sin\phi_k \\ -{}^{\{R\}}z_k^{x_i}\sin\phi_k - {}^{\{R\}}z_k^{y_i}\cos\phi_k \end{bmatrix} & \text{for } m = n = 3 \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{otherwise} \end{cases} \quad 3.18$$

when $\Delta_{\{R\}}^{\{R^-\}}\mathbf{p}_k$ is computed, ${}_{\{R\}}^{\{R^-\}}\mathbf{p}_k$ is then updated by:

$${}_{\{R\}}^{\{R^-\}}\mathbf{p}_k \leftarrow {}_{\{R\}}^{\{R^-\}}\mathbf{p}_k + \Delta_{\{R\}}^{\{R^-\}}\mathbf{p}_k \quad 3.19$$

3.5 Dead Reckoning using Scan Matching Techniques

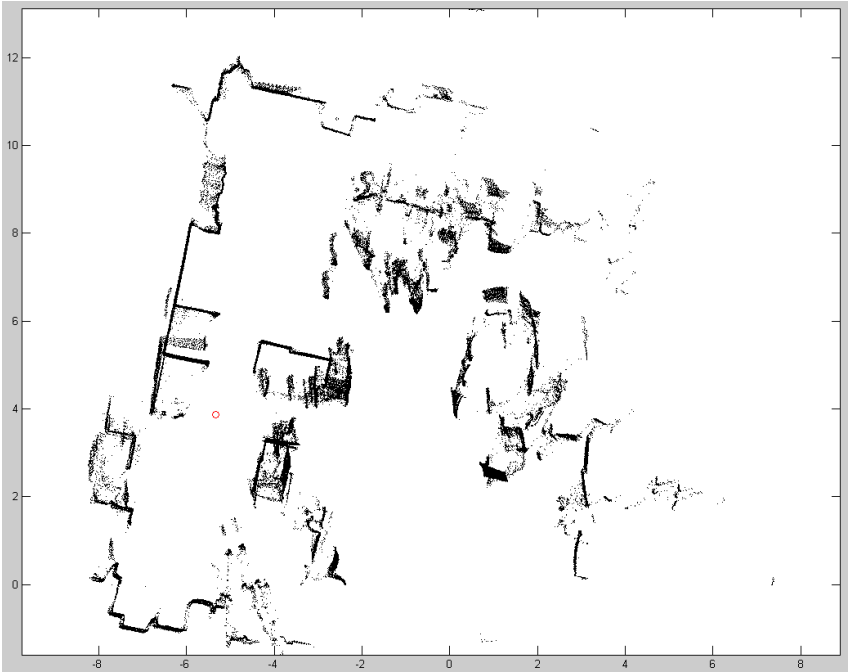
Figure 10 shows the dead reckoning based on the ICP and the NDT scan matching technique. In this experiment the robot took 362 scans all together in a real indoor environment. The new scan was matched only to the previous scan at every acquisition of the new scan, and the ICP and the NDT techniques derived the transformation parameters in real time. As shown in the figure the dead reckoning by the NDT performed better than that by the ICP technique in this environment. This is because the scan points on the same object in the environment after the ICP scan matching do not lie on the same position in the map, which indicates that the estimation of the transformation parameters are not accurate. Since the scan points are transformed after the scan matching process, the estimation of the robot pose is also considered

to be wrong. On the other hand, the quality of the map created by the NDT dead reckoning is reasonable good in the qualitative manner, and it can be said that the pose of the robot is thus relatively well estimated.

Figure 11 shows trajectories of the robot when the robot pose is estimated by the ICP and the NDT scan matching technique, where the initial position for each technique is equivalent to $[0, 0]^t$. The trajectories of the robot by the two techniques are almost on the same locations for a while after the initial time step. However, they become different around the location of $[-0.7, 2.5]^t$, since the scan matching techniques start deriving different transformation parameters. Considering that the quality of the map created by the ICP dead reckoning is not good, it is expected that the error between the true and the estimated locations of the robot is greater when the robot pose is estimated by the ICP scan matching than the NDT scan matching.

Figure 12 shows the position and the orientation differences at each time step between robot poses by the ICP and the NDT dead reckoning. As can be seen in the previous figure showing the trajectories of the robot, the position and the orientation differences for the first 46 time steps are very small. Since then, the position and the orientation differences become large, which leads to difference trajectories by the ICP and the NDT dead reckoning.

Although the NDT scan matching technique has shown a better result in this experiment, it is not necessarily true that the NDT always works better than the ICP or other scan matching techniques. The scan matching performance can vary by a lot of conditions such as the environmental conditions and the default parameters for the scan matching technique. In other words, the ICP scan matching technique can derive the transformation parameters more accurately than the NDT technique in some other occasions. A scan matching technique always brings the scan matching error and this is the reason that the scan matching technique cannot be a solution to the SLAM problem by itself and that it should be used as a technique for solving the SLAM problem.



(a) Sequential ICP scan matching

(b) Sequential NDT scan matching

Figure 10. Dead reckoning after 362 scans

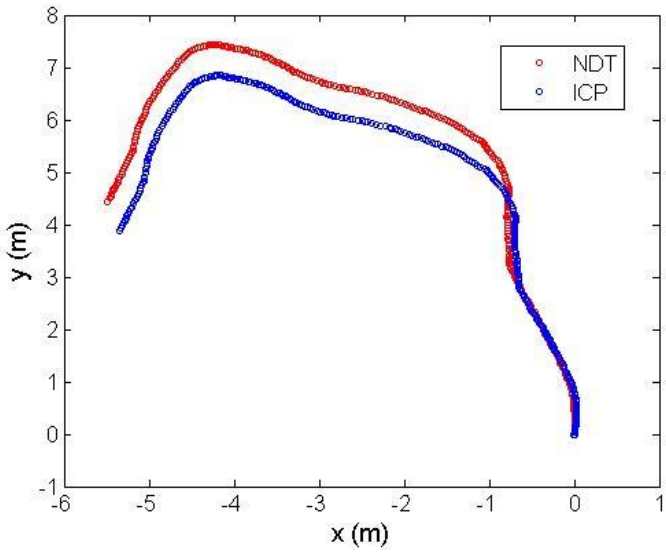


Figure 11. Trajectories of the robot by the ICP and the NDT dead reckoning

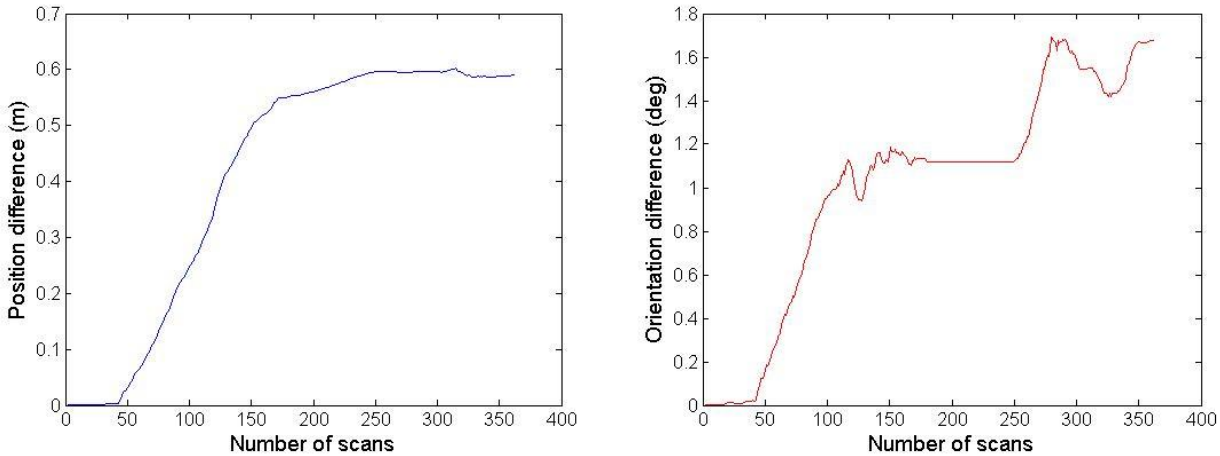


Figure 12. Position difference (left) and orientation difference (right) between the robot poses estimated by the ICP and the NDT dead reckoning

3.6 Summary

This chapter has briefly reviewed existing scan matching techniques and presented mathematical formulations of two specific techniques, the ICP and the NDT. The ICP technique finds corresponding

points for every new scan point and estimates the transformation parameters by minimizing the error metric between the new scan points and their correspondence. This technique is theoretically simple and easy to implement. However, the error metric to be minimized requires the point-to-point correspondence and the performance of the ICP is thus sensitive to the point-to-point correspondence. Unlike the ICP technique the NDT scan matching technique does not need the point-to-point correspondence. It represents the previous scan, to which the new scan is to be matched, by a collection of ND on a grid space. The NDT technique then associates multiple new scan points to one ND to compute the transformation parameters. The scan matching performance can be increased by avoiding the point-to-point correspondence. However, the capability of the NDT technique can be affected by the grid size and the initial guess for the transformation parameters.

The experimental result shows that the dead reckoning using the NDT has worked better than that using the ICP scan matching in the specific environment. However, any scan matching technique cannot show its superiority over other scan matching techniques in all situations. Moreover, every scan matching technique generates the scan matching error, which means additional efforts that associate the new scan to the past scans are required to solve the SLAM problem.

Chapter 4

Scan-to-Map Matching

This chapter proposes a grid-based scan-to-map matching technique for SLAM. At every acquisition of a new scan, the proposed technique matches the new scan to the previous scan similarly to the conventional techniques, but further corrects the error by matching the new scan to the globally defined map. In order to achieve best scan-to-map matching at each acquisition, the map is represented as a grid map with multiple NDs in each cell. Additionally, the new scan is also represented by NDs, developing a novel ND-to-ND matching technique. The ND-to-ND matching technique has significant potential in the enhancement of the global matching as well as the computational efficiency. Section 4.1 briefly describes the overall process of the scan-to-map matching technique. Section 4.2 presents the grid map representation and selection of properties of the grid map to match a new scan to the map. The derivation of the transformation parameters and the update of the grid map are detailed in Section 4.3 and 4.4, respectively. Section 4.5 investigates the performance of the scan-to-map matching throughout a number of experimental results whereas Section 4.6 summarizes this chapter.

4.1 Overview

Figure 13 shows the overall process of the proposed grid-based scan-to-map matching technique which is based on the NDT's grid-based matching. Instead of the previous scan, the proposed technique matches the new scan to the globally defined map which is an accumulation of new scans after the scan-to-map matching. When the new scan ${}^{\{R\}}Z_k$ is obtained, the proposed technique first performs the ICP scan-to-scan matching to derive the transformation parameters, ${}^{\{R^-\}}_{\{R\}}\mathbf{p}_k^{\mathrm{ICP}}$, and transforms each new scan point in the $\{R\}$ coordinate system to that in the $\{R^-\}$ coordinate system:

$$ {}^{\{R^-\}}\mathbf{z}_k^i\left({}^{\{R^-\}}_{\{R\}}\mathbf{p}_k^{\mathrm{ICP}}\right) = \mathbf{R}\left(\phi_k^{\mathrm{ICP}}\right){}^{\{R\}}\mathbf{z}_k^i + \mathbf{t}_k^{\mathrm{ICP}} \qquad 4.1 $$

where ${}^{\{R^-\}}_{\{R\}}\mathbf{p}_k^{\mathrm{ICP}} = \left[\mathbf{t}_k^{\mathrm{ICP}t},\ \phi_k^{\mathrm{ICP}}\right]^t$. Having the new scan matched to the previous scan, each new scan point ${}^{\{R^-\}}\mathbf{z}_k^i$ is further transformed to that in the global coordinate system, $\{G\}$, using the robot pose estimated at the previous time step in the $\{G\}$ coordinate system:

$$ {}^{\{G\}}\mathbf{z}_k^i = \mathbf{R}\left({}^{\{G\}}\theta_{k-1}\right){}^{\{R^-\}}\mathbf{z}_k^i + {}^{\{G\}}\mathbf{x}_{k-1} \qquad 4.2 $$

where ${}^{\{G\}}\mathbf{x}_{k-1} = \left[{}^{\{G\}}x_{k-1},\ {}^{\{G\}}y_{k-1}\right]^t$ and ${}^{\{G\}}\theta_{k-1}$ are the robot pose in the global coordinate system estimated at time step $k-1$. The iterative estimation of the robot pose in the global coordinate system is performed by considering the robot movement, ${}^{\{R^-\}}\mathbf{x}_k$ and ${}^{\{R^-\}}\theta_k$, which is equivalent to $\mathbf{t}_k^{\mathrm{ICP}}$ and ϕ_k^{ICP}, respectively:

$$ \begin{gathered} {}^{\{G\}}\mathbf{x}_k = \mathbf{R}\left({}^{\{G\}}\theta_{k-1}\right){}^{\{R^-\}}\mathbf{x}_k + {}^{\{G\}}\mathbf{x}_{k-1} = \mathbf{R}\left({}^{\{G\}}\theta_{k-1}\right)\mathbf{t}_k^{\mathrm{ICP}} + {}^{\{G\}}\mathbf{x}_{k-1} \\ {}^{\{G\}}\theta_k = {}^{\{R^-\}}\theta_k + {}^{\{G\}}\theta_{k-1} = \phi_k^{\mathrm{ICP}} + {}^{\{G\}}\theta_{k-1} \end{gathered} \qquad 4.3 $$

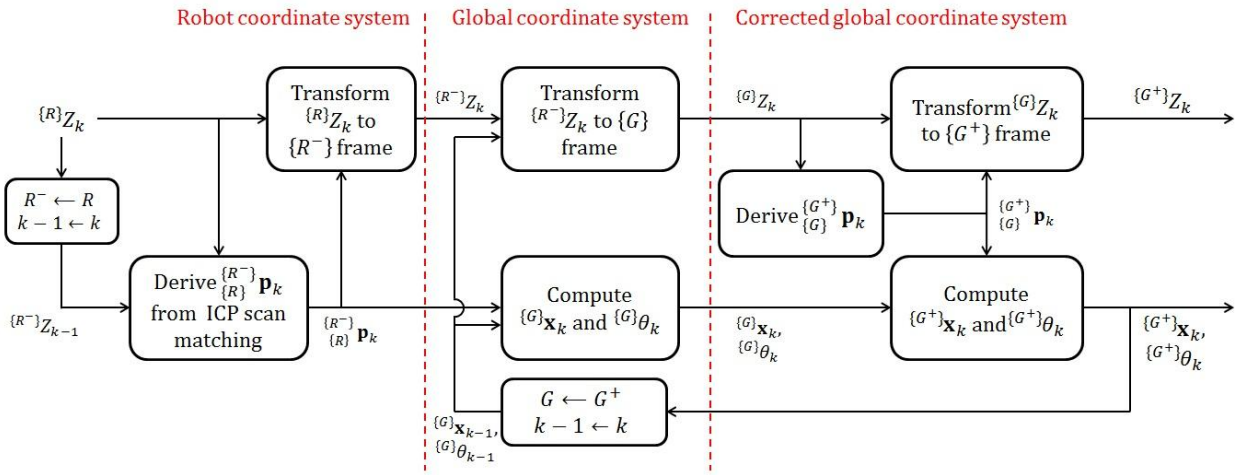


Figure 13. Proposed grid-based scan-to-map matching technique

This global coordinate system is, however, incorrectly located due to the misalignment of the previous robot coordinate system by the ICP scan-to-scan matching as well as the error of estimation of the robot pose. Once the new scan is transformed to the $\{G\}$ coordinate system, the proposed technique iteratively matches the new scan to the map in the $\{G^+\}$ coordinate system, which is the global coordinate system corrected by the proposed technique from the original guess of the global coordinate system, and derives the new scan in the $\{G^+\}$ coordinate system:

$$ {}^{\{G^+\}}\mathbf{z}_k^i \left({}^{\{G^+\}}_{\{G\}}\mathbf{p}_k \right) = \mathbf{R}(\phi_k) {}^{\{G\}}\mathbf{z}_k^i + \mathbf{t}_k \qquad 4.4 $$

where ${}^{\{G^+\}}_{\{G\}}\mathbf{p}_k = [\mathbf{t}_k{}^t,\ \phi_k]^t$ is the error correction parameters, or the scan-to-map matching transformation parameters, and transforms the new scan to the corrected global coordinate system. The derivation of the error correction parameters is detailed in the next subsections. Simultaneously, the robot pose in the $\{G\}$ coordinate system is also corrected by ${}^{\{G^+\}}_{\{G\}}\mathbf{p}_k$:

$$
\begin{aligned}
{}^{\{G^+\}}\mathbf{x}_k &= \mathbf{R}(\phi_k)\,{}^{\{G\}}\mathbf{x}_k + \mathbf{t}_k \\
{}^{\{G^+\}}\theta_k &= {}^{\{G\}}\theta_k + \phi_k
\end{aligned}
\qquad 4.5
$$

Because the misalignment of the previous robot coordinate system and the error of the robot pose are corrected by matching the new scan to the map, the proposed technique does not accumulate the scan-to-scan matching error as well as the pose estimation error.

Having the overall process of the scan-to-map matching identified, the representation of the grid map having multiple NDs in each cell is first defined in Section 4.2. In addition to the map NDs, the scan NDs are then derived from the new scan and paired with map NDs for scan-to-map matching. Section 4.3 presents the derivation of ${}^{\{G^+\}}_{\{G\}}\mathbf{p}_k$ via the ND-to-ND matching, whereas the update of the grid map using the derived ${}^{\{G^+\}}_{\{G\}}\mathbf{p}_k$ is detailed in Section 4.4. In order to simplify the notation the corrected global coordinate system, $\{G^+\}$, will be dropped from now on, and all notations in this chapter without the coordinate system are considered as being in the corrected global coordinate system.

4.2 Grid Map Representation and Selection of Matching Map Normal Distribution

Figure 14 illustratively shows the grid map with multiple map NDs in each cell together with the matching of new scan to the map NDs. As shown in the figure, the new scan of an object can be significantly different depending on where the scan is taken. The grid map with multiple NDs allows the matching of the new scan to a map ND irrespective of the robot pose. Mathematically, such a grid map updated up to time step $k-1$ for deriving ${}^{\{G^+\}}_{\{G\}}\mathbf{p}_k$ is represented as

$$
M_{1:k-1} = \left\{ M^j_{1:k-1} \mid \forall j \in \{1, \cdots, n^g\} \right\}
$$

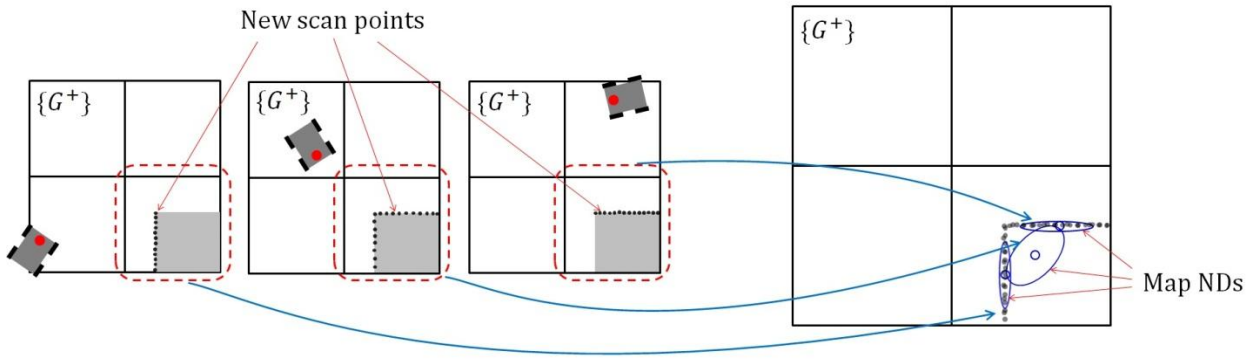


Figure 14. The grid map represented by multiple NDs (right) and new scans to be matched to the grid map (left)

where $M_{1:k-1}^{j}$ is the property of the jth grid cell, and n^g is the number of grid cells. $M_{1:k-1}^{j}$ is given by

$$M_{1:k-1}^{j} = \left\{ M_{1:k-1}^{j_l} = \left\{ \bar{\mathbf{z}}_{1:k-1}^{j_l}, \bar{\Sigma}_{1:k-1}^{j_l}, m_{1:k-1}^{j_l} \right\} | \forall l \in \left\{ 1, \cdots, n_{k-1}^{j} \right\} \right\}$$

where $M_{1:k-1}^{j}$ is the property of the lth map ND in the jth cell with the mean, $\bar{\mathbf{z}}_{1:k-1}^{j_l}$, covariance matrix, $\bar{\Sigma}_{1:k-1}^{j_l}$, and the total number of scan points, $m_{1:k-1}^{j_l}$. n_{k-1}^{j} denotes the total number of map NDs in the jth cell.

