A framework of probabilistic 3D map fusion for collaborative robots

Yue, Yufeng

2019

Yue, Y. (2019). A framework of probabilistic 3D map fusion for collaborative robots. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/89844>

<https://doi.org/10.32657/10220/47722>

A FRAMEWORK OF PROBABILISTIC 3D MAP FUSION FOR COLLABORATIVE ROBOTS

YUE YUFENG

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

23.Feb.2019

.....

Date

.....

YUE YUFENG

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

23.Feb.2019

.....

Date

.....

Supervisor

Prof. Wang Danwei

Authorship Attribution Statement

This thesis contains material from 3 papers published in the following peer-reviewed journals and conference papers where I was the first author.

Chapter 3 is published as Yufeng Yue, Danwei Wang, P. G. C. N. Senarathne, and Diluka Moratuwage, A hybrid probabilistic and point set registration approach for fusion of 3D occupancy grid maps, *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, 2016, pp. 1975-1980.

The contributions of the co-authors are as follows:

- I prepared the manuscript drafts, designed the algorithm, wrote the code and performed the simulations and experiments.
- Prof. Wang provided the initial direction and revised the manuscript drafts.
- Dr. Senarathne assisted in the algorithm design and the revision of the manuscript.
- Dr. Diluka assisted in the revision of the manuscript.

Chapter 4 is published as Yufeng Yue, Danwei Wang, P. G. C. N. Senarathne, and Chule Yang, Robust Submap-Based Probabilistic Inconsistency Detection for Multi-Robot Mapping, *2017 European Conference on Mobile Robotics (ECMR)*, Paris, 2017, pp. 1-6.

The contributions of the co-authors are as follows:

- I wrote the manuscript drafts, designed the algorithm and performed the experiments.

- Prof. Wang provided the initial direction and revised the manuscript drafts.
- Dr. Senarathne assisted in the data collection.
- Mr. Yang helped to revise the manuscript.

Chapter 5 is published as Yufeng Yue, P.G.C.N. Senarathne, Chule Yang, Jun Zhang, Mingxing Wen, and Danwei Wang, Hierarchical Probabilistic Fusion Framework for Matching and Merging of 3D Occupancy Maps, *IEEE Sensors Journal*, vol. 18, no. 21, pp. 8933-8949, 1 Nov.1, 2018.

The contributions of the co-authors are as follows:

- I wrote the manuscript drafts, designed the algorithm and performed the simulations and the experiments.
- Dr. Senarathne assisted in the experiment design and revised the manuscript.
- Mr. Yang helped to revise the manuscript.
- Mr. Zhang assisted in the data collection.
- Mr. Wen assisted in the data collection.
- Prof. Wang provided the initial direction and revised the manuscript drafts.

23.Feb.2019

.....

Date

.....

YUE YUFENG

Acknowledgements

First of all, I would like to express my special and great appreciation to my supervisor, Prof. Wang Danwei, for his invaluable guidance and continuous support throughout my Ph.D study. His encouragement and praise have brought me through the difficulties and harsh times in my Ph.D journey. This thesis would never have been possible without his brilliant ideas and extraordinary drive for research. He is not only my Ph.D supervisor, but also my life mentor.

I am grateful to NTU for providing me with the financial support and lab resources to carry out my research work. I wish to thank Namal, Diluka, Eshan and Pop for all the valuable discussions and the advise. Thanks to Wang Yuanzhe, Yang Chule, Zhang Jun, Wen Mingxing, Yosmar and Dai Bo for their assistance and cooperation in the completion of the project. Thanks to Jiang Wentao, Yang Shuai, Guo Fanghong, Fu Changhong, Zhao Wei, Cao Kun, Peng Guohao, Wu zhenyu, Tang Xiaoyu and Zhang Haoyuan, for their friendship and support during my research. It's my privilege of working with the kind and talented fellow researchers, which has been a rewarding experience.

Furthermore, I would like to thank my parents who instilled in me the belief of hard working and perseverance, and my loving wife Siqi for her endless love and support throughout my Ph.D time, and her family for all the support and encouraging me to do my best.