With the new scan transformed to the $\{G\}$ coordinate system, the scan ND in the jth cell to match to a map ND in the same cell is derived simply as

$$\begin{aligned} {}^{\{G\}}\bar{\mathbf{z}}_k^j &= \frac{1}{m_k^j} \sum_{i=1}^{m_k^j} {}^{\{G\}}\mathbf{z}_k^{j_i} \\ {}^{\{G\}}\bar{\Sigma}_k^j &= \frac{1}{m_k^j} \sum_{i=1}^{m_k^j} \left({}^{\{G\}}\mathbf{z}_k^{j_i} - {}^{\{G\}}\bar{\mathbf{z}}_k^j \right) \left({}^{\{G\}}\mathbf{z}_k^{j_i} - {}^{\{G\}}\bar{\mathbf{z}}_k^j \right)^t \end{aligned} \qquad 4.6$$

where ${}^{\{G\}}\mathbf{z}_k^{j_i}$ is the ith scan point in the jth cell and m_k^j is the total number of points in the jth cell. The selection of a matching map ND for the scan ND in the proposed technique starts with quantifying the similarity of the scan ND to each map ND in the same cell. The similarity can be computed by the KL

divergence, D_{KL}, which is a mathematically solid method for measuring the distance between two probability distributions:

$$S\left(N\left({}^{\{G\}}\bar{\mathbf{z}}_k^j, {}^{\{G\}}\bar{\Sigma}_k^j\right), N\left(\bar{\mathbf{z}}_{1:k-1}^{j_l}, \bar{\Sigma}_{1:k-1}^{j_l}\right)\right) = -D_{KL}\left(N\left({}^{\{G\}}\bar{\mathbf{z}}_k^j, {}^{\{G\}}\bar{\Sigma}_k^j\right) || N\left(\bar{\mathbf{z}}_{1:k-1}^{j_l}, \bar{\Sigma}_{1:k-1}^{j_l}\right)\right)$$

$$= -\frac{1}{2}\left(\mathrm{tr}\left(\left(\bar{\Sigma}_{1:k-1}^{j_l}\right)^{-1} {}^{\{G\}}\bar{\Sigma}_k^j\right) + \left(\bar{\mathbf{z}}_{1:k-1}^{j_l} - {}^{\{G\}}\bar{\mathbf{z}}_k^j\right)^t \left(\bar{\Sigma}_{1:k-1}^{j_l}\right)^{-1}\left(\bar{\mathbf{z}}_{1:k-1}^{j_l} - {}^{\{G\}}\bar{\mathbf{z}}_k^j\right) - \ln\left(\frac{\det\left({}^{\{G\}}\bar{\Sigma}_k^j\right)}{\det\left(\bar{\Sigma}_{1:k-1}^{j_l}\right)}\right) - \lambda\right) \quad 4.7$$

where $l \in \{1, \cdots, n_{k-1}^j\}$, λ is the dimension of the NDs, and $\left({}^{\{G\}}\bar{\mathbf{z}}_k^j, {}^{\{G\}}\bar{\Sigma}_k^j\right)$ and $N\left(\bar{\mathbf{z}}_{1:k-1}^{j_l}, \bar{\Sigma}_{1:k-1}^{j_l}\right)$ are the scan ND and the lth map ND, respectively. Out of the map NDs the most similar one to the scan ND is that with the highest similarity value:

$$S\left(N\left({}^{\{G\}}\bar{\mathbf{z}}_k^j, {}^{\{G\}}\bar{\Sigma}_k^j\right), N\left(\bar{\mathbf{z}}_{1:k-1}^{j_{l^*}}, \bar{\Sigma}_{1:k-1}^{j_{l^*}}\right)\right) = \min\left\{S\left(N\left({}^{\{G\}}\bar{\mathbf{z}}_k^j, {}^{\{G\}}\bar{\Sigma}_k^j\right), N\left(\bar{\mathbf{z}}_{1:k-1}^{j_l}, \bar{\Sigma}_{1:k-1}^{j_l}\right)\right) | l \in \{1, \cdots, n_{k-1}^j\}\right\} \quad 4.8$$

The l^*th map ND is regarded as the matching map ND for the scan ND if the similarity is greater than the specified threshold value:

$$S\left(N\left({}^{\{G\}}\bar{\mathbf{z}}_k^j, {}^{\{G\}}\bar{\Sigma}_k^j\right), N\left(\bar{\mathbf{z}}_{1:k-1}^{j_{l^*}}, \bar{\Sigma}_{1:k-1}^{j_{l^*}}\right)\right) > \gamma \quad 4.9$$

Having the matching map ND identified for each scan ND, the derivation of ${}^{\{G^+\}}_{\{G\}}\mathbf{p}_k$ is possible by matching all the scan NDs to the corresponding matching map NDs.

4.3 Derivation of Error Correction Parameters

Since a scan ND in each cell is compared to a matching map ND, the correspondence is derived not for every point but for every cell, i.e.$Y_k^j = \{\hat{\mathbf{z}}_k^j, \hat{\Sigma}_k^j\}$. It is equivalent to the property of the matching map ND:

$$\begin{aligned} \hat{\mathbf{z}}_k^j &\leftarrow \bar{\mathbf{z}}_{1:k-1}^{j_{l^*}} \\ \hat{\Sigma}_k^j &\leftarrow \bar{\Sigma}_{1:k-1}^{j_{l^*}} \end{aligned} \qquad \text{if } S\left(N\left({}^{\{G\}}\bar{\mathbf{z}}_k^j, {}^{\{G\}}\bar{\Sigma}_k^j\right), N\left(\bar{\mathbf{z}}_{1:k-1}^{j_{l^*}}, \bar{\Sigma}_{1:k-1}^{j_{l^*}}\right)\right) > \gamma \qquad 4.10$$

Note that a scan ND that does not have a matching map ND is not thus considered in the derivation of ${}^{\{G^+\}}_{\{G\}}\mathbf{p}_k$. Given the correspondence of the scan NDs, the derivation of ${}^{\{G^+\}}_{\{G\}}\mathbf{p}_k$ begins with the initial values set to **0** as it is valid to assume that the ICP scan-to-scan matching and the previous robot pose estimation is reasonably correct. The proposed technique first transforms the mean and covariance matrix of each scan ND to those in the $\{G^+\}$ coordinate system using the currently guessed ${}^{\{G^+\}}_{\{G\}}\mathbf{p}_k$:

$$\begin{aligned} \bar{\mathbf{z}}_k^j &= \mathbf{R}(\phi_k)\,{}^{\{G\}}\bar{\mathbf{z}}_k^j + \mathbf{t}_k \\ \bar{\Sigma}_k^j &= \mathbf{R}(\phi_k)\,{}^{\{G\}}\bar{\Sigma}_k^j\,\mathbf{R}(\phi_k)^{-1} \end{aligned} \qquad 4.11$$

With all the scan NDs and the matching map NDs described in the $\{G^+\}$ coordinate system, the transformation parameters ${}^{\{G^+\}}_{\{G\}}\mathbf{p}_k$ can be then computed by maximizing the objective function given by the sum of similarities between the scan NDs and the matching map NDs:

$$f\left({}^{\{G^+\}}_{\{G\}}\mathbf{p}_k\right) = \sum_j S\left(N\left(\bar{\mathbf{z}}_k^j, \bar{\Sigma}_k^j\right), N\left(\hat{\mathbf{z}}_k^j, \hat{\Sigma}_k^j\right)\right) \qquad 4.12$$

The objective function of the proposed technique equally sums the similarities. In other words, similarities with a small number of scan points can be treated as equally as those with a large number of scan points. This could allow the proposed technique to match the new scan to the map more globally

than the conventional point-to-X techniques. The ND-to-ND matching could also dramatically improve the computation time.

Although the analytical expressions of the gradient and the Hessian may be obtained for the objective function, the small-size optimization problem, with only three parameters for the two-dimensional scan, could be easily solved with the Newton method numerically computing the gradient and the Hessian.

4.4 The Update of the Grid Map

The grid map is initially that with the first scan NDs, and this is regarded as the first grid map updated up to the previous time step. Given the mean and the covariance matrix of the scan ND of each cell in the $\{G\}$ coordinate system shown in Equation 4.6, the proposed technique then updates map NDs in the same cell differently depending on whether there is a matching map ND. If there is a matching map ND, only this matching map ND is updated with the scan ND. The mean and covariance matrix of the matching map ND in the jth cell are updated according to the weighted mean formulation:

$$\begin{aligned} \bar{\mathbf{z}}_{1:k}^{j_{l^*}} &= \frac{m_{1:k-1}^{j}\bar{\mathbf{z}}_{1:k-1}^{j_{l^*}} + m_k^j \bar{\mathbf{z}}_k^j}{m_{1:k-1}^{j} + m_k^j} \\ \bar{\Sigma}_{1:k}^{j_{l^*}} &= \frac{m_{1:k-1}^{j}\bar{\Sigma}_{1:k-1}^{j_{l^*}} + m_k^j \bar{\Sigma}_k^j}{m_{1:k-1}^{j} + m_k^j} \end{aligned} \qquad 4.13$$

After the update, the number of scan points for the map ND is also updated:

$$m_{1:k}^{j_{l^*}} = m_{1:k-1}^{j_{l^*}} + m_k^j \qquad 4.14$$

On the other hand, if the scan ND has found no matching map ND, the scan ND is simply added as a new map ND without any update to the current map NDs. Let the index of the new map ND be $l^+ = n_k^j + 1$. The mean and the covariance matrix of the map ND in the jth cell are given by

$$\bar{\mathbf{z}}_{1:k}^{j_{l^+}} = \bar{\mathbf{z}}_k^j$$
$$\bar{\Sigma}_{1:k}^{j_{l^+}} = \bar{\Sigma}_k^j \qquad 4.15$$

The number of scan points of the map ND is, similarly, the number of scan points of the scan ND:

$$m_{1:k}^{j_{l^+}} = m_k^j \qquad 4.16$$

After the scan ND is added, the number of the map NDs becomes $n_k^j \leftarrow n_k^j + 1$. The update of the grid map completes by applying the cell-wise update to all the grid cells.

4.5 Experimental Results

This section is aimed at investigating the performance of the proposed scan-to-map matching technique and demonstrating the applicability of the proposed technique in real indoor environments. All experiments were conducted using a ground mobile robot with a forward-facing LRF, Hokuyo UTM-30lx, mounted on the robot (Figure 15). No other sensors such as an odometer and an IMU were used to estimate the pose of the robot and to build a map. In the first experiment, the performance of the proposed technique is investigated based on the position and orientation error seen from landmarks at every matching of the new scan to the map. Second experiment focuses on showing the effectiveness of multi-ND representation within a grid cell instead of having a single ND. Finally, the proposed technique is tested within a number of real indoor environments each of which is relatively large and unstructured. Table 1 shows the parameters used in the experiments.

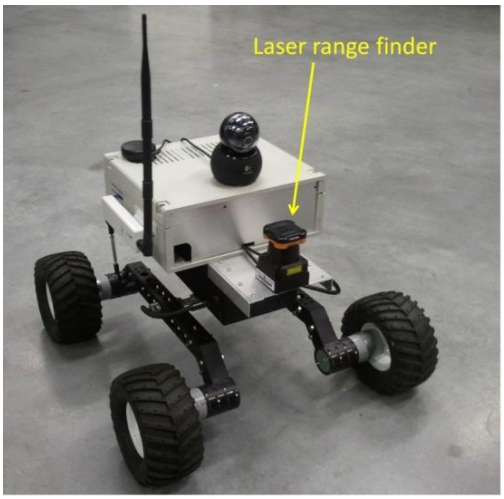


Figure 15. Ground mobile robot with laser range finder

Table 1. Default parameters for the scan-to-map matching technique

Parameters	**Value**
LRF scanning angle	0° to 180°
Scanning interval	0.5°
LRF scanning frequency	10 Hz
Grid cell size	$1\,m \times 1\,m$
Threshold (δ) for the scan-to-scan matching technique	0.001
Threshold similarity	0.3

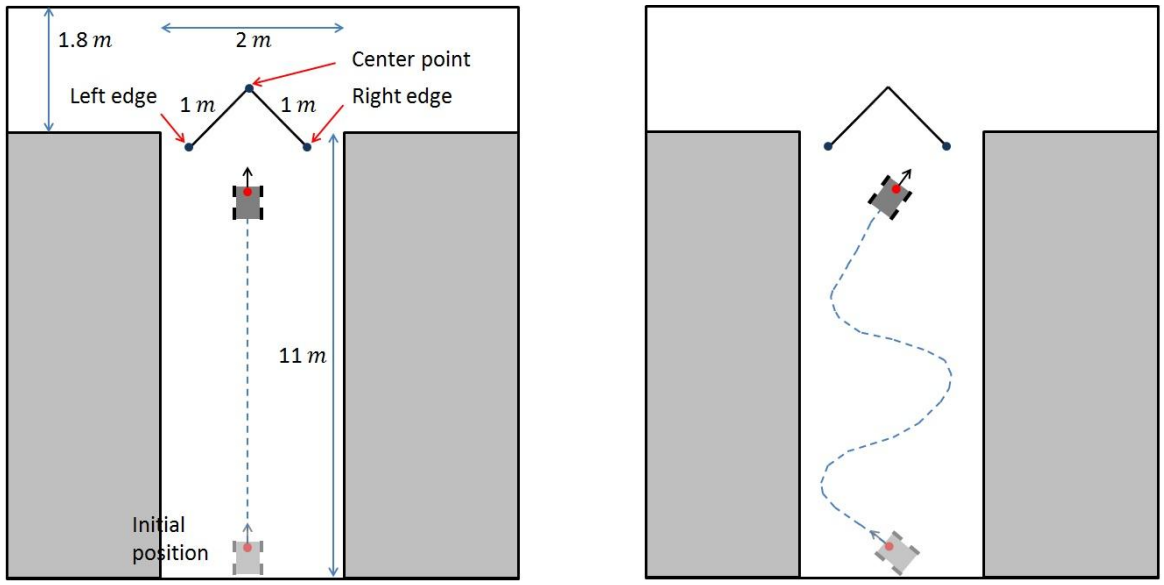


Figure 16. Experiment 1

4.5.1 Effect of the Scan-to-Map Matching

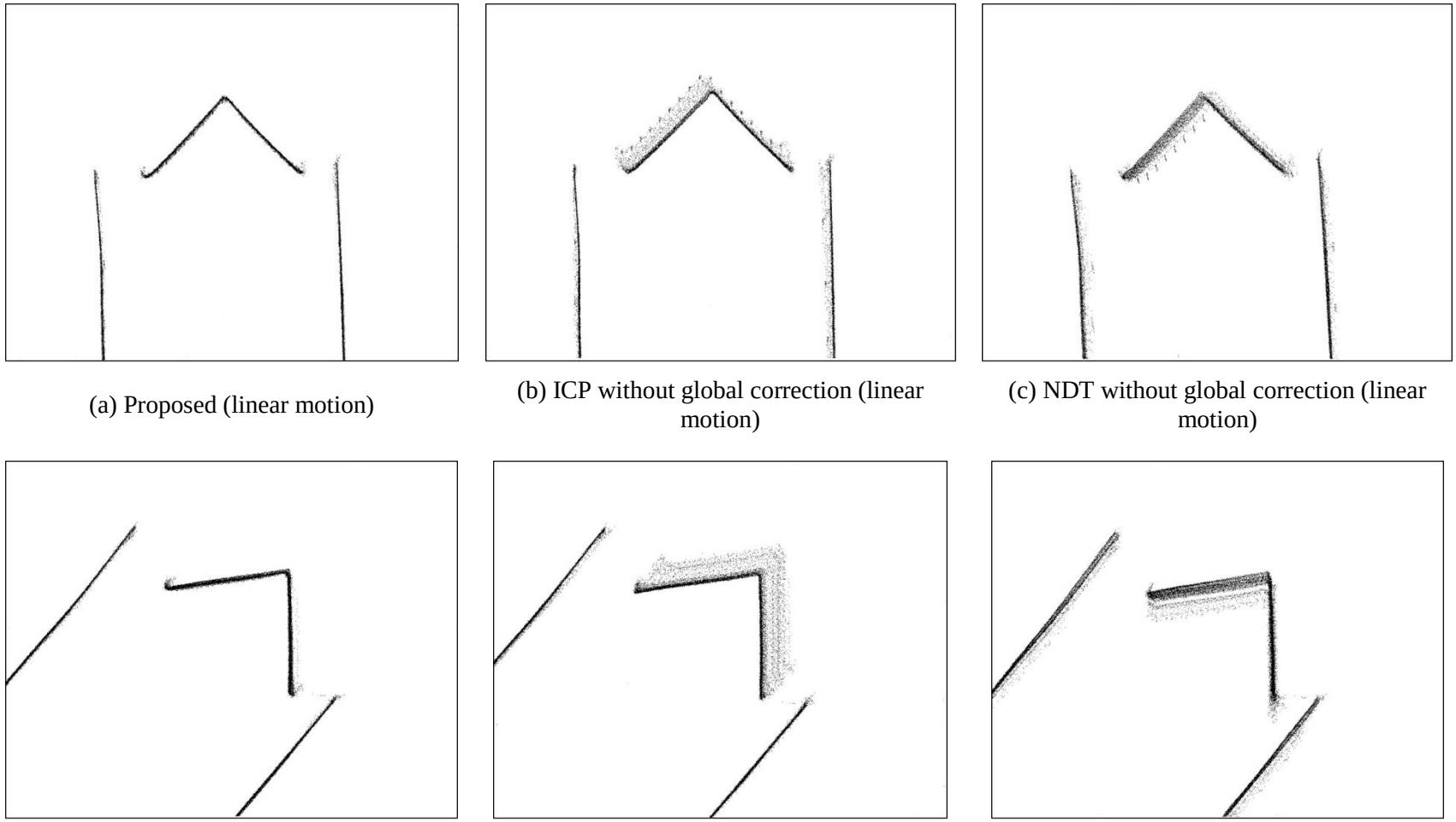
Figure 16 shows the first experiment where there is a L-shaped object at the end of a corridor. The robot was initially located at the starting point which was known in the global coordinate system. In order to exclude environmental parameters that might have influence on the experiment, the environment was selected to be simple. In the experiment the robot observed the entire object at all time and was manually driven along two different paths, one of which was a straight line and the other was a curvature. The robot took 190 scans and 348 scans for linear motion and nonlinear motion, respectively. At every acquisition of the new scan, the robot performed the scan-to-map matching and every scan points were mapped into the global coordinate system. Considering the left and right edge, and the center point of the object as detectable features, the position error, ε, at the left edge at time step k is given by

$$\varepsilon_k^l = \sqrt{\left(x_0^l - x_k^l\right)^2 + \left(y_0^l - y_k^l\right)^2} \qquad 4.17$$

where $\left[x_k^l,\ y_k^l\right]^t$ and $\left[x_1^l,\ y_1^l\right]^t$ are the position of the left edge mapped into the corrected global coordinate system at time step k and the initial position of the left edge, respectively. Simultaneously, the slope of the line connecting the center point and the left edge of the object is also calculated to see the orientation error.

Figure 17(a) and (d) show every consecutive scan after the scan-to-map matching for the linear and nonlinear motion, respectively. To address the effect of the proposed technique the figure also shows every scan point transformed to the global coordinate system after sequential scan-to-scan matchings by the ICP and NDT technique, but without the global correctness. As shown in the figure, for the proposed technique scanned points of the object from the initial time step to the end are well matched to one another showing that they are on the same positions as solid lines which are supposed to be the object and the walls. This indicates that local scan-to-scan matchings are well corrected by the map which is globally updated (see Equation 4.13 – 4.16) by new scans after the scan-to-map matching. The effect of the global correctness can be qualitatively verified by seeing the results of two scan-matching-only techniques. For the scan-to-scan matching techniques without the map matching, scan points do not lie on the same position, which indicates that new scans are not matched well to past scans and they are gradually away from the initial position as time goes by. In both cases of the linear and nonlinear motion the ICP technique generates relatively larger position error than the others, whereas the rotation error seems to be small since scanned points mapped at different time steps slide parallelly to one another. Similar to the ICP technique, the error produced by the NDT technique is mostly about the translation error, however, for the nonlinear motion walls next to the L-shape object seem to have several lines which are somewhat rotated.