Abstract

Utilizing multi-robot systems for environmental mapping, inspection or surveillance tasks have found significant attention over the past few years due to the inherent advantages of robustness and efficiency. Due to its limited sensing capabilities and computational power, each robot only has partial information of the surroundings. To maintain a comprehensive understanding of the environment under limited communication bandwidth, fusing local 3D maps generated by individual robots to a globally consistent map is a critical challenge in multi-robot mapping missions. Most of the existing approaches to collaborative mapping are subject to certain challenges such as uncertainty modeling, estimating relative transformation, information integrating, inconsistency detection, and communication management. This thesis proposes a systematic probabilistic multi-robot mapping framework that takes these challenges into account.

First of all, a hybrid probabilistic and point set registration approach for fusion of 3D occupancy grid maps is proposed. Previous approaches ignored the foremost property of the 3D occupancy grid map, which is the occupancy probabilities of individual voxels. To incorporate the probability information, a novel error metric and optimization strategy need to be proposed. The Occupancy Iterative Close Point (OICP) algorithm combines positional coordinate with occupancy probability to register point cloud extracted from the 3D map. The weight between the positional error and the probability uncertainty is controlled by modeling errors using Shannon entropy and positional misalignment. In addition, an environment mea-

surement model is applied to evaluate the relative transformation and a relative entropy filter is used to integrate the map dissimilarities. The accuracy of the proposed algorithm is evaluated using maps generated from both simulated and real world environments, where the result is shown to generate more consistent global maps.

Secondly, submap-based probabilistic inconsistency detection for multi-robot mapping is investigated. The primary goal of employing multiple robots in active mapping tasks is to generate a globally consistent map efficiently. However, all the current approaches focus on single robot inconsistency detection. The challenge comes from designing a framework that is able to detect inconsistency at single robot and multi-robot levels. In this thesis, a novel multi-level approach is introduced to measure the full 3D map inconsistency, where submap-based tests are performed on both single and multi-robot levels. The conformance test based on submaps is done by modeling the histogram of the misalignment error metric into a truncated Gaussian distribution. Besides, the detected inconsistency is further validated through the OICP map registration algorithm proposed in this thesis. The accuracy of the proposed method is evaluated using submaps generated from both indoor and outdoor environments, where the result illustrates its usefulness and robustness for multi-robot mapping tasks.

Thirdly, hierarchical probabilistic fusion framework for matching and merging of 3D occupancy maps is studied by extending the positive results of integrating probabilistic map information in OICP. Existing approaches have relied on point set registration based algorithms for map matching. However, they have not provided a comprehensive analysis of uncertainty modeling, map matching, transformation evaluation and map merging. In addition, high-level geometry features should be incorporated if available. The thesis addresses these issues by proposing a novel hierarchical probabilistic map fusion framework for 3D occupancy grid maps. Before the fusion of maps, map features and their uncertainties are explicitly modeled and integrated. For map matching, a two-level Probabilistic Map Matching (PMM)

algorithm is developed to include high-level structural and low-level voxel features. The statistical testing model is applied to evaluate the matching result, where the acceptance threshold is auto-tuned based on the overlapping between the maps. The results verify the accuracy and efficiency of the proposed 3D occupancy map fusion approach, and exhibit the improved convergence in these scenarios.

Finally, a general probabilistic framework for collaborative 3D mapping using heterogeneous sensors is proposed. There have been few works on 3D map fusion of heterogeneous map data types in arbitrary (structured/unstructured) environments, especially merging of sparse map with dense maps. In the thesis, a general probabilistic framework is proposed to address the integrated map fusion problem, which is independent of sensor types and SLAM algorithms. The framework provides the theoretical basis for computing the relative transformations (map matching) and merging probabilistic map information (map merging), which enables the team of robots to work with heterogeneous sensor types. Since maps produced by heterogeneous sensors possess diverse physical properties, an Expectation-maximization (EM) based map matching algorithm is proposed that automatically decides the numbers of voxels to be associated. To distribute the maps among robots, a time-sequential map merging framework is developed that makes fusing maps from distributed multi-robot system efficient. In addition, the uncertainty embedded in the map and caused by the relative transformation are also propagated and integrated. The proposed approach is evaluated in real world experiments in various environments with heterogeneous sensors, which shows its accuracy and versatility in 3D map fusion for multi-robot mapping missions.