Figure 18 and 19 quantitatively show the position error and the slope of the line connecting the center point and the left edge for the linear and nonlinear motion. As expected from the previous figure showing the accumulation of scans, the proposed technique shows the smallest position errors in both



(a) Proposed (linear motion)

(b) ICP without global correction (linear motion)

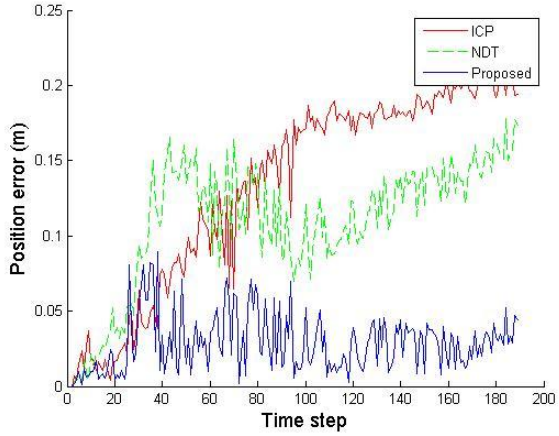
(c) NDT without global correction (linear motion)

(d) Proposed (nonlinear motion)

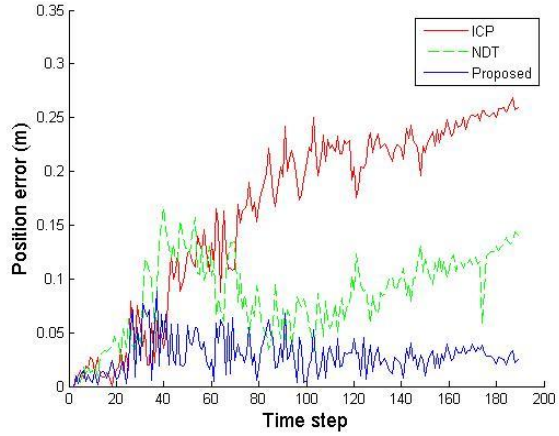
(b) ICP without global correction (nonlinear motion)

(c) NDT without global correction (nonlinear motion)

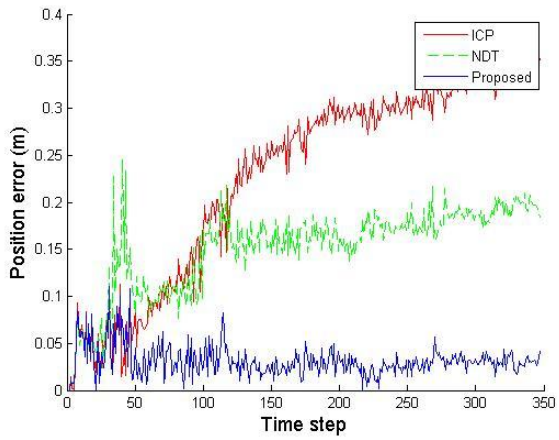
Figure 17. Accumulated scan points of the object transformed by the proposed, ICP, and NDT



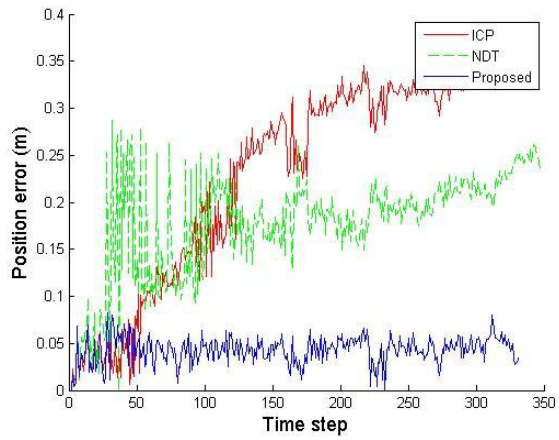
(a) Left edge (linear motion)



(b) Right edge (linear motion)



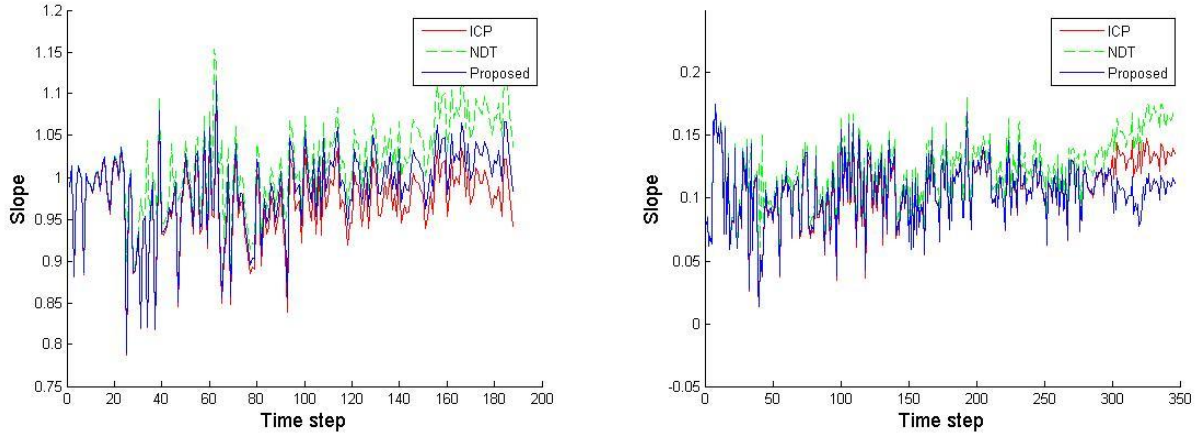
(c) Left edge (nonlinear motion)



(d) Right edge (nonlinear motion)

Figure 18. Position error of the left and right edge

motions. There is nearly no difference between the linear and nonlinear motion cases and the error is consistent in its value regardless of time step. The slope does not change a lot with respect to time, indicating that the orientation error is small and not accumulated with time. Note that these errors are caused not only by the matching process, but also by the LRF with the scanning interval of 0.5° which observes the features at different positions for each scan. However, when new scans are matched only to



(a) Linear motion (b) Nonlinear motion

Figure 19. The slope of the line connecting the center point and the left edge of the object

their previous scans using the ICP and NDT technique, position errors become large. For both the linear and nonlinear motion the ICP scan-to-scan matching technique shows the largest position errors which increase with time. Position errors by the NDT scan-to-scan matching are less than ICP, whereas, slopes at the initial and the last time step are slightly distinct, which implies there exist orientation errors. According to Table 2 the proposed technique has successfully removed the position errors with 6.67 and 4.62 times lesser errors than the ICP and NDT technique without the global correctness for the linear motion, and 9.62 and 5.64 times lesser errors for the nonlinear motion, respectively. It is important that the position errors do not accumulate with time step instead it stays within 4 cm.

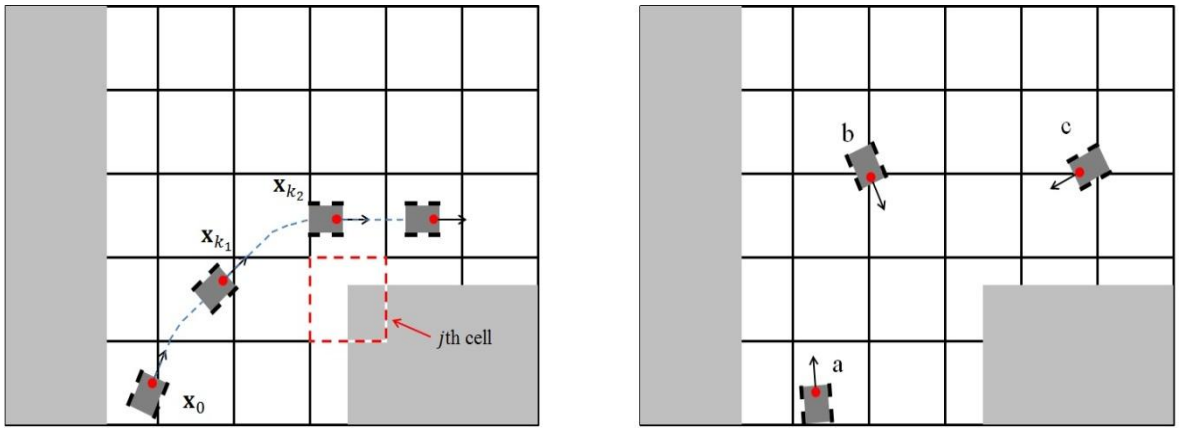
4.5.2 Effect of Multiple Normal Distributions in a Single Cell

Having identified the effect of global correction capability by the proposed scan-to-map matching technique, this experiment investigates the effectiveness of maintaining multiple NDs instead of a ND in a single grid cell. Unlike the previous experiment the robot possibly sees different parts of an object while it operates in a simulated environment, so that NDs from scan points in the *j*th cell can be largely different

Table 2. Initial and final positions of left (LE) and right edges (RE) estimated by three methods and position errors

			Proposed	ICP	NDT
Linear motion	Left edge	$\left[x_0^{f_l}, y_0^{f_l}\right]$	$[-0.85,\ 9.64]$		
		$\left[x_{190}^{f_l}, y_{190}^{f_l}\right]$	$[-0.89,\ 9.64]$	$[-0.81,\ 9.45]$	$[-0.99,\ 9.74]$
		$\varepsilon_{190}^{f_l}$	0.044	0.194	0.174
	Right edge	$\left[x_0^{f_r}, y_0^{f_r}\right]$	$[0.5,\ 9.69]$		
		$\left[x_{190}^{f_r}, y_{190}^{f_r}\right]$	$[0.52,\ 9.67]$	$[0.61,\ 9.45]$	$[0.42,\ 9.81]$
		$\varepsilon_{190}^{f_r}$	0.025	0.26	0.141
Nonlinear motion	Left edge	$\left[x_0^{f_l}, y_0^{f_l}\right]$	$[5.92,\ 7.85]$		
		$\left[x_{348}^{f_l}, y_{348}^{f_l}\right]$	$[0.589,\ 7.82]$	$[0.571,\ 7.57]$	$[5.9,\ 8.04]$
		$\varepsilon_{348}^{f_l}$	0.041	0.353	0.185
	Right edge	$\left[x_0^{f_r}, y_0^{f_r}\right]$	$[6.98,\ 6.97]$		
		$\left[x_{348}^{f_r}, y_{348}^{f_r}\right]$	$[6.94,\ 6.97]$	$[6.74,\ 6.71]$	$[6.95,\ 7.21]$
		$\varepsilon_{348}^{f_r}$	0.034	0.358	0.233

depending on the pose of the robot. The experiment is composed of two parts. First, the robot was given the initial pose, $\left[\mathbf{x}_0^t, \theta_0\ \right]^t$, and driven in a simulated environment shown in Figure 20(a). During the simulation the robot was able to see only one side of the corner at the initial time step, however, as it approached to the corner at time step k1 it began to observe both sides of the corner. Using the same data two different grid maps were independently updated where the first grid map maintained multiple NDs within each grid cell whereas the other grid map contained a single ND in each cell. Having the two grid maps after the robot fully stopped, the next is to locate the robot at three different positions and to compare similarities between map NDs and the scan NDs to find out what the most similar map ND is for each case (Figure 20(b)). For building the grid maps the robot took 43 scans, and at each time step during this period the similarity between the scan ND and the most similar map ND in the *j*th cell was calculated by Equation 4.7.

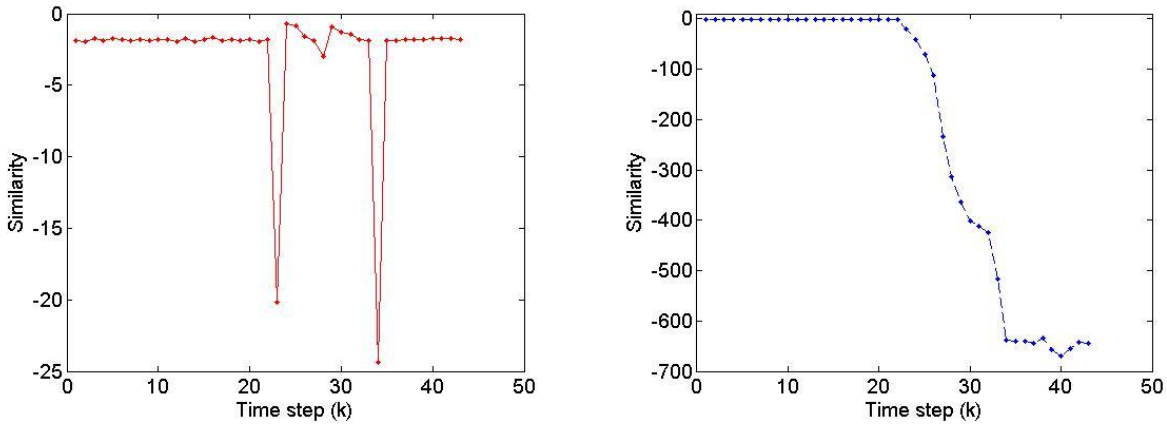


(a) (b)

Figure 20. Experiment 2

Figure 21(a) and 21(b) show the similarities between the scan NDs and the most similar map NDs in the jth cell when the grid map maintains multiple NDs and a single ND, respectively. For the multi-ND representation, when the robot first observes different side of the corner at time step k_1 (i.e. $k_1 = 23$), the similarity drops down to -20.17. However, the similarity immediately goes up at the next time step after the scan ND at time step k1 is added as a new map ND. Similarly, at time step $k = 28$ and k_2 (i.e. $k_2 = 34$), when the similarity is lower than the threshold value, a new map NDs is added and the similarity increases again at the next time step. When the grid map is updated using the single-ND representation, the similarity drops at time step k_1 and keeps decreasing since the existing map ND becomes less and less similar to the scan ND as the robot sees the other part of the corner more. The corresponding ND to the scan ND after time step k_1 are not appropriate, which may cause inaccurate matching between the scan ND and the map ND. Conclusively, the multi-ND representation in each cell can increase the accuracy of the scan-to-map matching.

After the grid map is updated by 43 scans using the multi-ND representation, the jth cell contains 4 map NDs. Having the grid map updated, Table 3 shows the similarities between the scan NDs and map



(a) Multi-ND representation (b) Single-ND representation

Figure 21. Similarity between the scan ND and the most similar map ND

NDs when the robot is placed at three different locations. When the robot is on Position (a) and obtains the new scan, the first map ND, updated by the first 22 scans, is selected as the most similar map ND to the scan ND with the similarity of -1.72. The last map ND, updated by the last 10 scans, has the minimum similarity to the scan ND. When the robot observes the corner from Position (b), both the second map ND, updated by the 23rd to 27th scans, and the third map ND, updated by the 28th to 33rd scans, are quite similar to the scan ND. Although the third map ND can be a good matching map ND for the scan ND, the second map ND is selected as the matching map ND, which enables the proposed technique to match the new scan more accurately to the grid map. When the robot is on Position (c), the result seems to be opposite to the first case, where the last map ND has the maximum similarity and the first map ND has the minimum value. Due to the multi-ND representation, the scan ND can find the exact matching map ND regardless of the pose of the robot.

4.5.3 Application to SLAM in Large Environments

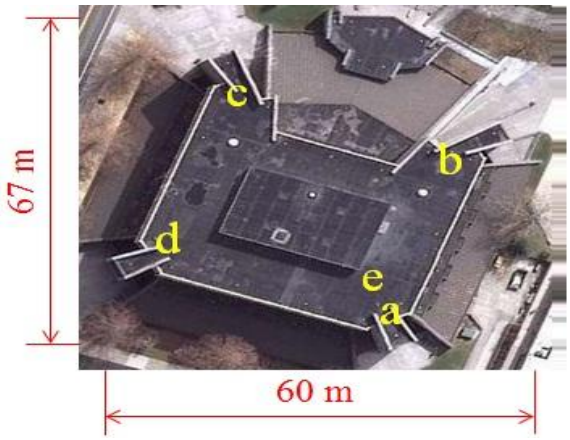
This subsection demonstrates high accuracy and versatility of the proposed technique in solving the

Table 3. Similarities between the scan ND and the map NDs

Position	Map ND 1	Map ND 2	Map ND 3	Map ND 4
a	-1.72	-2.89	-3.63	-98.37
b	-47.12	-1.12	-2.89	-61.38
c	-173.28	-4.27	-2.793	-1.732

SLAM problem in indoor environments shown in Figure 22. The environments were partially structured or unstructured with static and mobile objects including walking people. The environments had long corridors (Environment 2 and 3), a large number of random shaped objects such as chairs and desks (Environment 1, 2, and 3), and large loops (Environment 1,2 and 3) to be closed. During the experiment the robot took 2489, 2426, and 3873 scans for Environment 1, 2, and 3, respectively. Similar to the first experiment, since the autonomous exploration was out of the scope of this dissertation, the robot was manually driven following pre-determined paths (i.e. a-e-b-c-d-e for Environment 1, a-b-c-d-c-b-a for Environment 2, a-b-a-c-a-d-e-d-f-g-d for Environment 3).

Figure 23 shows trajectories of the robot in the three environments estimated by the proposed technique. In order to emphasize the effect of the global correction made by the proposed technique, trajectories of the robot estimated by the two scan-to-scan matching techniques without the global correction are also plotted in the same figure. As shown in the figure when the pose of the robot is computed by the proposed technique the robot successfully closes loops in the environments. For all the environment the total distances the robot travelled are 186 m, 258.2 m, and 374 m, and accumulated orientation changes of the robot are 3646°, 1949°, and 3729°, respectively. On the other hand, when the pose of the robot is estimated by the scan-to-scan matching techniques without the global correction, there exist deviations in the trajectories of the robot ending up with the failure in closing the loops. Table 4 shows the position and orientation of the robot estimated by the proposed technique as well as those by the two scan-to-scan matching techniques without the global correction at intersections in the environments. The poses of the robot by the ICP and NDT without the global correction are significantly

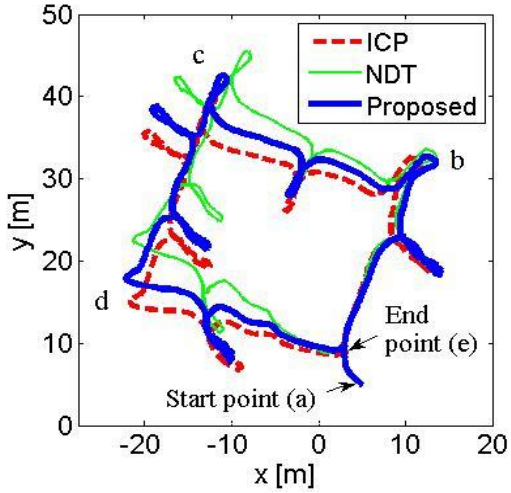


(a) Environment 1 : McBryde Hall

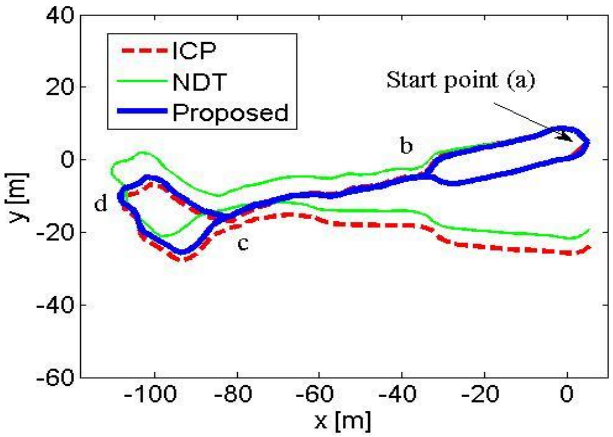
(b) Environment 2 : Newman library

(c) Environment 3 : Randolph Hall

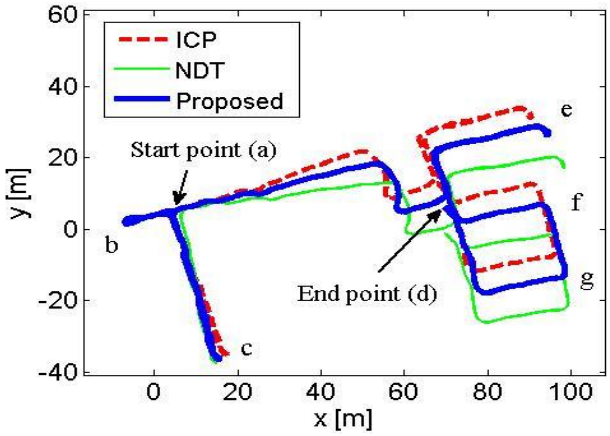
Figure 22. Test environments



(a) Environment 1



(a) Environment 2



(a) Environment 3

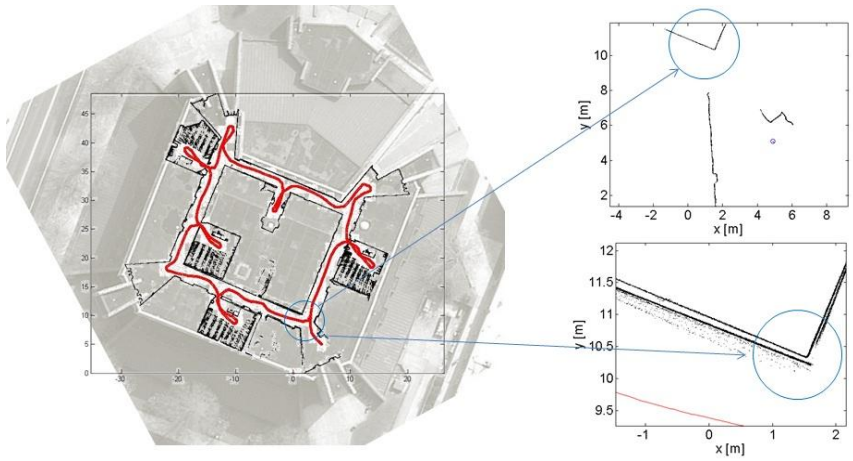
Figure 23. Trajectory of the robot for all environments

Table 4. Robot pose estimated by three techniques

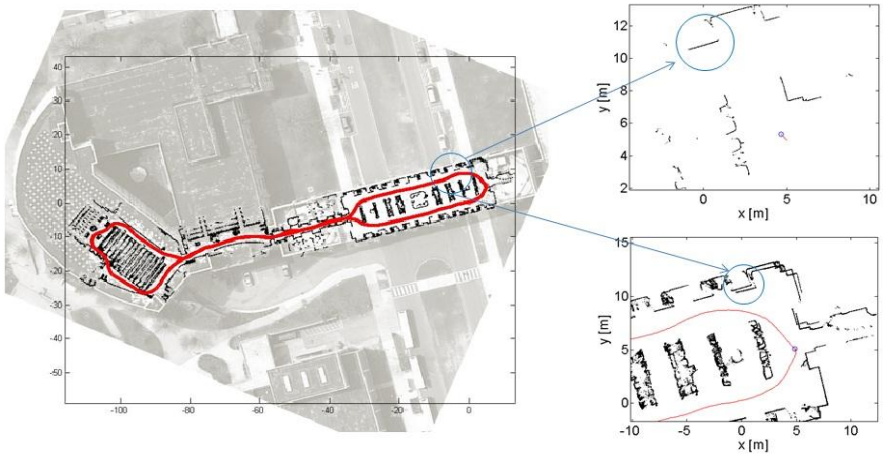
Environment	Proposed	ICP	NDT
1 (Point e)	$[2.87, 9.63, -40.23^{\circ}]$	$[3.24, 9.07, -34.45^{\circ}]$	$[2.1, 9.23, -51.44^{\circ}]$
2 (Point c)	$[-82.33, -15.83, -65.1^{\circ}]$	$[-85.02, -20.56, -56.46^{\circ}]$	$[-85.18, -14.26, -78.95^{\circ}]$
3 (Point a)	$[7.67, 4.12, -62.1^{\circ}]$	$[8.01, 4.79, -60.5^{\circ}]$	$[9.15, 3.97, -69.1^{\circ}]$

different from the pose by the proposed technique. For all three environments the orientation of the robot are very different at the intersections, which means the trajectories of the robot after these intersections become significantly different.

In Figure 24 maps of all three environments built by the proposed technique are shown. The localization error can be qualitatively analyzed by seeing the quality of the map since the localization and the map building are directly related to each other. In order to have better understanding of the accuracy, each map is overlapped on top of the satellite view of each environment where a red line is the trajectory of the robot. No other additional techniques such as the loop closure are used. For all the environments the results demonstrate the high accuracy of the proposed scan-to-map matching technique in real SLAM scenarios. When the robot visits the same areas in the environments more than once, it observes the same features more than once. The accumulated pose error is then roughly calculated by comparing the positions of the same features observed at different time steps. As shown in the figure the intersections that the robot has visited more than once are zoomed-in and the features in the intersections are highlighted. In each environment there exists small amount of a position error (Environment 1), both a position and an orientation error (Environment 2), or nearly no error (Environment 3) after the robot comes back to the same area. In Table 5 numerical results of the position and orientation errors from the proposed technique are listed. For all the environments the error rates defined as the position or the orientation error divided by the total distance or the accumulated orientation changes are less than 0.001, which indicates that the position and the orientation errors are less 10 cm and 0.1° after travelling 100 m



(a) Environment 1



(a) Environment 2

(a) Environment 3

Figure 24. Maps of test environments by the proposed method overlapped on satellite view of the environments

Table 5. Position and orientation error produced by the proposed technique in Environment 1,2, and 3

Environment	Total distance / orientation changes	Error	Error rate
1	186 m	12.2 cm	0.000656
	3646°	0.64°	0.000176
2	258 m	25 cm	0.00097
	1594°	1.17°	0.00073
3 (Point a)	112 m	0 cm	0
	1240°	0°	0
3 (Point d)	182 m	0 cm	0
	1770°	0°	0

with the accumulated orientation changes of 100°. The results show that the proposed technique is successful in estimating the robot pose and building the map in three different real environments.

4.6 Summary

In this chapter a grid-based scan-to-map matching technique for accurate SLAM has been proposed. The proposed technique performs the local scan-to-scan matching and corrects the error from the matching by the global scan-to-map matching. The map to match is a grid map which may hold multiple NDs within each grid cell. Due to the scan-to-map matching with the multi-ND representation, the proposed technique exhibits little errors in the scan-to-map matching and does not further accumulate the errors. The new scan is also represented by NDs enabling the novel ND-to-ND matching. The equal treatment of cells in the proposed ND-to-ND matching could further contribute to the accurate global scan-to-map matching.

The proposed scan-to-map matching technique was applied to three different experiments. The first experiment investigated the proposed scan-to-map matching technique in terms of the position and orientation error while the robot moved and sequentially obtained new scans. The experiment showed that the accumulated orientation error was negligible and that the position error stayed within 4 cm after travelling around 10 m. The second experiment, investigating the effectiveness of maintaining multiple

NDs within a cell, showed that the scans were matched better to the map with higher similarity when the cell had multiple NDs than one ND. The experiment also demonstrated the robust effect of the use of multiple NDs. Finally, the proposed technique was applied to the SLAM in three real environments to demonstrate its applicability to real problems. The resulting maps showed that the proposed technique without any post processes such as the loop closure generated position errors in the order of ten centimeters with very small orientation errors for the three environments after travelling around 200 m with large orientation changes.

Chapter 5

Map-based Semi-Autonomous Robot Navigation

This chapter presents a map-based semi-autonomous robot navigation using tele-operation which is primarily used to navigate a mobile robot in certain situations where human access as well as visibility is very limited. In the map-based robot navigation, the pose of the robot is estimated by the scan-to-map matching technique presented in Chapter 4, and sent to the base station together with the map. For a better understanding of the environment, the on-site robot is also capable of creating three-dimensional map with additional LRF. At the base station, a human operator controls the robot via tele-operation based on the received map. Meanwhile the operator is provided with possible waypoints for the robot navigation by computer aided guidance. Additionally, in case of communication loss, which frequently happens in harsh environments, the robot autonomously returns to the home position. The performance and the usefulness of the navigation strategy will be demonstrated using a ground robot in an artificial disaster area with an implementation of a graphical user interface (GUI).

5.1 Introduction

For the past several decades tele-operated systems (Figure 25) have been used for a variety of different purposes. In the early era, the tele-operation was simply remote connection between human operators and machines [89]. Together with outstanding improvement of the artificial intelligence and its application to mobile robots, remote control systems are now able to be used for more complex situations such as disaster relief. In such scenarios, the environments are usually inaccessible, dangerous and sometimes unknown where wireless network is not always guaranteed. For these reasons there have been increasing demands for mobile robots with advanced sensors and intelligence. This not only gives more information on the environment to human operators, but also allows robots to have more freedom while in operation, so that they can effectively deal with uncertainties caused by the environment. In order for the tele-operated navigation to be used in unknown areas, it is required that the information the robot collects should be useful and easy to recognize. In addition, the human operator at the base station also needs information on the status of the robot at all time. This includes how the environment looks like, where the current location of the robot is, and whether the wireless communication is alive, etc.

Many different approaches were developed for remote navigation, using different types of sensors and functionalities. Using vision sensors is one common solution to the navigation system considering their light weight and low power consumption [90]. This approach relies on one or more cameras where the human operator uses live video images as prior knowledge and controls the mobile robot based on the information [91, 92]. It is cost efficient [93], and also useful for obstacle avoidance. However, there are possibilities that in unstructured areas images from the vision sensor are not reliable, recognizable, and even distorted. And it is assumed that there exist time delays in sending the images and limitation on bandwidth. Due to the restrictions above plus the limitation on the field of view, the vision based method is also used in combination with other sensors [94, 95]. This provides both to the human operator and the robot itself with more information, making it possible that the operator gives more autonomy to the robot. However, at the same time, it is expansive with heavy computations and

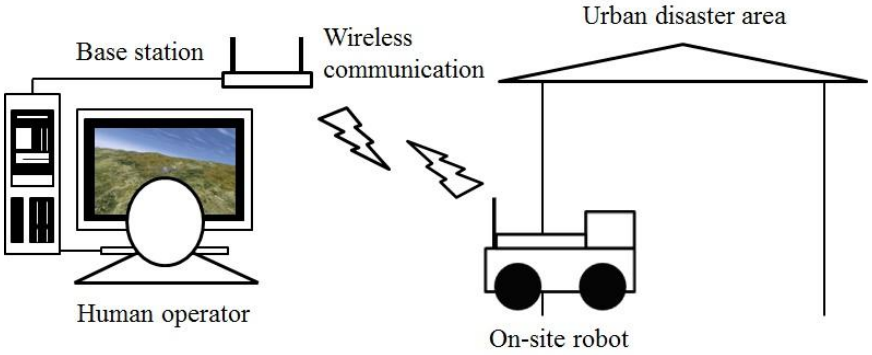


Figure 25. The concept of a tele-operated system

sometime the artificial intelligence prevents the human operator from making good decisions especially in noisy and unstructured environment.

Another approach is to use LRFs with the implementation of SLAM [96]. Since the mobile robot is supposed to complete different types of missions, it is very important for the human operator at the base station to understand the map and the location of the robot. In this respect, this method takes an advantage of the ease of building relatively accurate map.

5.2 Tele-operated Navigation Strategy

Figure 26 shows the schematic diagram of the strategy for the map-based robot navigation. Once a decision is made by the human operator, the base station is in charge of sending commands to the robot. The operator usually sends either “next waypoint” command or “autonomous return” command that makes the robot return to the home position. When sending commands, it is required to check if the wireless communication is alive, otherwise commands never get to the robot. If the communication is available, the robot receives the command and starts doing waypoint navigation. In case of “autonomous return” command, the robot automatically calculates its waypoints to go back to the home position. During the waypoint navigation, the robot computes its own position and orientation and creates the map based on the scan-to-map matching technique, and sends its pose and the map to the base station.

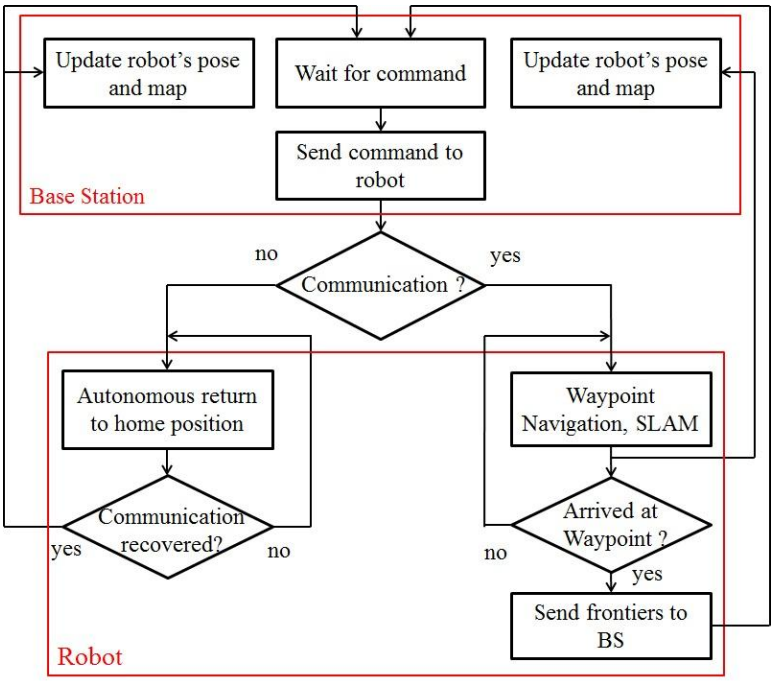


Figure 26. Overview of the strategy for map-based robot navigation using tele-operation

The robot stops at the waypoint when the below criterion is met.

$$d(\mathbf{x}_k, \mathbf{x}_k^w) < \rho \qquad 5.1$$

where $d(\cdot,\cdot)$ denotes the distance between two points, $\mathbf{x}_k^w$ is the next waypoint at time step k, and ρ is a threshold value. Upon completion of the waypoint navigation, the robot sends a set of frontiers which is defined as regions on the boundary between open space and unexplored space [97].

5.2.1 Frontier-based Guidance

We introduce this concept and formulate how to select and evaluate frontiers to assist the human operator by providing distance-based information where the possible next waypoints are. This additional knowledge takes advantage since it is difficult for humans to identify open spaces only with vision sensors. Once the robot arrives at the waypoint, it finds new frontiers and adds them to the set of frontier

candidates if there are any. The selection of frontiers is described in Algorithm 1. Note that a new frontier is added to the set of frontiers when two conditions are met.

1) Distance difference between two neighboring sample points is greater than β,

2) Each cluster of two neighboring sample point has more than γ elements in it.

Through the two steps the incorrect registration of frontiers caused by the environmental condition or the hardware constraints can be removed. At the last waypoint, the robot updates the set of frontiers, and computes scores for the frontiers. For the ith frontier, the score, s_k^i, is given by

$$s_k^i = d\left(\mathbf{x}_k, \mathbf{x}_k^{f_i}\right) + \omega\, \mathbf{g}\left(\mathbf{x}_k^{f_i}, E\right) \tag{5.2}$$

where $\mathbf{x}_k^{f_i}$ is the position of the ith frontier and ω is a weight. $\mathbf{g}\left(\mathbf{x}_k^{f_i}, E\right)$ is defined as

$$\mathbf{g}\left(\mathbf{x}_k^{F_i}, E\right) = \begin{cases} 1 & \text{if } \min\left\{d\left(\mathbf{x}_k^{f_i}, \mathbf{x}_k^{e_j}\right) \mid \forall j \in \{1, \cdots, n^e\}\right\} < \lambda \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

where $E = \left\{\mathbf{x}^{e_j} \mid \forall j \in \{1, \cdots, n^e\}\right\}$ is the set of frontiers that the robot has already explored and n^e is the number of explored frontiers. The frontier which has the minimum score becomes a strong candidate for the next waypoint.

5.2.2 Autonomous Return

To be able to operate properly in inaccessible environments, the robot has two different modes, namely, “tele-operation” mode and “autonomous-return” mode. In “tele-operation” mode the robot is controlled by the human operator whereas in “autonomous-return” mode the robot is designed to autonomously return to the home position. The system starts with “tele-operation” mode, however, it turns to “autonomous-return” mode when the operator forces the robot to return or the communication is lost.

Algorithm 1 : Selection of frontiers

Inputs : $\mathbf{x}_k$, $\mathbf{z}^{in}$

for $i := 1, \cdots, n$ **do**
    // compute distance to each end point
    $d^i := \left\| [x_k, y_k]^t - \mathbf{z}_i^{in} \right\|$
Collect all index i where $d^i > r_{max}$ and group neighboring indices into same cluster, $\mathbf{C}$
for $j := 1, \cdots, nc$ **do**   // where, nc is number of clusters
    // number of elements of jth cluster
    if $n\{\mathbf{C}^j\} > \alpha$
        frontier $= \mathbf{z}_m^{in}$ // where, index m is the median $cluster^j$
$flag := 0$
for $i := 1, \cdots, n-1$ **do**
    // find difference of distances of two consecutive end points
    if $d^{i+1} - d^i > \beta$
        if $flag = 1$ **and** $count < \gamma$
            Remove newly added $frontier$
        $flag := 0$
        if $count \geq \gamma$
            frontier $= \left(\mathbf{z}_i^{in} + \mathbf{z}_{i+1}^{in}\right)/2$
            // $flag$ is 1 when new $frontier$ is added
            $flag := 1$
    $count := 0$
    else
        $count := count + 1$

output : **frontier**

To certify the network connectivity, the base station sends a dummy message at a regular interval and the robot checks the message every one second. A counter triggers if the robot fails to read the message and the robot makes itself get ready for autonomous return to home position after receiving no message until the counter reaches 20.

The process of returning to the home position is straight forward. While the robot is in "tele-operation" mode, it sequentially stores all the waypoints given by the human operator when it arrives at the new waypoint. A waypoint as a single node can be connected to other waypoints if the waypoint is reachable from the other waypoints. The adjacency matrix for all waypoints can be created and it shows

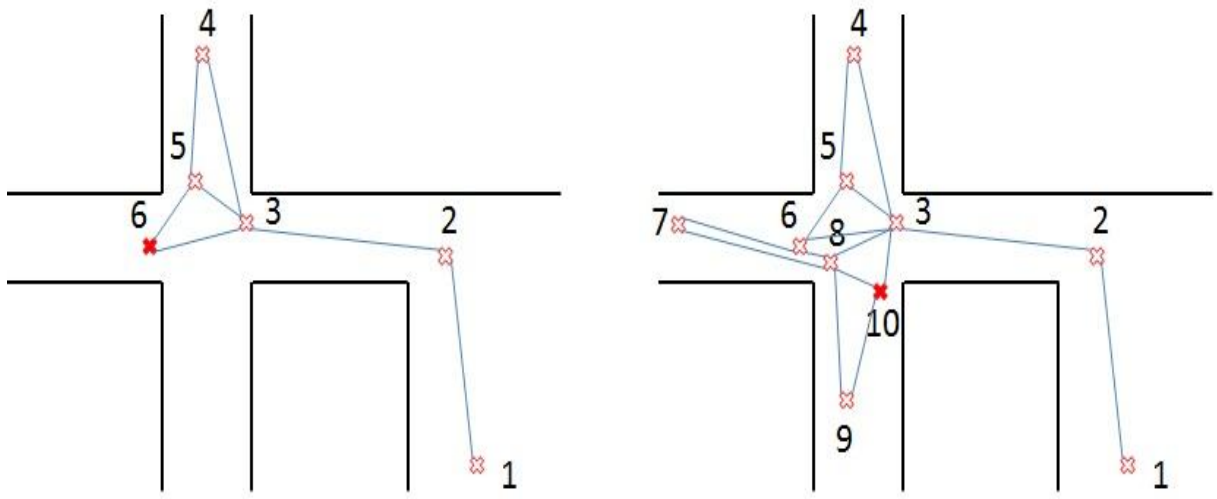


Figure 27. Waypoints in the order that the robot visited and the connections between waypoints after the robot has arrived at waypoint 6 (left) and waypoint 10 (right)

which waypoints are neighboring to each other. When the communication gets lost or the operator wants the robot back, the robot calculates the optimal path to the starting position by inversely tracking the stored waypoints. In Figure 27, for example, the wireless network is lost after the robot has cleared the waypoint 10. Without the optimal path planning it can simply go back to the starting position by following the trajectory it travelled (i.e. 10-9-8-7-6-5-4-3-2-1). Knowing the connection between all the waypoints, the robot can reduce efforts to return to the home position (i.e. 10-3-2-1). In this process, obstacle avoidance is also applied to prevent the robot from any collisions.

5.3 Graphical User Interface

In order to enhance the efficiency of controlling the on-site robot introducing a GUI is required. The GUI allows the human operator to directly interact with the robot without typing any command line arguments, and clearly shows information sent from the robot. The GUI for the map-based robot navigation is composed of Display Panel (DP) 1 and 2, Toolbox, Control Option, and Status Bar (Figure 28). On DP1 the map of the entire environment in the corrected global coordinate frame is displayed, whereas on DP2

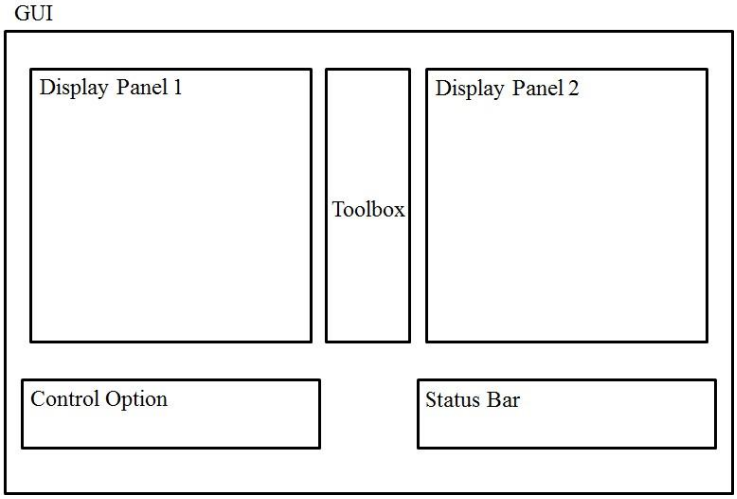


Figure 28. Graphical user interface

only part of the environment with the zoomed-in map is displayed. DP2 is also used when selecting a new waypoint. Due to the zoomed-in map the operator can see the details of the environment around the robot, so that the operator has less chance to select a wrong position for the new waypoint. In Toolbox the operator can choose what to display in display panels such as the trajectory of the robot, and the list of waypoints. Other functionalities such as rotating, zooming in/out the map and selecting a new waypoint are also available in Toolbox. The new waypoint is automatically sent to the robot after it is selected by the following steps:

1) Check Adding Waypoint button in the toolbox,

2) Move the cursor to DP2,

3) Click a point on the map to which the operator wants to navigate the robot.

Control Option has Start and Stop buttons to initiate or shut down the robot, and the operation mode of the robot can be manually switched from “tele-operation” mode to “autonomous-return” mode or vice versa in Control Option. Status Bar briefly shows the current robot pose, the velocity of the robot, the communication status, and the distance to the current waypoint.