List of Abbreviations

C-SLAM	Collaborative Simultaneous Localization And Mapping
EKF	Extended Kalman Filter
EM	Expectation Maximization
EMM	Environment Measurement Model
GICP	Generalized Iterative Closet Point
HPFF	Hierarchical Probabilistic Fusion Framework
ICP	Iterative Closet Point
KL	Kullback-Leibler
MLE	Maximum Likelihood Estimation
MSE	Mean Squared Error
NDT	Normal Distribution Transform
OICP	Occupancy Iterative Closet Point
PHD	Probability Hypothesis Density
PMM	Probabilistic Map Matching
RFS	Random Finite Set
RV	Random Variable
SEIF	Sparse Extended Information Filters
SLAM	Simultaneous Localization And Mapping

Contents

Acknowledgements	ii
Abstract	v
List of Abbreviations	vi
List of Contents	xi
List of Figures	xvi
List of Tables	xvii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Thesis Objectives and Approaches	4
1.3 Contributions	5
1.4 Thesis Organization	7

2	Technical Background and Literature Review	9
2.1	Collaborative Mapping	10
2.1.1	Single-Robot Simultaneous Localization and Mapping	10
2.1.2	Multi-Robot Simultaneous Localization and Mapping	13
2.1.3	Multi-Robot Map Fusion	16
2.2	Map Matching	18
2.2.1	Registration of Sensor Data	19
2.2.2	Homogeneous Map Matching	20
2.2.3	Heterogeneous Map Matching	22
2.3	Map Merging	24
2.3.1	Map Inconsistency Detection	24
2.3.2	Probabilistic Information Fusion	26
2.4	Conclusions	28
3	Probabilistic Point Set Registration Approach for Fusion of 3D Occupancy Grid Maps	29
3.1	Introduction	29
3.2	OICP Algorithm	31
3.2.1	Uncertainty in Occupancy Probability	35
3.2.2	Uncertainty in Positional Value	37
3.3	Transformation Evaluation and Probability Fusion	39

3.3.1	Transformation Evaluation	40
3.3.2	Relative Entropy Filter	42
3.4	Experimental Results	43
3.4.1	Registration Results	45
3.4.2	Transformation evaluation	46
3.4.3	Probabilistic Map Fusion	48
3.5	Conclusions	49
4	Robust Submap-Based Probabilistic Inconsistency Detection for Multi-Robot Mapping	51
4.1	Introduction	51
4.2	Topological Structure of Map Inconsistency Detection	53
4.2.1	Definition of Node	53
4.2.2	Definition of Consistency	55
4.3	Submap-Based Inconsistency Detection	57
4.3.1	Submap-wise Inconsistency Distance	57
4.3.2	Modeling of Probability Distribution	58
4.3.3	Truncated Gaussian Distribution	60
4.3.4	Inconsistency Testing	61
4.4	Experimental Results	62
4.4.1	Indoor Environment	62

4.4.2	Outdoor Environment	66
4.5	Conclusions	68
5	Hierarchical Probabilistic Fusion Framework for Matching and Merging of 3D Occupancy Maps	70
5.1	Introduction	70
5.2	System Overview	73
5.3	Map Uncertainty Modeling	75
5.3.1	Individual Voxel Uncertainty	75
5.3.2	Structural Edge Uncertainty	77
5.3.3	Local Uncertainty Propagation	80
5.4	Two-Level Probabilistic Map Matching	81
5.4.1	The Formulation of Two-Level Probabilistic Map Matching Problem	82
5.4.2	Probabilistic Data Association	84
5.4.3	Error Metric Optimization	87
5.5	Transformation Evaluation and Probability Merging	89
5.5.1	Transformation Evaluation	89
5.5.2	Relative Entropy Filter	91
5.6	Experimental Results	92
5.6.1	Evaluation Protocol	93