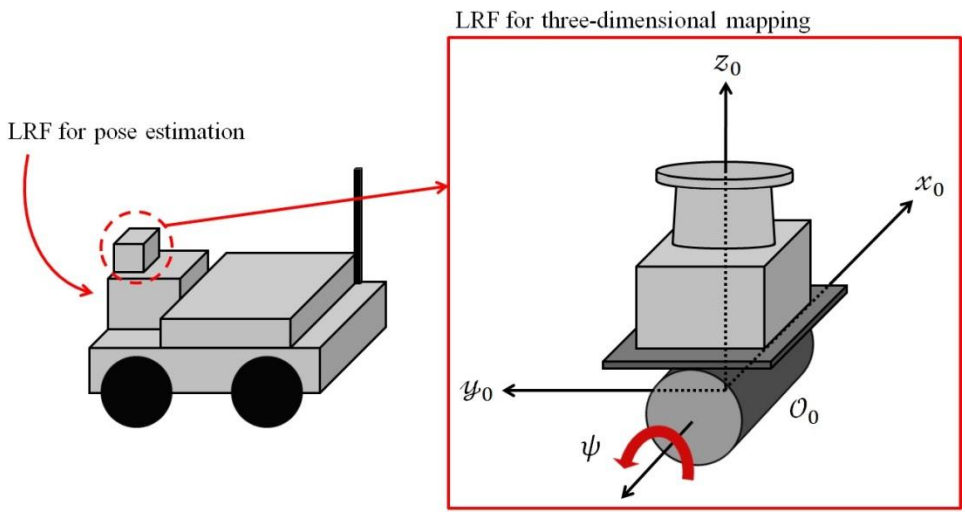


Figure 29. Two LRFs mounted on the robot (left) and the LRF for three-dimensional mapping

5.4 Three-Dimensional Mapping

Figure 29 is a schematic drawing of two LRFs on the robot, one for the two-dimensional SLAM and the other for the three-dimensional mapping, where the second LRF is mounted on a unit which is driven by a servo motor. In the figure x_0, y_0, and z_0 denotes the axes of the servo motor coordinate system, $\{S\}$, with the origin $\mathcal{O}_0$. Note here that the unit is rigidly fixed to the robot, thus the orientation of y_0 is equal to the orientation of the robot. The three-dimensional mapping begins with transforming a scan of the second LRF in the second LRF coordinate system, $\{R'\}$, to that in the $\{S\}$ coordinate system. Figure 30 shows the transformation using the Denavit-Hartenberg (D-H) convention, where there are three intermediate coordinate system with the origins of $\mathcal{O}_3$, $\mathcal{O}_2$, and $\mathcal{O}_1$. The transformation matrix, ${}^{\{S\}}_{\{R'\}}\mathbf{A}$, that transforms the scan from the $\{R'\}$ to the $\{S\}$ coordinate system is given by

$$ {}^{\{S\}}_{\{R'\}}\mathbf{A} = \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4 \qquad 5.4 $$

where $\mathbf{A}_i$ represents the transformation from the coordinate system of $\mathcal{O}_i$ to $\mathcal{O}_{i-1}$, and the details on the

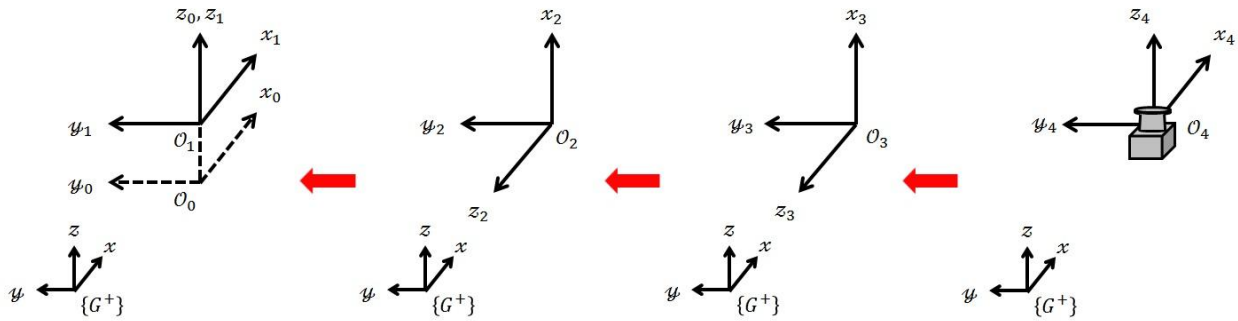


Figure 30. Transformation using the Denavit-Hartenberg convention

computation ${}^{\{S\}}_{\{R'\}}\mathbf{A}$ is shown in Appendix 2. According to the D-H convention, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$ and $\mathbf{A}_4$ are given by

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1^{DH} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{A}_2 = \begin{bmatrix} \cos\left(-\frac{\pi}{2}\right) & 0 & -\sin\left(-\frac{\pi}{2}\right) & 0 \\ 0 & 1 & 0 & 0 \\ \sin-\left(\frac{\pi}{2}\right) & 0 & \cos\left(-\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{A}_3 = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{A}_4 = \begin{bmatrix} \cos\left(\frac{\pi}{2}\right) & 0 & -\sin\left(\frac{\pi}{2}\right) & 0 \\ 0 & 1 & 0 & 0 \\ \sin\left(\frac{\pi}{2}\right) & 0 & \cos\left(\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where d_1^{DH} is the distance between $\mathcal{O}_0$ and $\mathcal{O}_1$, and ψ is the rotation by the servo motor.

Having identified the three-dimensional transformation matrix, the i-th new scan point in the $\{R'\}$ coordinate system, $\left[{}^{\{R'\}}z_k'^{\,x_i}, {}^{\{R'\}}z_k'^{\,y_i}\right]^t$, can be further transformed to that in the three-dimensional $\{G^+\}$ coordinate system using the robot position:

$$
{}^{\{G^+\}}\mathbf{z}_k'^{\,i} = \begin{bmatrix} {}^{\{G^+\}}z_k'^{\,x_i} \\ {}^{\{G^+\}}z_k'^{\,y_i} \\ {}^{\{G^+\}}z_k'^{\,z_i} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \end{bmatrix} \begin{bmatrix} {}^{\{R'\}}z_k'^{\,x_i} \\ {}^{\{R'\}}z_k'^{\,y_i} \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} {}^{\{G^+\}}x_k \\ {}^{\{G^+\}}y_k \\ 0 \end{bmatrix} \qquad 5.5
$$

where ${}^{\{G^+\}}\mathbf{z}_k'^{\,i}$ is the ith scan point of the second LRF in the $\{G^+\}$ and A_{mn} is the entry of ${}^{\{S\}}_{\{R'\}}\mathbf{A}$ in the m-th row and the n-th column. For simplicity the translation between the position of the robot and $\mathcal{O}_0$ is assumed to be $\mathbf{0}$.

5.5 Results

Experiments were conducted using the same robot shown in the previous chapter with Ubiquiti Networks PicoStation2HP to improve the signal reception. Test areas shown in Figure 31 are artificial disaster areas and totally unknown both to the robot and the human operator. The robot was located near the entrance of the environment at the beginning, since it is the most reasonable starting position in real situation, although the robot can be placed on any arbitrary position in the test environment. The base station was located in the lab which is approximately 20m away from the stating position of the robot. Technically, the wireless network consisting of a wireless router and the wireless access point can cover up to 100 m, however, the connection was usually lost beyond 80m in this test environment.

Figure 32 is the screen capture of the GUI after the robot completed to explore the test environment. Display panels show the resulting map created by the robot and the status bar the robot is still in tele-operation mode. Considering the quality of the map, pose estimation of the robot is good. The initial position of the robot is $[0,0]^t$ in the corrected global coordinate system, and the total distance the robot travelled is 92.47 m. In the map on the DP2, the trajectory of the robot and all waypoints human operator has assigned to the robot are displayed.

Figure 33 shows frontiers and corresponding scores when the robot is located at the initial position and the third waypoint during the exploration of the test areas. At the initial position there is only one

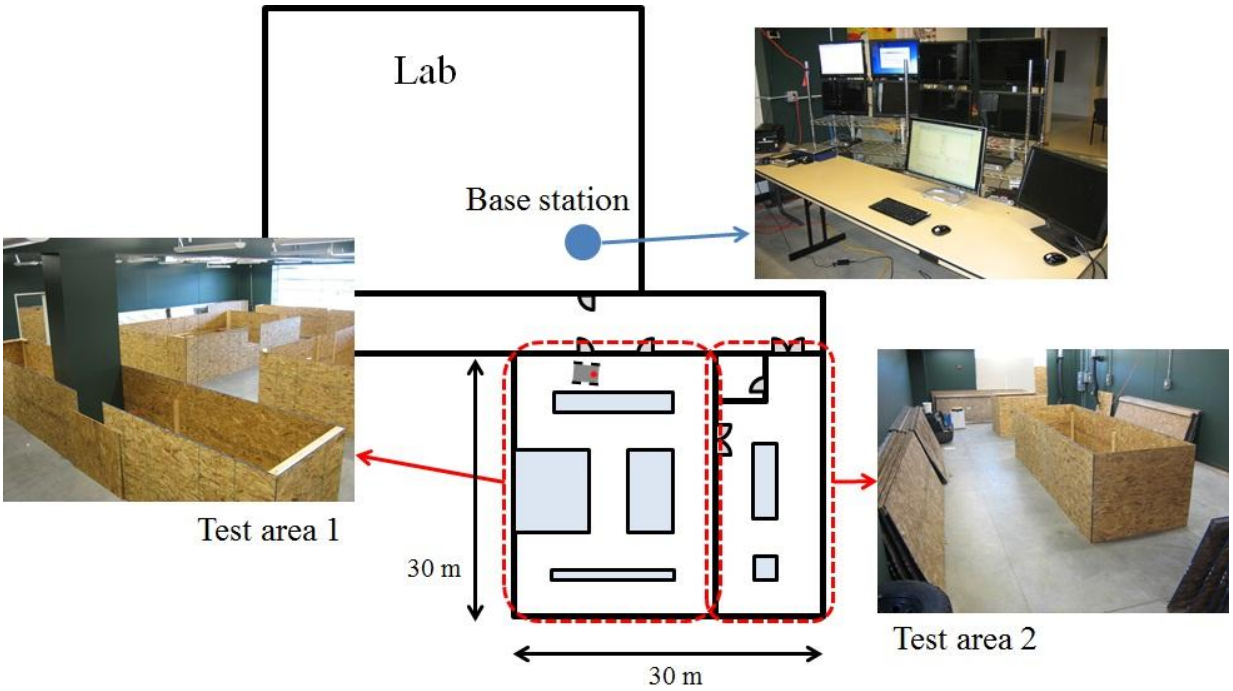


Figure 31. Test areas

frontier with score of 5.09 and at the third waypoint there are three frontiers with scores of 12.3, 12.9, 14.7, respectively. The scores can change depending on the weight in Equation 5.2 or the threshold value, λ, in Equation 5.3, where both ω and λ are set to 8 in this experiment. During the map-based robot navigation in the test environment, there was no loss of wireless connection between the base station and the robot. To check if the robot can autonomously return to the home position, the wireless connection was manually cut after the robot has completed to explore the entire area. The robot waited for 20 seconds trying to reconnect to the base station and it automatically switched its operation mode to "autonomous-return" mode. From the last waypoint, or the 44th waypoint, it calculated the optimal path to the home position as presented in Section 5.2. As a result, the 28th, 27th, 4th, 3rd, 2nd, 1st waypoint and the initial position were followed by the 44th waypoint. While the robot was in "autonomous-return" mode the human operator could not see the robot in the map and the map did not update until the wireless connection recovered again.

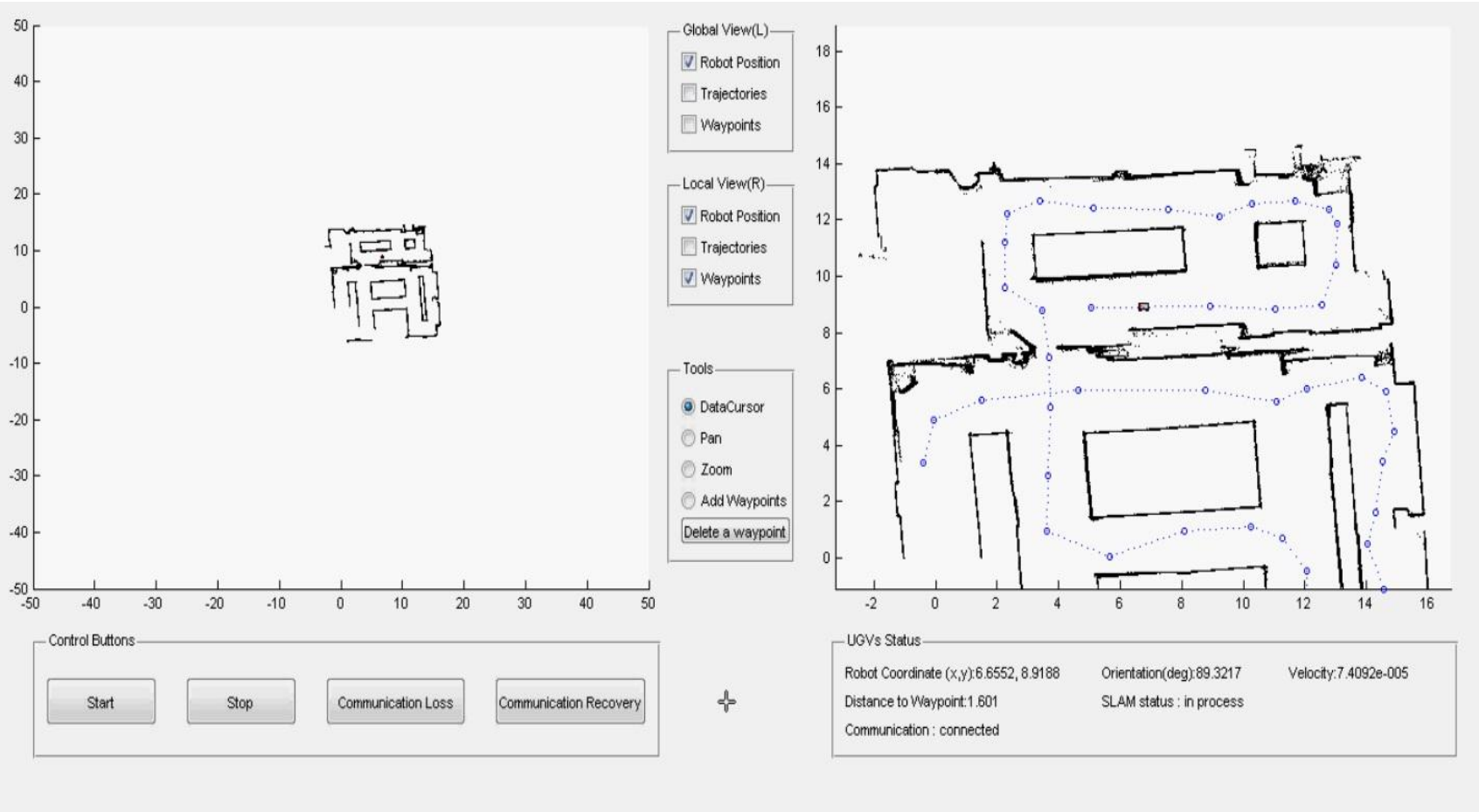


Figure 32. Screen capture of the GUI

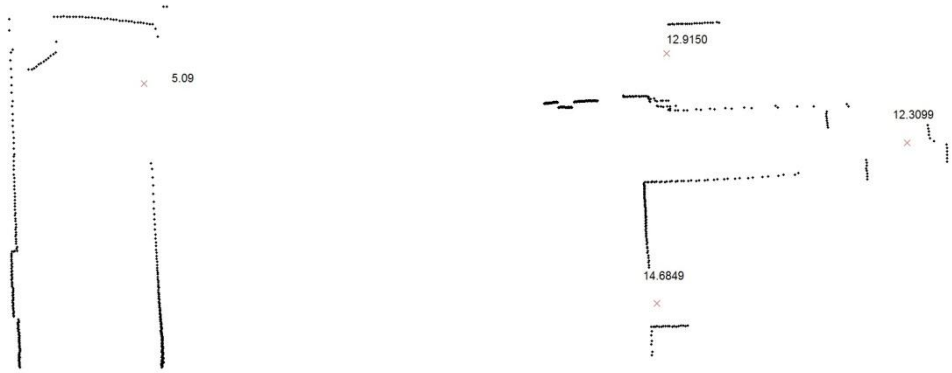


Figure 33. Computed frontiers when the robot was at home position (left), and at the third waypoint (right)

Figure 34 shows the three-dimensional map the on-site robot created during its exploration. As shown in the figure, the scan points obtained by the second LRF are mapped onto the three-dimensional $\{G^+\}$ coordinate system using the robot pose. Therefore, the accuracy of the three-dimensional map is equivalent to that of the two-dimensional map. In order for the human operator to have a better understanding of the environment, the scan points are in different colors according to their heights from the ground. With the three-dimensional map, the human operator can better deal with possible changes in the environment, which also affects the navigation of the robot.

5.6 Summary

In this chapter the map-based mobile robot navigation has been presented. The robot navigation has been achieved via tele-operation with an implementation of the GUI and tested in indoor test areas. Based on the currently proven state-of-the art technique, the proposed approach has successfully demonstrated its capability as an immediately applicable solution to USAR scenarios. The effectiveness of the frontier-based guidance and autonomous return to home position are also demonstrated. Due to the accurate pose estimation, the quality of the three-dimensional map is good enough to be effectively used for better understanding of the environment.

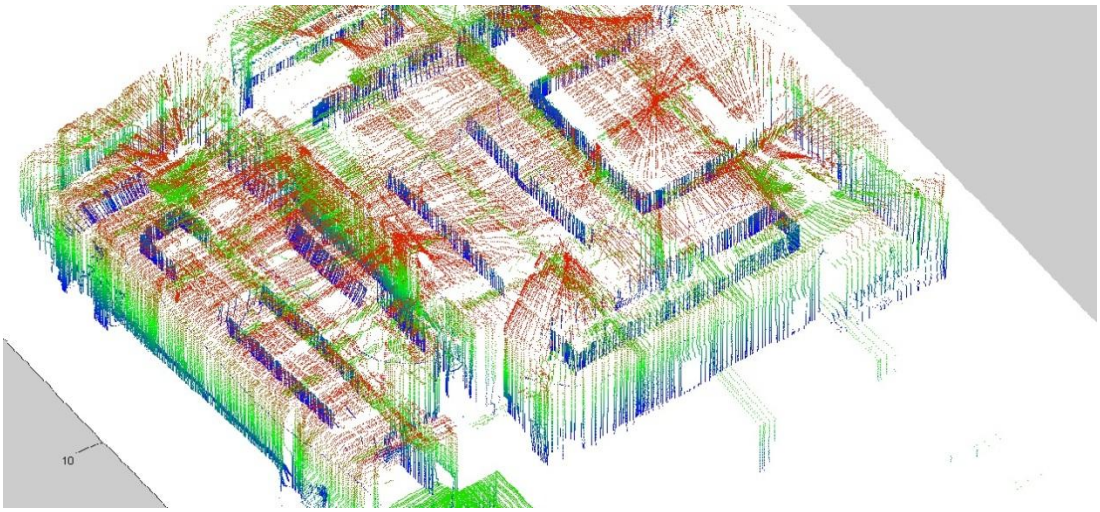


Figure 34. Three-dimensional map created by the robot

On the other hand, for the improvement of the robot navigation it is required to deal with full three-dimensionality doing the three-dimensional SLAM. This is because in real disaster areas the ground robot has to navigate not only flat surfaces but also slopes and rough surfaces, which means the two-dimensional SLAM is likely to create an inaccurate map. The enhancement of artificial intelligence is also desirable to deal with uncertainties in real world.

Chapter 6

Fully-Autonomous Cooperative Search, Tracking, Localization and Mapping

This chapter presents a fully-autonomous search, tracking, localization and mapping (STLAM) for USAR using multiple robots. Each robot is capable of estimating its pose and creating the map. As a team, the robots communicate each other while sending collected information to the base station. Since the environment the robots need to explore is assumed to be unknown, the robots search for objects of interest (OOIs) based on the area coverage method with the frontier-based exploration. Once the OOIs are detected, the robots keep tracking them where their positions are estimated and corrected by the EKF. Unlike the semi-autonomous strategy presented in Chapter 5, the team of robots explores the environment, searches for and tracks the OOIs without any control from the human operator. The cooperative performance of multiple autonomous robots can be evaluated by linking computers each to be mounted on a real robot to the network and running and testing autonomous robots in a virtual environment using the developed platform- and hardware-in-the-loop (PHILS) simulator. The results demonstrated the applicability of the proposed fully-autonomous strategy to the real USAR scenarios.

6.1 Object and Robot Model

Consider the jth OOI, o_j, the motion of which is discretely given by

$$\mathbf{x}_k^{o_j} = \mathbf{f}^{o_j}\left(\mathbf{x}_{k-1}^{o_j}, \mathbf{u}_k^{o_j}\right) + \mathbf{w}_k^{o_j} \tag{6.1}$$

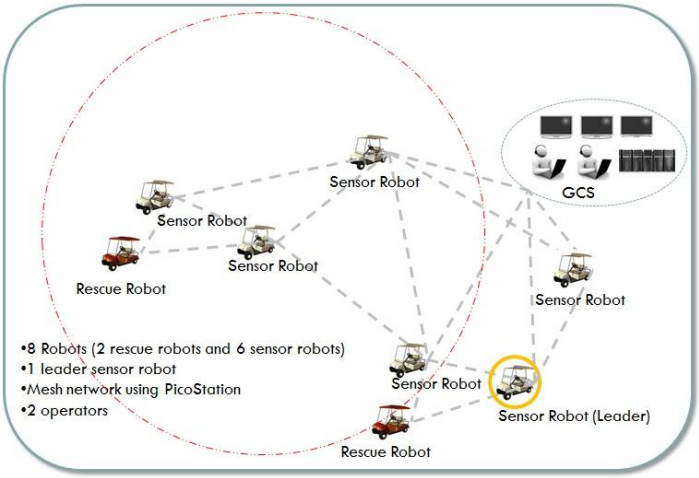
where $\mathbf{x}_k^{o_j} \in \mathcal{X}^{o_j}$ is the state of the object at time step k, $\mathbf{u}_k^{o_j} \in \mathcal{U}^{o_j}$ is the control inputs of the object, and $\mathbf{w}_k^{o_j} \in \mathcal{W}^{o_j}$ is the "system noise" of the object. In order for the formulation of the cooperative estimation and control problem, this moving object is searched and tracked by a group of sensor platforms $s = \{s_1, \cdots, s_{n_s}\}$. Assuming the global states of sensor platforms are assumed to be known, the motion model of sensor platform s_i is thus given by:

$$\mathbf{x}_k^{s_i} = \mathbf{f}^{s_i}\left(\mathbf{x}_{k-1}^{s_i}, \mathbf{u}_k^{s_i}\right) + \mathbf{w}_k^{s_i} \tag{6.2}$$

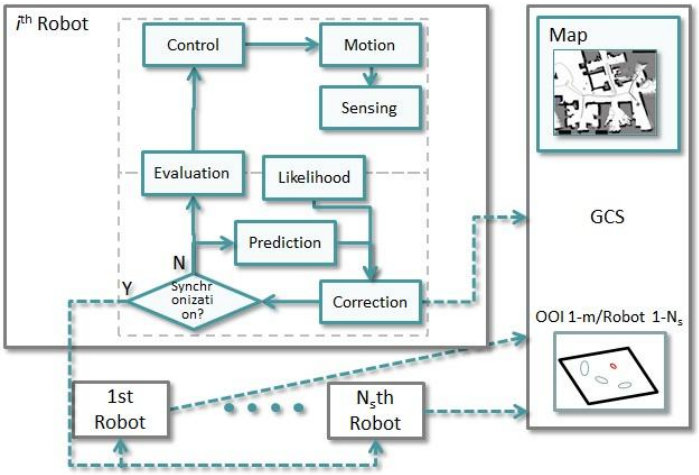
where $\mathbf{x}_k^{s_i} \in \mathcal{X}^{s_i}$ and $\mathbf{u}_k^{s_i} \in \mathcal{U}^{s_i}$ represent the state and control input of the ith vehicle, respectively, and $\mathbf{w}_k^{s_i} \in \mathcal{W}^{s_i}$ is the "system noise" of the sensor platform. The sensor platform also carries a sensor with an "observable region" as its physical limitation to observe an OOI. The observable region is determined not only by the properties of the sensor such as signal intensity but also the properties of the object such as the reflectivity. Defining the probability of detection (POD) $0 \leq P_D\left(\mathbf{x}_k^{o_j} | \mathbf{x}_k^{s_i}\right) \leq 1$ from these factors as a reliability measure for detecting the object o_j, the observable region can be expressed as ${}^{s_i}\mathcal{X}_O^{o_j} = \left\{\mathbf{x}_k^{o_j} | 0 < P_D\left(\mathbf{x}_k^{o_j} | \mathbf{x}_k^{s_i}\right) \leq 1\right\}$. Accordingly, the object state observed from the sensor platform, ${}^{s_i}\mathbf{z}_k^{o_j} \in \mathcal{X}^{o_j}$, is given by:

$${}^{s_i}\mathbf{z}_k^{o_j} = \begin{cases} {}^{s_i}\mathbf{h}^{o_j}\left(\mathbf{x}_k^{o_j}, \mathbf{x}_k^{s_i}\right) + {}^{s_i}\mathbf{v}_k^{o_j} & \mathbf{x}_k^o \in {}^{s_i}\mathcal{X}_O^o \\ \emptyset & \mathbf{x}_k^o \notin {}^{s_i}\mathcal{X}_O^o \end{cases} \tag{6.3}$$

where ${}^{s_i}\mathbf{v}_k^{o_j}$ represents the observation noise, and $\emptyset$ represents an "empty element", indicating that the observation contained no information on the object or that the object is unobservable when it is not within the observable region. Note here that the terms "sensor platform" and "robot" are used interchangeably as the configuration of sensors can be negligible.



(a)



(b)

Figure 35 Cooperative multi-robot system and overall process of cooperative search and tracking

6.2 Cooperative Search and Tracking

Figure 35 shows the concept of the fully-autonomous search and tracking strategy using multiple robots. In order for the multiple robots to become a team and to work cooperatively, they are linked to each other throughout the wireless connection. To enhance the efficiency of their cooperation there is a leader robot who is in charge of doing data fusion for the update of the belief. Since each robot is required to report its operation status with its own pose and map, it has to have a capability of estimating its pose and of creating the map. The base station collects information from each individual and combines it together and

sends back to robots as needed. When the robots explore the same environment search and tracking in the fully-autonomous strategy can be summarized as below:

Search

In general, the environment the robots explore is totally unknown which means there is no prior information such as the size and the shape of the environment. The environment can be roughly divided into several districts and separately explored by the robots, however, it can be said that area coverage is sometimes more reliable for search in unknown environments. When the robots cover the entire environment, they are guided by the frontier-based exploration to increase the efficiency of search.

Tracking

Once an OOI is detected, the robot keeps tracking it if the OOI is not stationary, so that every member as well as the base station can realize the existence of the OOI. The position of the OOI is estimated and corrected by the EKF. For with the jth object at time step k, the mean $\bar{\mathbf{x}}_{k|k-1}^{o_j}$ and the covariance $\mathbf{\Sigma}_{\bar{\mathbf{x}}_{k|k-1}^{o_j}}$ are predicted as

$$\begin{aligned} \bar{\mathbf{x}}_{k|k-1}^{o_j} &= \mathbf{f}^{o_j}\left(\mathbf{x}_{k-1|k-1}^{o_j}, \mathbf{u}_k^{o_j}\right) \\ \mathbf{\Sigma}_{\bar{\mathbf{x}}_{k|k-1}^{o_j}} &= \mathbf{A}_{k-1}\mathbf{\Sigma}_{\bar{\mathbf{x}}_{k|k-1}^{o_j}}\mathbf{A}_{k-1}{}^{t} + \mathbf{\Sigma}_{\mathbf{w}_{k-1}^{o_j}} \end{aligned} \qquad 6.4$$

where $\mathbf{A}_{k-1}$ is given by

$$\mathbf{A}_{k-1} = \frac{\partial \mathbf{f}^{o_j}\left(\mathbf{x}_{k-1|k-1}^{o_j}, \mathbf{u}_k^{o_j}\right)}{\partial \mathbf{x}_{k-1|k-1}^{o_j}} \qquad 6.5$$

Given the observation ${}^{s_i}\mathbf{z}_k^{o_j}$ and the sensor model $\mathbf{h}(\cdot)$, the mean $\bar{\mathbf{x}}_{k|k}^{o_j}$ and the covariance $\mathbf{\Sigma}_{\bar{\mathbf{x}}_{k|k}^{o_j}}$ are corrected as

$$\begin{aligned} \bar{\mathbf{x}}_{k|k}^{o_j} &= \bar{\mathbf{x}}_{k|k+1}^{o_j} + \mathbf{K}_k\left({}^{s_i}\mathbf{z}_k^{o_j} - {}^{s_i}\mathbf{h}^{o_j}\left(\bar{\mathbf{x}}_{k|k-1}^{o_j}, \mathbf{x}_k^{s_i}\right)\right) \\ \mathbf{\Sigma}_{\bar{\mathbf{x}}_{k|k}^{o_j}} &= (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\mathbf{\Sigma}_{\bar{\mathbf{x}}_{k|k-1}^{o_j}} \end{aligned} \qquad 6.6$$

where $\mathbf{K}_k$ is the Kalman gain which is given by

$$\mathbf{K}_k = \mathbf{\Sigma}_{\bar{\mathbf{x}}_{k|k-1}^{o_j}} \mathbf{C}_k{}^t \left(\mathbf{C}_k \mathbf{\Sigma}_{\bar{\mathbf{x}}_{k|k-1}^{o_j}} \mathbf{C}_k{}^t + \mathbf{v}_k^{o_j} \right)^{-1}$$
$$\mathbf{C}_k = \frac{\partial^{s_i}\mathbf{h}^{o_j}\left(\bar{\mathbf{x}}_{k|k+1}^{o_j}, \mathbf{x}_k^{s_i}\right)}{\partial \bar{\mathbf{x}}_{k|k+1}^{o_j}} \qquad 6.7$$

If the OOI is not moving object, there is no need for tracking.

6.3 Platform- and Hardware-in-the-loop Simulator

With the fully-autonomous search and tracking defined, this section focuses on the evaluation of the performance of the robots. In real environment, cooperation between the robots is very difficult to be evaluated. Moreover, since the purpose of the evaluation is not to identify the performance of real systems, but to validate the effectiveness of the proposed strategy, cooperative performance is evaluated within a virtual environment.

6.3.1 Concept and Design

The platform- and hardware-in-the-loop simulator (PHILS) was developed for the evaluation of cooperative performance of multiple autonomous robots and their testing in virtual environments. Figure 36 shows the schematic design of the PHILS. The PHILS consists of computers, monitors, a network switch that links the computers, and a server-client simulation software system installed on the computers. Out of the computers, three computers create an environment: one computer runs a server program so that client computers can share the same environment; another computer calculates motion of mobile objects, if there are any in the environment, using GPU since the motion of multiple objects can be calculated in parallel; the last computer with a GPU and a monitor acts as the environmental server and manages environmental parameters such as time, weather and communication speed whilst visualizing the behavior of all the autonomous robots in the environment under the support of GPU. The other computers are each equipped with a GPU and connected to a monitor, run a client visualizer using the GPU and view the

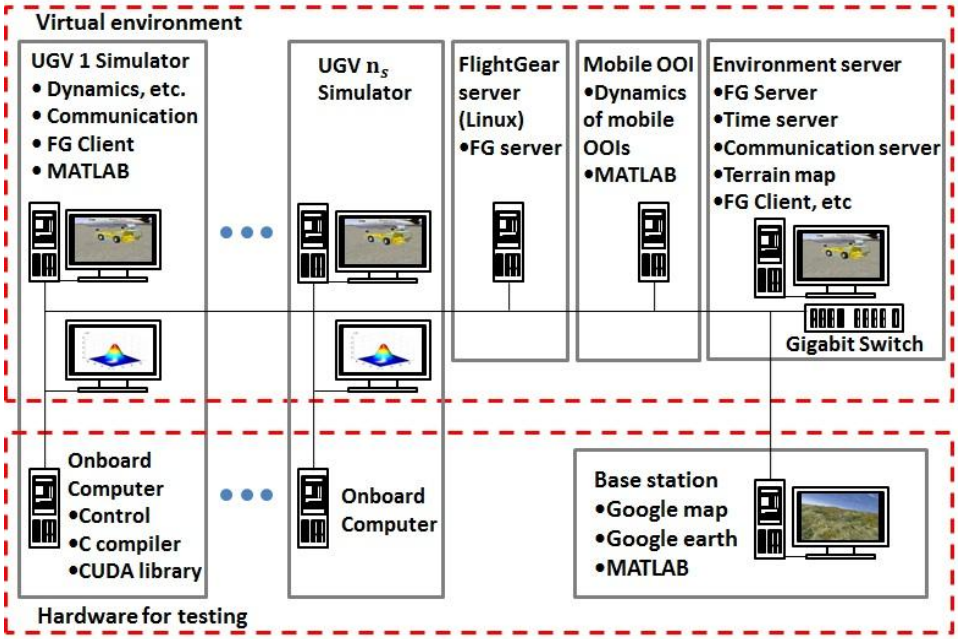


Figure 36. Concept of the platform- and hardware-in-the-loop simulator

environment with static and mobile objects as well as autonomous robots where one computer is allocated to each autonomous robot to calculate its motion and show its view.

The cooperative performance of multiple autonomous robots can be evaluated by linking computers each to be mounted on a robot to the network switch and by testing cooperative strategies such as cooperative mapping and exploration. The PHILS provides a monitor to each on-board computer, since the performance of the on-board computers, which we check at the base station, can be monitored simultaneously. The computer to be used as the base station can also be connected and tested in the virtual environment. Unlike the conventional hardware-in-the-loop simulators or multi-robot simulators, the primary advantage of the PHILS is that it can test cooperative autonomous robots and analyze their cooperative performance as well as hardware performance in a real-time virtual environment, enabling the implementation of synchronous and asynchronous communication strategies and the control of communication delay and loss.

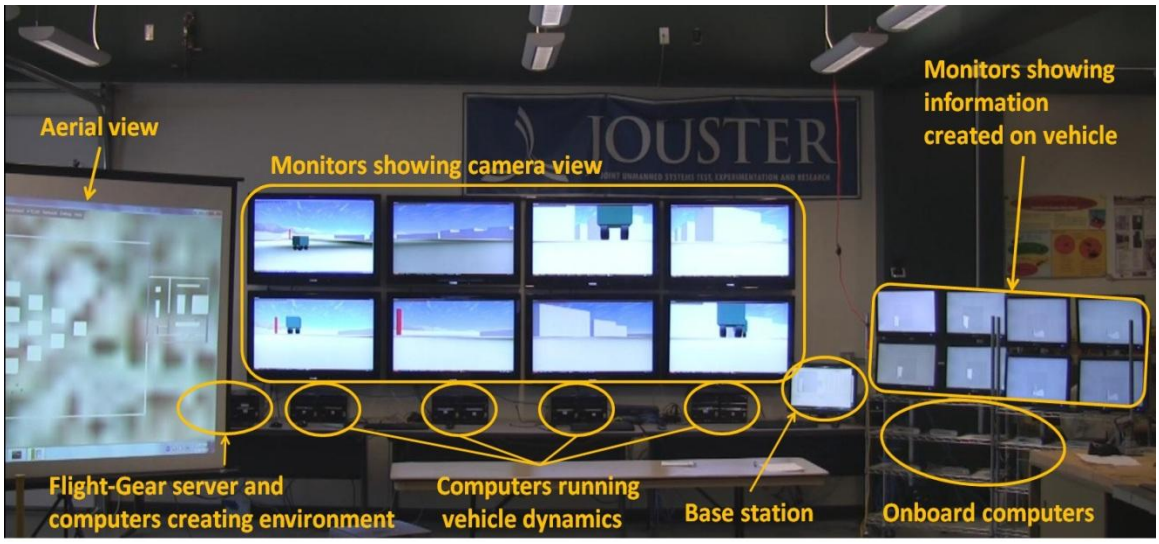


Figure 37. Developed platform- and hardwar-in-the-loop simulator

6.3.2 Development and Implementation

Figure 37 shows the PHILS developed by realizing the design whereas the detailed specifications of components of the developed PHILS are listed in Table 6. In the current setup, the PHILS has eight sets of computers meaning that it can accommodate the cooperation up to eight autonomous robots. The eight computers, as well as the other three computers that create an environment, are all those with the CPU of Dual Core 2.4GHz and the GPU of 32 stream processors. The eight monitors showing the views of autonomous robots are of 40 inch in size while the other eight monitors to connect to the on-board computers are of 19 inch in size. The network switch is of Gigabit speed so that the speed of wireless communication can be controlled with delay. The server, the client and the visualizer are all of Flight-Gear, which is an open-source simulator which was primarily designed for aerial vehicles but can now also incorporate ground vehicles. By accessing to the server, the FlightGear client can possess information on all the autonomous robots and mobile objects as well as the other environmental objects such as terrain and static objects and visualize them on the client computer. For the network communication, both the TCP/IP and the UDP are utilized.

Table 6. PHILS specifications

Type	Quantity	Specifications
Computer	11	Shuttle XPC
CPU	11	Intel Core2Duo, 2.4 GHz
Memory	11	3.25 GB RAM
GPU	10	NVidia GeForce 8400Gs
LCD monitor	8	Toshiba 40 inch
LCD monitor	8	Gateway 19 inch
LCD projector	1	Epson SVGA
Network switch	1	NETGEAR Gigabit 24 port
Visualizer	-	FlightGear
Server-client	-	FlightGear
Network protocol	-	TCP/IP, UDP

6.4 Performance Evaluation within the PHILS

Unlike real environments, environmental conditions in the PHILS are manageable, which enables the performance of the robots can be assessed under the designed conditions regardless of number of repetition. Additionally, by modeling motions of environments their true positions can be easily monitored and used to numerically evaluate performance of robots. Performances of a team of robots can be analyzed in several different ways depending on types of systems and objectives of evaluations. Considering a simple multi robot exploration scenario where there are OOIs at random positions, multiple robots can be numerically evaluated in terms of accuracy of mapping, efficiency of cooperative exploration, and accuracy of OOI localization.

The accuracy of mapping and OOI localization heavily relies on the solution to the problem of the pose estimation and map building of each robot, and it is given by the SLAM algorithm. Multi robot operation is not directly related to the problem of pose estimation and map building of each robot, however, the quality of the map of the whole environment can be associated with the cooperation of the robots. For analyzing the accuracy of the map, users of the PHILS can easily put recognizable landmarks on the virtual environment whose true locations are given only to users. While the robots cooperatively

explore the environment, the combined map can be created by the maps of individual robots in the same coordinate frame. Once the team of robots is finished with building the combined map, the quality of the map can be assessed by computing the average position errors between true positions of landmarks and mapped positions of landmarks:

$$e^{l} = \frac{1}{n_l}\sum_{i=1}^{n_l}\left\|\mathbf{x}_k^{l_i} - \hat{\mathbf{x}}_k^{l_i}\right\|^2 \qquad 6.8$$

where $\mathbf{x}_k^{l_i}$ and $\hat{\mathbf{x}}_k^{l_i}$ is true and mapped position of the ith landmark, respectively, and n_l is the number of landmarks. Similarly, the accuracy of OOI localization can be evaluated by the error between true and estimated positions of the OOIs.

In order to evaluate the efficiency of *n*-robot cooperation, let $t_{\beta,n}$ be the elapsed time for *n* robots to explore more than β % of the whole environment. $t_{\beta,n}$ may vary depending on the number of robots, thus how much the number of robots can affect the efficiency of the cooperation can be analyzed by changing the number of robots and comparing elapsed times for each case. When n_1 robots and n_2 robots explore the same area separately, time difference between these two cases is given by:

$$\Delta t_{\beta,n_2,n_1} = \frac{t_{\beta,n_2} - t_{\beta,n_1}}{n_2 - n_1} \qquad 6.9$$

where $n_2 > n_1$. A negative $\Delta t_{\beta,n_2,n_1}$ indicates that the total elapsed time for exploring β % of the whole environment is decreased. On the other hand, if $\Delta t_{\beta,n_2,n_1}$ is nonnegative, it is not worth adding more robots.

6.5 Experimental Results

This section investigates the effectiveness of the fully-autonomous search and tracking for USAR in two steps. The first step is aimed at the concept-proving of the fully-autonomous strategy by using two real robots. In the second step, a team of multiple robots (up to eight robots) are linked together within the

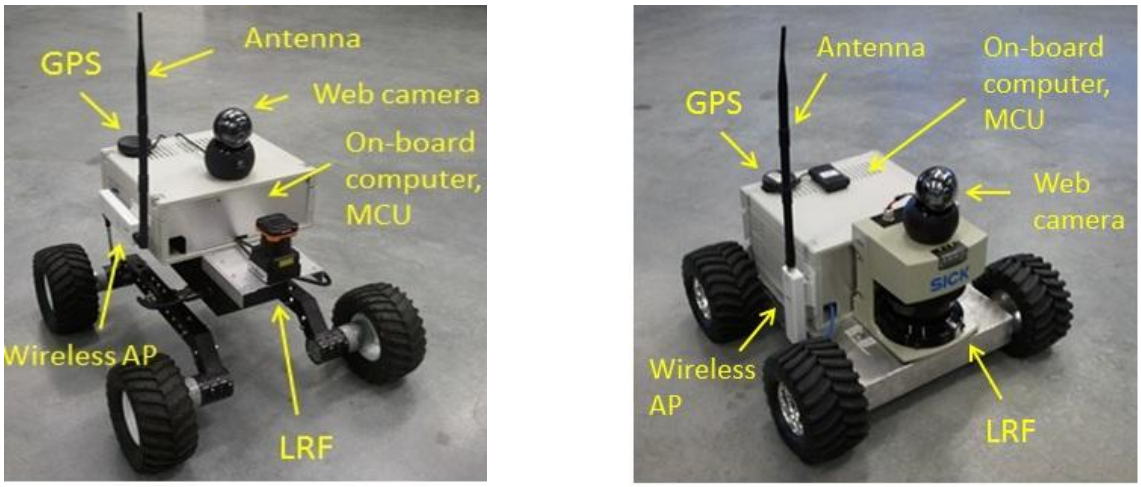


Figure 38. UGA 1 (left) and UGV 2 (right)

PHILS and cooperation between the robots based on the fully-autonomous search and tracking is evaluated to identify the applicability of the approach to USAR.

6.5.1 Validation

The fully-autonomous and cooperative search and tracking approach has been integrated into two real robots (Figure 38), and tested in an indoor environment $(20\,m\ \times 30\,m)$. During the experiment, the robots estimated its pose and created the map of the environment using the SLAM technique presented in Chapter 4. In the environment there were two static OOIs and the robots were asked to cooperatively find both of them while doing SLAM and sending the map to the base station. The OOIs are in yellow and their sizes are already known to the robots, so the robot can detect them based on a simple color detection algorithm. The mission is ended when the two robots find the OOIs are neutralize them by pointing them for several seconds, which can be regarded as rescue in USAR.

Figure 39 shows the screen shot of the GUI at the base station where information from each robot is combined together. As can be seen from the top left image showing the satellite map together with trajectories of the robots, the robots were initially located on the same position. The trajectories indicate that the frontier-based area coverage exploration leads the robots to explore two different areas resulting

Figure 39. Screen shot of the GUI at the base station

in the exploration becomes more efficient. In other words, the trajectories show that each robot considers not only the history of explored areas by itself but also the history of explored areas by the other robot at every moment it needs to make a decision on it waypoint. The top right image shows the combined map of the environment using two maps each created by one of the two robots which are displayed at the bottom left. The bottom right image shows one of the two robots (i.e. with its trajectory in blue line) detects and points out one of the OOIs. Conclusively, the experiment has successfully demonstrated its applicability to real systems and to real situations, although the mission given to the robots is not as complicated as missions in real USAR.

6.5.2 Performance Evaluation

A simplified multi-robot autonomous exploration scenario was considered within a 100 m $\times$ 100 m

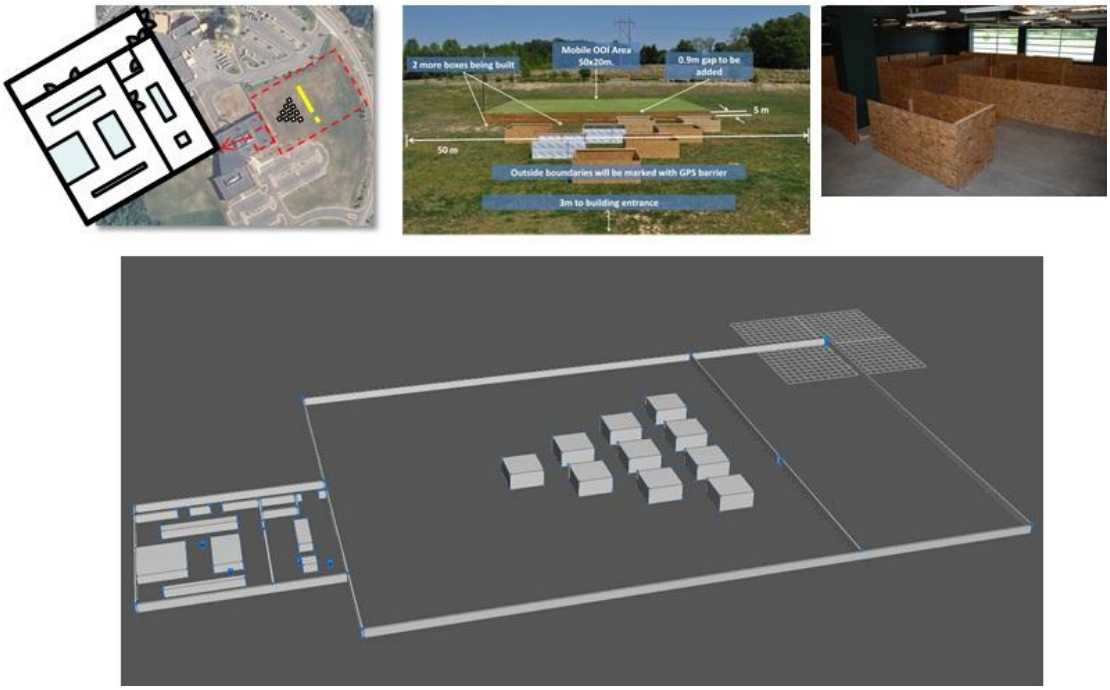


Figure 40. Virtual environment and corresponding real environment

unknown virtual environment which had both outdoor and indoor areas (Figure 40). There were 4 dynamic OOIs and 2 static OOIs in outdoor and indoor environment, respectively, and 4 up to 8 virtual robots operated at the same time. A team of robots was required to complete the mapping task and to neutralize all the OOIs. As discussed in Section 6.4, the cooperative performance of multiple robots was mainly evaluated based on the quality of mapping, the efficiency of the cooperative exploration, and the accuracy of OOI localization. Additionally, the robustness of the cooperative strategy applied to the team of the robots is evaluated throughout the success rate of the completion of the scenario. An OOI was regarded as being neutralized when two robots detected and successfully tracked the OOI for a few seconds keeping certain distance.

Figure 41 shows the average accumulated mapping error with respect to number of landmarks for three tests with different number of robots. Each test was conducted 50 times, and landmarks are assumed to be detected if they are within the field of view of a robot. As can be seen in the figure the average accumulated mapping error increases almost linearly, which means each mapped landmark is equally

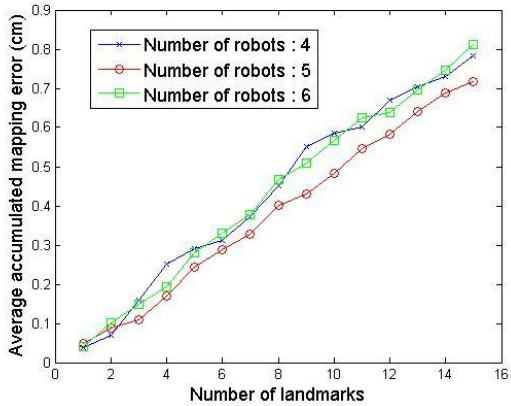
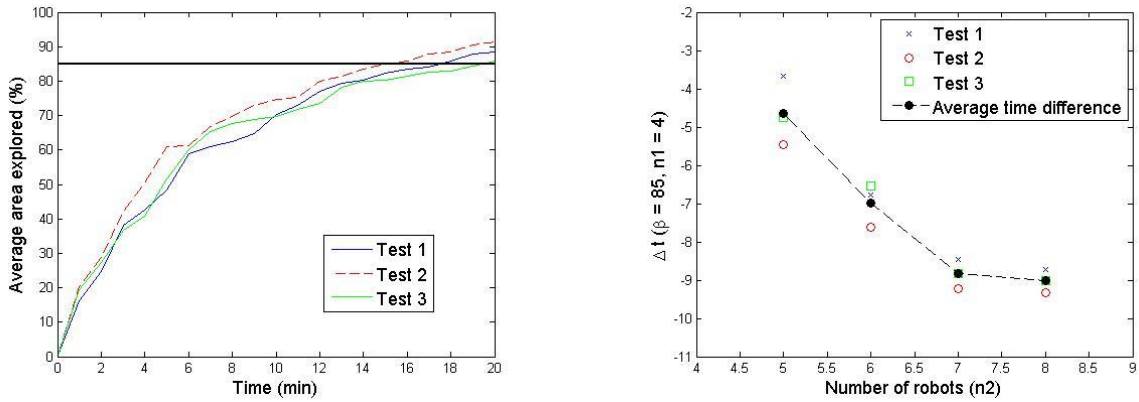


Figure 41. Average accumulated mapping error

away from the true position of the corresponding landmark. The average mapping error, defined as the average accumulated error divided by the number of landmarks, is within a few centimeters regardless of the number of robots indicating that the cooperative mapping is very accurate and the accuracy is not dependent on the complexity of the environment, and the virtual test environment used for performance evaluation is relatively well structured. Also, since these landmarks are not used for localization of the robot, the number of landmarks has no impact on the accuracy of the map.

Figure 42(a) shows the average percentage of the area explored out of the whole environment with respect to time when the number of robot is 4. The figure shows three different tests with different initial positions of the robots, and each test was conducted 50 times of experiments with the same initial positions of the robots. In this experiment, OOI neutralization was not included since time required to explore the environment could be significantly different depending on locations of OOIs. Although an increase of speed of each robot can accelerate exploration, it has nothing to do with efficiency of cooperative strategy. Thus, velocity of robots is set to 8 km/h for outdoor exploration and 4 km/h for the indoor exploration. The result shows that the virtual environment is entirely explored ($\beta = 85$) within 19 minutes on average, and that the cooperative exploration generally works fine since the initial positions of the robots do not play significant roles during the experiment. The simplicity of the environment also helps the result be consistent. As the number of robots increases, the efficiency of exploration also



(a) (b)

Figure 42. Average area explored vs. time and time difference vs. number of robots

improves as shown in Figure 42(b). It can be seen that the performance of cooperative exploration gradually increases until the number of robots reaches 7, but there is no big difference between using 7 and 8 robots.

The accuracy of position estimations of the OOIs is shown in Figure 43. Note that each test has the same initial positions of the OOIs, and was conducted 50 times, where OOI 1 to 5 are mobile OOIs and OOI 6 and 7 are static OOIs. From this figure variance from the true position and estimated position of OOIs is small. For localization of mobile OOIs, the error rate is slightly higher than that of static OOI since the level of uncertainty is relatively large. However, there is no single failure of mobile OOI neutralization since velocities of mobile OOIs are set to 3 km/h, which are slower than those of robots.

Last three experiments have shown the cooperative performance of a single mission. Figure 44, on the other hand, shows the success rate of completion of two missions at the same time to verify the robustness of the cooperative strategy applied to the team of the robots. Two missions include the neutralization of all the OOIs and the exploration of the entire environment ($\beta = 85$) within 20 minutes. When the number of robots is less than 7, the success rate stays within 30 % after 100 times of experiments each of which has different initial positions of robots and OOIs. However, the success rate

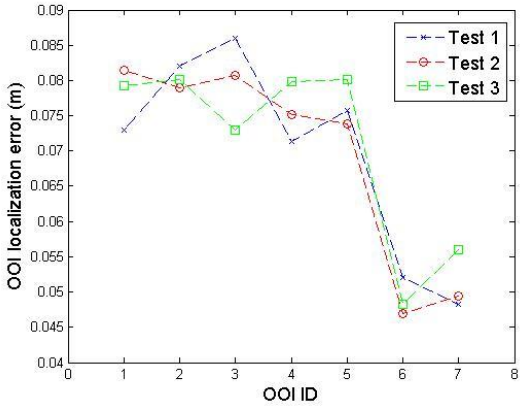


Figure 43. OOI localization error

suddenly rises when the number of robots is greater than or equal to 7. From the result it can be said that the cooperative algorithms implemented on each robot requires at least 7 robots to successfully explore the environment within 20 minutes.

6.6 Summary

This chapter has presented fully-autonomous search and tracking using a team of robots. As a member of the team, each robot shares information with other robots and the base station. For the enhancement of the efficiency in exploring an unknown environment, they are guided by frontier-based area coverage method. During its operation each robot detects and tracks OOIs similar to search and rescue in real USAR

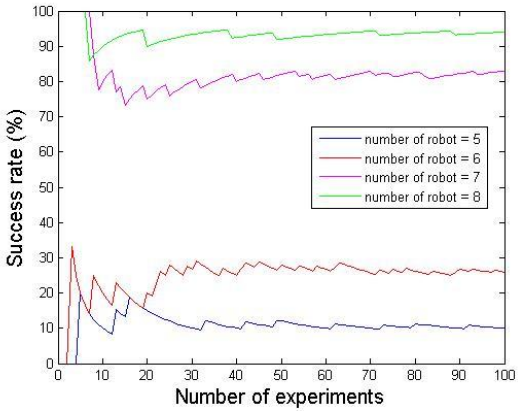


Figure 44 Success rate of completion of two missions

scenarios. The results showed the effectiveness of the concept of the fully-autonomous search and tracking. Further investigations on cooperation between robots show its applicability to real USAR. There still remains lots of work to be done to actually use the approach in real USAR. However, the fully-autonomous solution might become a promising solution to USAR in the sense that there are no control inputs from human operators.

Chapter 7

Conclusion and Future Work

This dissertation has attempted to provide two possible robotic solutions to USAR scenarios. The first solution is the map-based semi-autonomous robot navigation strategy via tele-operation, whereas the second solution is the fully-autonomous robotic search and tracking strategy using multiple mobile robots. As a core part of the solutions, this dissertation has proposed a technique, the so-called grid-based scan-to-map matching technique. This technique corrects the estimation error by matching every new scan to the globally defined grid map which maintains multiple NDs within each cell. The scan-to-map matching is achieved by the ND-to-ND matching after representing the new scan as NDs. In the map-based semi-autonomous robot navigation strategy, the human operator is given the map of the environment in which the robot is placed, and sends commands to the robot via tele-operation. In case of communication loss, the robot comes to the home position by inversely tracking previous waypoints. In the fully-autonomous robotic strategy, the environment is searched by multiple robots each of which has its own task and communicates with others during the operation. In the meantime, they send collected information to the base station.

The proposed scan-to-map matching technique was applied to three different experiments. The first experiment investigated the proposed scan-to-map matching technique in terms of the position and

orientation error while the robot moved and sequentially obtained new scans. The experiment showed that the accumulated orientation error was negligible and that the position error stayed within 4cm after travelling around 10m. The second experiment, investigating the effectiveness of maintaining multiple NDs within a cell, showed that the scans were matched better to the map with higher similarity when the cell had multiple NDs than one ND. The experiment also demonstrated the robust effect of the use of multiple NDs. Finally, the proposed technique was applied to the SLAM in three real environments to demonstrate its applicability to real problems. The resulting maps showed that the proposed technique without any post processes such as the loop closure generated position errors in the order of ten centimeters with very small orientation errors for the three environments after travelling around 200m with large orientation changes.

The scan-to-map matching technique also played an important role in the proposed semi-autonomous robot navigation using tele-operation. The semi-autonomous robot navigation strategy was integrated into a mobile robot and tested in artificial disaster areas. The experiment demonstrated the effectiveness of the map-based approach for the remote navigation and the ease of controlling the robot with the developed GUI. During the experiment it was found that the frontier-based guidance properly provided possible waypoints for the efficient exploration. Together with the accurate two-dimensional map, the three-dimensional map has significant potential in having much better understanding of the environment. When the wireless communication was manually discontinued, the robot successfully returned to the home position without any collision.

The proposed fully-autonomous robotics search and rescue was integrated into two real robots and tested for a multi-task scenario in a real environment. During the experiment the robots were successful in mapping and localizing OOIs, and the base station reliably collected information from the robots. The cooperative performance of the fully-autonomous strategy was further investigated within the developed PHILS using up to eight virtual robots. The results showed that the quality of the map was not influenced by the number of robots. On the other hand, the capability of exploring the entire environment was related

to the number of robots. In the virtual environment, elapsed time for exploring the environment reduced as the number of robots increased, but there was no big difference between the exploration by seven robots and the exploration by eight robots. When the proposed fully-autonomous approach was applied, another experimental result showed that at least seven robots were required to reliably complete the given multi-task mission in given environment.

This dissertation has focused on fundamental work for finding robotic solutions to USAR and much work is still left open to demonstrate its practical usefulness. Ongoing work primarily includes the extension of the proposed scan-to-map matching technique to the three-dimensional scan-to-map matching, since it is crucial to better deal with real environments. Further investigations on the capability of the scan-to-map matching technique in extremely noisy environments are also required so that the current formulation can be extended to effectively handle uncertainties from the environment. The map-based semi-autonomous robot navigation strategy can be improved by the enhancement of artificial intelligence, so that the robot can independently operate as needed. In consideration of the need for tele-operated robot capable of monitoring the environment, the strategy can be applied to solve different problems. As the future solution to USAR scenario the fully-autonomous strategy has a lot more future work which includes the development of enhanced cooperative algorithms in combination with the other works of the author's research group [98, 99].

References

[1] R. Smith and P. Cheesman, “On the representation of spatial uncertainty,” *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56-68, 1987.

[2] H.F. Durrant-Whyte, “Uncertain geometry in robotics,” *IEEE Trans. Robot. Automat.*, vol. 4, no. 1, pp. 23-31, 1988

[3] Jose A. Castellanos, Juan D. Tardos, Mobile Robot Localization and Map Building: A Multisensor Fusion Approach, Springer, 1999

[4] T. Bailey, Mobile Robot Localisation and Mapping in Extensive Outdoor Environments. PhD thesis, University of Sydney, Sydney, NSW, Australia, 2002.

[5] H. P. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 116–121, 1985.

[6] Moravec, H.P. and Blackwell, M. 1992. “Learning sensor models for evidence grids,” In *Robotics Institute Research Review*, Pittsburgh, PA.

[7] Elfes, A., Occupancy grids: A stochastic spatial representation for active robot perception. In *Sixth Conference on Uncertainty in AI*, 1990.

[8] Elfes, A., Dynamic control of robot perception using multiproperty inference grids. In *International Conference on Robotics and Automation*, 1992.

[9] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, pages 46-57, June 1989.

[10] A.C. Schultz, W. Adams, B. Yamauchi, and M. Jones. Unifying exploration, localization, navigation and planning through a common representation. In *IEEE International Conference on Robotics and Automation*, pages 2651–2658, 1999.

[11] Thomas Bräunl, Nicholas Tay, “Combining configuration space and occupancy grid for robot navigation”, *Industrial Robot: An International Journal*, Vol. 28 Iss: 3, pp.233 – 24, 2001

[12] F. Wallner, T. C. Lueth, and F. Langinieux. "Fast local path planning for a mobile robot", In *ICARCV'92 Second International Conference on Automation, Robotics and Computer Vision*, September 1992.

[13] J. Borenstein and Y. Kore, "Real-time obstacle avoidance for fast mobile robots in cluttered environments", In *IEEE International Conference on Robotics and Automation*, pages 572-577, 1990.

[14] S. Thrun., Learning occupancy grids with forward models, In *Proceedings of the Conference on Intelligent Robots and Systems (IROS2001)*, Hawaii, 2001.

[15] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation, *Artificial Intelligence*, 99(1):21–71, 1998.

[16] D. Murray and J.J. Little, "Using Real-Time Stereo Vision for Mobile Robot Navigation", presented at *Auton. Robots*, pp.161-171, 2000

[17] H.P. Moravec and M.C. Martin. Robot navigation by 3D spatial evidence grids. Technical Report, Mobile Robots Laboratory, Robotics Institute, Carnegie Mellon University, 1994.

[18] Kaustubh Pathak, Andreas Birk, Jann Poppinga, and Sören Schwertfeger, 3D Forward Sensor Modeling and Application to Occupancy Grid based Sensor Fusion, *International Conference on Intelligent Robots and Systems (IROS)*, 2007

[19] M. Montemerlo and S. Thrun. A multi-resolution pyramid for outdoor robot terrain perception. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2004.

[20] Daniel Meyer-Delius, Maximilian Beinhofer and Wolfram Burgard, “Occupancy Grid Models for Robot Mapping in Changing Environments”, In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012

[21] Rybski, P.E., Zacharias, F., Lett, J.-F., “Using visual features to build topological maps of indoor environments,” Proceedings of *the 2003 IEEE International Conference on Robotics & Automation*, Taiwan, September 14-19, 2003

[22] M. Liu, D. Scaramuzza, C. Pradalier, R. Siegwart and Q. Chen , “Scene Recognition with Omnidirectional Vision for Topological Map using Lightweight Adaptive Descriptors,” *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 11-15, 2009 St. Louis, USA

[23] Hironobu Sasaki, Naoyuki Kubota, Kazuhiko Taniguchi, Growing Topological Map for SLAM of Mobile Robots, *SICE Annual Conference*, August, 2008

[24] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[25] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Fourth International Symposium of Robotics Research*, pages 467–474, 1987.

[26] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[27] T. Bar-Shalom and T.E. Fortmann, Tracking and Data Association, Academic Press In., Boston, MA, 1988.

[28] H.J.S. Feder, J.J. Leonard, and C.M. Smith, "Adaptive mobile robot navigation and mapping," *Int. J. Robotics Research*, vol. 18, no. 7, pp. 650-668, 1999.

[29] P.M. Newman, On The Structure and Solution of the Simultaneous Localisation and Map Building Problem, Ph.D. thesis, Dept. Mechanical and Mechatronic Engineering, University of Sydney, 1999

[30] Rencken, W., Feiten, W., and Zöllner, R. (1998). Relocalisation by partial map matching. *Lecture Notes in Computer Science*, volume 1724, pp. 21–35.

[31] Bosse, M. C. (2004). *ATLAS: A Framework for Large Scale Automated Mapping and Localization*. *Ph.D. Thesis*, Massachusetts Institute of Technology.

[32] J. Neira and J.D. Tardos, Data association in stochastic mapping using the joint compatibility test, *IEEE Transactions on Robotics and Automation*, 17(6):890-897, 2001

[33] Neira, J., Tardós, J. D., and Castellanos, J. A. (2003). Linear time vehicle relocation in SLAM. *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, pp. 427–433.

[34] M. Montemerlo, S. Thrun D. Koller, and B. Wegbreit, FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, *In Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1151-1156, Acapulco, Mexico, 2003.

[35] Juan Nieto, Jose Guivant, Eduardo Nebot, "Real Time Data Association for FastSLAM", In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, page 412-418, 2003

[36] Sebastian Thrun, et al., "FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *Journal of Machine Learning Research*, 2004

[37] Javier Civera, et al., 1-Point RANSAC for EKF-Based Structure from Motion, *2009 IEEERSJ International Conference on Intelligent Robots and Systems*, 2009.

[38] Choi, Y., Lee, T., Oh, S., "A line feature based SLAM with low grade range sensors using geometric constraints and active exploration," *Autonomous Robots*, vol. 24, 2008.

[39] Geovany Araujo Borges and Marie-Jose Aldon, Line extraction in 2D range images for mobile robotics, *Journal of Intelligent and Robotics Systems*, 40: 267-297, 2004

[40] Jochen Schmidt, Chee K.Wong, andWai K. Yeap, A split & merge approach to metric-topological map-building, *International Conference on Pattern Recognition (ICPR)*, vol. 3: 1069-1072, 2006.

[41] F. Lu and E.Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18:249.275, 1997.

[42] P. Biber and W. Straßer, The normal distributions transform: A new approach to laser scan matching, In *IEEE Int. Conf. on Intelligent Robots and Systems*, Las Vegas, USA, 2003.

[43] A. Nüchter, J. Elseberg, P. Schneider, D. Paulus, Linearization of Rotations for Globally Consistent n-Scan Matching, 2010 *IEEE International Conference on Robotics and Automation (ICRA)*, 1373-1379, 2010

[44] J. L. Martinez, J. Morales, A. Mandow, A. Garcia-Cerezo, Incremental Closed-Form Solution to Globally Consistent 2D Range Scan Mapping with Two-Step Pose Estimation, *The 11th IEEE International Workshop on Advanced Motion Control*, March 21-24, 2010, Nagaoka, Japan

[45] S. Thrun, W. Bugard, D. Fox, A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping, In *proc. of the International Conference on Robotics and Automation*, 2000

[46] M. Bosse, P. Newman, J. J. Leonard and S. Teller. Slam in large-scale cyclic environments using the atlas framework. *International Journal of Robotics Research*, 23(12):1113-1139, 2004.

[47] Bailey, T., Nieto, J., Guivant, J., Stevens, M., Nebot, E., Consistency of the EKF SLAM Algorithm, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006

[48] Jan Weingarten and Roland Siegwart, "EKF-based 3D SLAM for Structured Environment Reconstruction", In *Proceedings of Int. Conf. on Intelligent Robots and Systems (IROS)*, Edmonton, Canada, August 2-6, 2005.

[49] Andrew J. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera," *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)*, 2003

[50] S. Ahn, M. Choi, J. Choi and W. K. Chung, "Data Association Using Visual Object Recognition for EKF-SLAM in Home Environment," *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China

[51] S. Huang and G. Dissanayake. Convergence analysis for extended Kalman filter based SLAM. In *IEEE International Conference on Robotics and Automation*, 2006.

[52] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. of the American Control Conference*, pages 1628–1632, Seattle, WA, USA, 1995.

[53] J.A. Castellanos, R. Martinez-Cantin, J.D. Tardos, J. Neira , Robocentric map joining: Improving the consistency of EKF-SLAM, *Robotics and Autonomous Systems* (2006)

[54] S. J. Julier and J. K. Uhlmann. Building a Million Beacon Map. In *Sensor Fusion*. SPIE, 2001.

[55] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL. 29, NO. 6, JUNE 2007

[56] Henning Lategahn, Andreas Geiger and Bernd Kitt, “Visual SLAM for Autonomous Ground Vehicles,” *2011IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1732-1737, 2011

[57] A. Doucet, J.F.G. de Freitas, K. Murphy, and S. Russel. Rao-Blackwellized partcile filtering for dynamic bayesian networks. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 176–183, Stanford, CA, USA, 2000.

[58] K. Murphy. Bayesian map learning in dynamic environments. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, pages 1015–1021, Denver, CO, USA, 1999.

[59] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods In Practice*. Springer, 2001

[60] Cyrill Stachniss, Giorgio Grisetti and Wolfram Burgard, “Recovering Particle Diversity in a Rao-Blackwellized Particle Filter for SLAM After Actively Closing Loops,” In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 667-672, Barcelona, Spain, 2005.

[61] A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultainous localization and mapping without predetermined landmarks. In *Proc. of theInt. Conf. on Artificial Intelligence (IJCAI)*, pages 1135–1142, Acapulco, Mexico, 2003.

[62] D. Hähnel, W. Burgard, D. Fox, and S. Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 206–211, Las Vegas, NV, USA, 2003.

[63] Andrew Howard. Multi-robot simultaneous localization and mapping using particle filters. In *Robotics: Science and Systems*, pages 201–208, Cambridge, MA, USA, 2005.

[64] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard, “Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 36-46, 2007

[65] Pantelis Elinas, Robert Sim, James J. Little, *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, Florida - May 2006

[66] Hu, W. *et al.* A Modified Particle Filter for Simultaneous Robot Localization and Landmark Tracking in an Indoor Environment, *The 2004 Australasian Conference on Robotics and Automation*, Canberra, 6-8, December 2004

[67] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Journal of Autonomous Robots*, 4:333–349, 1997

[68] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. of the IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 318–325, Monterey, CA, USA, 1999.

[69] D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun. Towards lazy data association in slam. In *Proc. of the Int. Symposium of Robotics Research (ISRR)*, pages 421–431, Siena, Italy, 2003

[70] E. Olson, J.J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proc. of the IEEE Int. Conf. on Robotics & Automation*, pages 2262–2269, 2006.

[71] Giorgio Grisetti, Cyrill Stachniss, Slawomir Grzonka, and Wolfram Burgard, A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent. *Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, 2007.

[72] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Fast incremental smoothing and mapping with efficient data association," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2007.

[73] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Transactions on Robotics*, 21(2):1–12, 2005.

[74] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[75] Z. Zhang. "Iterative Point Matching for Registration of Free-Form Curves and surfaces," *International Journal of Computer Vision*, Vol. 13, Issue 2, Oct., 1994

[76] Andreas N¨uchter, Kai Lingemann, and Joachim Hertzberg, Cached k-d tree search for ICP algorithms, *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM)*, 2007

[77] D.M. Mount. ANN programming manual. Technical report, University of Maryland, Department of Computer Science, 1998.

[78] Feng Lu, Milios, E.E., "Robot pose estimation in unknown environments by matching 2D range scans," Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., *1994 IEEE Computer Society Conference on*, vol., no., pp.935-938, 21-23 Jun 1994

[79] J. Minguez, L. Montesano, and F. Lamiraux. Metric-based iterative closest point scan matching for sensor displacement estimation. *IEEE Transactions on Robotics*, 22(5):1047–1054, October 2006.

[80]A. Censi, "An icp variant using a point-to-line metric," in *Proceedings of the IEEE International Conference on Robtics and Automation (ICRA)*, 2008

[81] Tahir Yaqub, Michael J Tordon and Jayantha Katupitiya, Line Segment Based Scan Matching for Con-current Mapping and Localization of a Mobile Robot, *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2006

[82] Y. Chen, G. Medioni. "Object Modeling by Registration of Multiple Range Images," *Proc. of the 1992 IEEE Intl. Conf. on Robotics and Automation*, pp. 2724-2729, 1991.

[83] Tomohito Takubo *et al.*, NDT Scan Matching Method for High Resolution Grid Map, In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2009

[84] Eijiro Takeuchi and Takashi Tsubouchi, A 3-D Scan Matching using Improved 3-D Normal Distributions Transform for Mobile Robotic Mapping, *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct, 2006

[85] Ripperda, N., Brenner, C. (2005): Marker-Free Registration of Terrestrial Laser Scans Using the Normal Distribution Transform, In *Proceedings of the ISPRS Working Group V/4 Workshop 3D-ARCH* 2005

[86] S.T. Pfister, K.L. Kreichbaum, S.I. Roumeliotis, and J.W. Burdick. Weighted range sensor matching algorithms for mobile robot displacement estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1667–1674, May 2002.

[87] L. Montesano, J. Minguez, and L. Montano. Probabilistic scan matching for motion estimation in unstructured environments. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, pages 1445–1450, August 2005.

[88] K. S. Arun, T.S. Huang, S.D. Blostein, Least-Squares Fitting of Two 3-D Point Sets, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 698-700, 1987.

[89] Corliss, W. R. *Teleoperators: Man's Machine Partners.*, United States Atomic Energy Commission. Office of Information Services, 1972.

[90] Zingg, S., Scaramuzza, D., Weiss, S., and Siegwart, R. MAV Navigation through Indoor Corridors Using Optical Flow, *IEEE International Conference on Robotics and Automation (ICRA2010)*, Anchorage, Alaska, May, 2010.

[91] Nefian, A.V., Bradski, G.R. "Detection of Drivable Corridors for Off-Road Autonomous Navigation". *ICIP-06: Proceedings of the IEEE International Conference on Image Processing*. pp. 3025-3028.

[92] Gaspar, J., Winters, N., and Santos-Victor, J. Vision-based Navigation and Environmental Representations with an Omni-directional Camera. *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, Dec, 2000.

[93] Scaramuzza, D., Siegwart, R. Appearance Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles. *IEEE Transactions on Robotics*, vol. 24, issue 5, October 2008.

[94] Ohya, A., Kosaka, A., and Kak, A. Vision-Based Navigation by a Mobile Robot with Obstacle Avoidance Using Single-Camera Vision and Ultrasonic Sensing. *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, Dec, 1998.

[95] Urmson, Chris et al. (2008). "Autonomous driving in urban environments: Boss and the Urban Challenge," *Journal of Field Robotics*, Vol. 25, Special Issue on the 2007 DARPA Urban Challenge, Part *I*. Pages 425-466. 2008.

[96] Tsubouchi, T., Tanaka, A., Ishioka, A., Tomono, M., and Yuta S. A SLAM Based Teleoperation and Interface System for Indoor Environment Reconnaissance in Rescue Activities. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[97] Yamauchi, B. A frontier-based approach for autonomous exploration. In Proceedings of the 1997 *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 146-151, Monterey, CA, July, 1997.

[98] Tomonari Furukawa, Lin Chi Mak, Kunjin Ryu, Xianqiao Tong and Gamini Dissanayake, “Bayesian Search, Tracking, Localization and Mapping: A Unified Strategy for Multi-task Mission,” *INFORMS 2011 Annual Meeting*, November 13-16, 2011, Charlotte, 2011

[99] Furukawa, T., Durrant-Whyte, H.F., Lavis, B., The element-based method - theory and its application to bayesian search and tracking -, *The 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

Appendix A

Extended Kalman Filter for Simultaneous Localization and Mapping

Let the robot motion and the observation of the environment be $\mathbf{x}_k^r = [x_k^r, y_k^r, \theta_k^r]^t$ and $\mathbf{z}_k = \left[\mathbf{z}_k^1, \mathbf{z}_k^2, \cdots, \mathbf{z}_k^n\right]$, where $\mathbf{z}_k^i$ is the observation of the ith landmark at time step k. In the EKF SLAM the robot motion and the observation are described in the forms shown below:

$$\begin{aligned} \mathbf{x}_k^r &= \mathbf{f}^r(\mathbf{x}_{k-1}^r, \mathbf{u}_k^r) + \mathbf{w}_k^r \\ \mathbf{z}_k &= \mathbf{h}^m(\mathbf{x}_k^r, \mathbf{x}^m) + \mathbf{v}_k^m \end{aligned} \qquad \text{A1.1}$$

where $\mathbf{f}^r(\cdot)$ and $\mathbf{h}^m(\cdot)$ describe the motion of the robot and the location of the observation, and $\mathbf{x}^m = \left[\left[x^{m_1}, y^{m_1}\right]^t, \cdots, \left[y^{m_\eta}, y^{m_\eta}\right]^t\right]$ is the feature map containing the location of each ith landmark, $\left[x^{m_i}, y^{m_i}\right]^t$**,** which is time invariant (i.e. static). $\mathbf{u}_k^r$ is the control input at time step k, $\mathbf{w}_k^r$ and $\mathbf{v}_k^m$ denote Gaussian noise of the robot motion and the observation errors with the covariance $\mathbf{Q}_k$ and $\boldsymbol{Q}_k$, respectively.

The robot state, $\mathbf{x}_k^r$, is then represented by the position and the orientation with mean, $\bar{\mathbf{x}}_k^r$, and covariance, $\mathbf{P}_{k|k}^r$:

$$\bar{\mathbf{x}}_k^r = [\bar{x}_k^r, \bar{y}_k^r, \bar{\theta}_k^r]^t$$

$$\mathbf{P}_{k|k}^r = \begin{bmatrix} \sigma_{x^r x^r}^2 & \sigma_{x^r y^r}^2 & \sigma_{x^r \theta^r}^2 \\ \sigma_{x^r y^r}^2 & \sigma_{y^r y^r}^2 & \sigma_{y^r \theta^r}^2 \\ \sigma_{x^r \theta^r}^2 & \sigma_{y^r \theta^r}^2 & \sigma_{\theta^r \theta^r}^2 \end{bmatrix}_{k|k} \qquad \text{A1.2}$$

The map is also represented by the set of locations of all landmarks with the mean of these landmarks, $\bar{\mathbf{x}}_k^m$, and the covariance matrix, $\mathbf{P}_{k|k}^m$:

$$\bar{\mathbf{x}}_k^m = \left[\left[\bar{x}_k^{m_1}, \bar{y}_k^{m_1}\right]^t, \ \cdots, \ \left[\bar{x}_k^{m_\eta}, \bar{y}_k^{m_\eta}\right]^t \right]$$

$$\mathbf{P}_{k|k}^m = \begin{bmatrix} \sigma_{x^1 x^1}^2 & \sigma_{x^1 y^1}^2 & \cdots & \sigma_{x^1 x^\eta}^2 & \sigma_{x^1 y^\eta}^2 \\ \sigma_{x^1 y^1}^2 & \sigma_{y^1 y^1}^2 & \cdots & \sigma_{y^1 x^\eta}^2 & \sigma_{y^1 y^\eta}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_{x^1 x^\eta}^2 & \sigma_{y^1 x^\eta}^2 & \cdots & \sigma_{x^\eta x^\eta}^2 & \sigma_{x^\eta y^\eta}^2 \\ \sigma_{x^1 y^\eta}^2 & \sigma_{y^1 y^\eta}^2 & \cdots & \sigma_{x^\eta y^\eta}^2 & \sigma_{y^\eta y^\eta}^2 \end{bmatrix}_{k|k} \qquad \text{A1.3}$$

In probabilistic term, SLAM is the problem of determining $P(\mathbf{x}_k^r, \mathbf{x}^m | \, \mathbf{Z}_{1:k}^m, \mathbf{U}_{1:k}, \ \mathbf{x}_1^r)$, where $\mathbf{Z}_{1:k}^m = \left[\mathbf{Z}_{1:k-1}^m, \ \mathbf{z}_k \right]$ is the set of all observation and $\mathbf{U}_{1:k} = [\mathbf{U}_{1:k-1}, \ \mathbf{u}_k]$ is the history of the control inputs. The mean and the covariance of this joint posterior distribution can be formulated as

$$\begin{bmatrix} \bar{\mathbf{x}}_{k|k}^r \\ \bar{\mathbf{x}}_k^m \end{bmatrix} = \mathrm{E}\begin{bmatrix} \mathbf{x}_k^r \\ \mathbf{x}^m \end{bmatrix} | \, \mathbf{Z}_{1:k}^m \Bigg]$$

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{k|k}^r & \mathbf{P}_{k|k}^{rm} \\ \left[\mathbf{P}_{k|k}^{rm}\right]^t & \mathbf{P}_{k|k}^m \end{bmatrix} = \mathrm{E}\left[\begin{pmatrix} \mathbf{x}_k^r - \bar{\mathbf{x}}_k^r \\ \mathbf{x}^m - \bar{\mathbf{x}}_k^m \end{pmatrix} \begin{pmatrix} \mathbf{x}_k^r - \bar{\mathbf{x}}_k^r \\ \mathbf{x}^m - \bar{\mathbf{x}}_k^m \end{pmatrix}^t | \, \mathbf{Z}_{1:k}^m \right] \qquad \text{A1.4}$$

where the mean and the covariance are computed by applying the EKF method:

<u>Prediction</u>

In the prediction process, or time update process, the mean and the covariance are updated by time. Since

the map is static, the mean and the covariance of the map are not predicted from the previous time step. While, the mean and the covariance of the robot pose are predicted as

$$\bar{\mathbf{x}}_{k|k-1}^{r} = \mathbf{f}\left(\bar{\mathbf{x}}_{k-1|k-1}^{r}, \mathbf{u}_k\right)$$
$$\mathbf{P}_{k|k-1}^{r} = \partial\mathbf{f}\mathbf{P}_{k-1|k-1}^{r}\,\partial\mathbf{f}^{t} + \mathbf{Q}_k \qquad \text{A1.5}$$

where $\partial\mathbf{f}$ is the Jacobian of $\mathbf{f}$.

<u>Correction</u>

In the correction process, or observation update process, the mean and the covariance are updated by the observation. Throughout the correction, the robot pose as well as the map is updated:

$$\begin{bmatrix} \bar{\mathbf{x}}_{k|k}^{r} \\ \bar{\mathbf{x}}_{k}^{m} \end{bmatrix} = \left[\bar{\mathbf{x}}_{k|k-1}^{r}\bar{\mathbf{x}}_{k-1}^{m}\right] + \mathbf{W}_k\left[\mathbf{z}_k - \mathbf{h}\left(\bar{\mathbf{x}}_{k|k-1}^{r}, \bar{\mathbf{x}}_{k-1}^{m}\right)\right]$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{W}_k\mathbf{S}_k\mathbf{W}_k^{t} \qquad \text{A1.6}$$

where $\mathbf{S}_k$ and $\mathbf{W}_k$ are defined as

$$\mathbf{S}_k = \partial\mathbf{h}\mathbf{P}_{k|k-1}\,\partial\mathbf{h}^{t} + \boldsymbol{Q}_k$$
$$\mathbf{W}_k = \mathbf{P}_{k|k-1}\,\partial\mathbf{h}^{t}\mathbf{S}_k^{-1} \qquad \text{A1.7}$$

where $\partial\mathbf{h}$ is the Jacobian of $\mathbf{h}$.

Appendix B

Denavit-Hartenberg Convention

Let x_i, y_i, and z_i be the x-, y-, and z-axis of the ith frame with the origin $\mathcal{O}_i$. Deriving forward kinematics from the base frame to the frame of end effector using the D-H convention is as follows:

Step 1: Locate and label the joint axis

Step 2: Establish the base frame. Set the origin anywhere on the z_0-axis. The x_0 and y_0 axes are chosen conveniently to form a right-hand frame.

For $i = 1, \cdots, n-1$, perform Steps 3 to 5.

Step 3: Locate the origin $\mathcal{O}_i$ where the common normal to z_i and z_{i-1} intersects z_i. If z_i intersects z_{i-1} locate $\mathcal{O}_i$ at this intersection. If z_i and z_{i-1} are parallel, locate $\mathcal{O}_i$ in any convenient position along z_i.

Step 4: Establish x_i along the common normal between z_{i-1} and z_i through $\mathcal{O}_i$, or in the direction normal to the $z_{i-1} - z_i$ plane if z_{i-1} and z_i intersect.

Step 5: Establish y_i to complete a right-hand frame.

Step 6: Establish the end-effector frame $o_n x_n y_n z_n$. Assuming the nth joint is revolute, set $z_n = \boldsymbol{a}$ along the direction z_{i-1}. Establish the origin $\mathcal{O}_n$ conveniently along z_n, preferably at the center of the gripper or at the tip of any tool that the manipulator may be carrying. Set $y_n = \boldsymbol{s}$ in the direction of the gripper

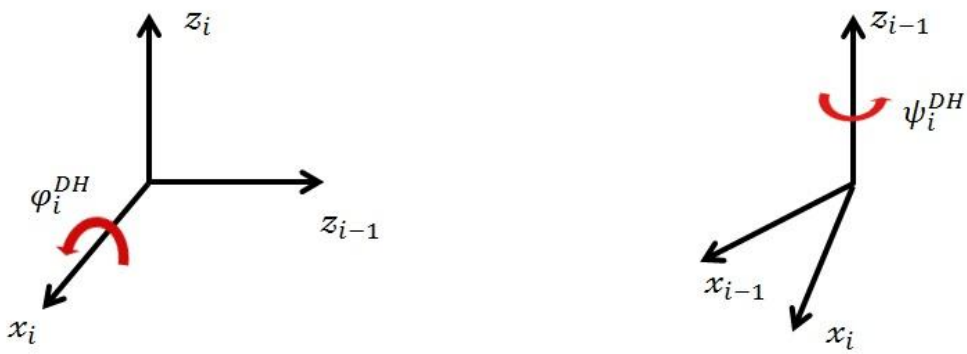


Figure 45. Positive sense for link parameters

closure and set $x_n = \boldsymbol{n}$ as $\boldsymbol{s} \times \boldsymbol{a}$. If the tool is not a simple gripper set x_n and y_n conveniently to form a right-hand frame.

Step 7: Create a table link parameters a_i^{DH} , $d_i^{DH}{}_i$, φ_i^{DH}, ψ_i^{DH} (Figure 45).

a_i^{DH}: Distance along x_i from $\mathcal{O}_i$ to the intersection of the x_i and z_{i-1} axes.

d_i^{DH}: Distance along z_{i-1} from $\mathcal{O}_{i-1}$ to the intersection of the x_i and z_{i-1} axes. d_i is variable if joint i is prismatic.

φ_i^{DH}: Angle between z_{i-1} and z_i measured about x_i

ψ_i^{DH}: Angle between x_{i-1} and x_i measured about z_{i-1}. θ_i is variable if joint i is revolute.

Step 8: Form the homogeneous transformation matrices $\mathbf{A}_i$:

$$\mathbf{A}_i = \begin{bmatrix} c_{\psi_i^{DH}} & -s_{\psi_i^{DH}} & 0 & 0 \\ s_{\psi_i^{DH}} & c_{\psi_i^{DH}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i^{DH} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i^{DH} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\varphi_i^{DH}} & -s_{\varphi_i^{DH}} & 0 \\ 0 & s_{\varphi_i^{DH}} & c_{\varphi_i^{DH}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{A2.1}$$

$$= \begin{bmatrix} c_{\psi_i^{DH}} & -s_{\psi_i^{DH}}\, c_{\varphi_i^{DH}} & s_{\psi_i^{DH}}\, s_{\varphi_i^{DH}} & a_i^{DH}\, c_{\psi_i^{DH}} \\ s_{\psi_i^{DH}} & c_{\psi_i^{DH}}\, c_{\varphi_i^{DH}} & -c_{\psi_i^{DH}}\, s_{\varphi_i^{DH}} & a_i^{DH}\, s_{\psi_i^{DH}} \\ 0 & s_{\varphi_i^{DH}} & c_{\varphi_i^{DH}} & d_i^{DH} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $c_{\psi_i^{DH}}$ and $s_{\psi_i^{DH}}$ represent $\cos\psi_i^{DH}$ and $\sin\psi_i^{DH}$, respectively.

Step 9: Form the transformation matrix, ${}^{0}_{n}\mathbf{A} = \mathbf{A}_1 \cdots \mathbf{A}_n$, that expresses the position and orientation of points in the nth frame with respect to the base frame