5.6.2	Edge Matching Analysis	94
5.6.3	Full Map Matching Analysis	96
5.6.4	Statistical Testing and Map Merging	101
5.7	Conclusions	104
6	A General Probabilistic Framework for Collaborative 3D Mapping using Heterogeneous Sensors	106
6.1	Introduction	106
6.2	Distributed Multi-Robot Map Fusion	108
6.2.1	System Architecture	109
6.2.2	System Framework Definition and Formulation	109
6.2.3	Map Fusion Definition and Formulation	111
6.3	Multi-Robot Map Matching	112
6.3.1	Mathematic Formulation	112
6.3.2	3D Occupancy Map Matching	113
6.3.3	E-step	116
6.3.4	M-step	117
6.4	Time-Sequential Map Merging	118
6.4.1	Uncertainty Propagation and Transformation	119
6.4.2	Uncertainty merge	121
6.5	Experimental Results	123

6.5.1	Evaluation Protocol	124
6.5.2	NTU Indoor	125
6.5.3	NTU Mixed Ground Floor	127
6.5.4	Changi Exhibition Center	128
6.5.5	NTU Forest Environment	129
6.5.6	Analysis of Experiment Results	131
6.6	Conclusions	133
7	Conclusions and Future Work	135
7.1	Conclusions	135
7.2	Future Work	139
	Author's Publications	141

List of Figures

2.1	An illustration of the SLAM problem, where simultaneously estimate both robot and landmark locations is required [1].	11
2.2	An example of multiple-robot SLAM with two robots [2].	14
2.3	Data fusion at different levels	16
2.4	An example of relative position between two robots.	19
2.5	Raw sensor data of a plane and Octomap generated from the data	21
2.6	The matching of maps generated by heterogeneous sensors.	22
2.7	An example of the inconsistency map, where the inconsistency is denoted in red.	25
2.8	Merging two features.	26
3.1	An example of Octomap and its related point cloud	31
3.2	Sensor model of Octomap	32
3.3	The value of shannon entropy with the change of probability value	36
3.4	Positional error model estimated in a spherical coordinate system.	38

3.5	An example of the registration error between matched features	41
3.6	Pioneer robot and partial map generated from WKW Lobby	44
3.7	Initial transformation for point cloud generated from Octomap	44
3.8	Mean Error in both rotation and translation for the experiments, blue color represents the results conducted in simulation, while the results conducted in WKW lobby is shown in red	46
3.9	Registration result of WKW Lobby	47
3.10	Registration result of simulated environment	47
3.11	Average Mahalanobis Distance	48
3.12	The percentage of inliers	48
3.13	The fused map of simulated environment	49
3.14	The fused map of WKW Lobby	49
4.1	An example of consistent and inconsistent map	52
4.2	Topological structure of map inconsistency detection.	54
4.3	An example of misaligned submaps and the histogram of inconsis- tency distance	58
4.4	The environment of the experiment	61
4.5	The ground truth and inconsistent map generated in indoor environ- ment.	63
4.6	The large difference of the color in the same pixel verifies the incon- sistency.	64

4.7	The top row shows the detected global inconsistency between submap b_9m (blue color) and a_8m (red color), while the bottom row verifies the inconsistency after performing map registration	66
4.8	The matrix Γ^v shows the verified inconsistent pairs by identifying the difference between Fig.4.6(a) and Fig. 4.6(b) in indoor scenario. . . .	67
4.9	Verified inconsistency matrix in outdoor environment. The deeper color represents a higher probability of inconsistency.	68
4.10	The inconsistent map generated in a car park.	68
5.1	An example of the proposed hierarchical probabilistic fusion framework.	71
5.2	The left image shows two consecutive scans of a continuous surface collected by the sensor.	73
5.3	The diagram of the system.	74
5.4	An example of the sensor uncertainty and motion uncertainty in the 3D map.	76
5.5	The extraction of structural edge.	79
5.6	The process of combing individual voxel covariance (left) and structural edge covariance (right), where the middle image shows the merged result (middle).	80
5.7	An example of data association established between two L-shape datasets.	86
5.8	Given the same datasets with two different rigid transformations. . .	88
5.9	Comparison between edge registration algorithms, where first level of PMM is compared against ICP based edge registration algorithm. . .	95

5.10	Sensitivity to the change of overlap in WKW dataset, where ICP edge matching and first level PMM are compared.	96
5.11	Summary of convergence of the evaluated methods under 125 initial perturbations. Translation error (top) and rotation error (bottom). .	97
5.12	Summary of success rates of the evaluated matching methods over all datasets. Translation estimation (top) and rotation estimation (bottom).	99
5.13	The distribution of the average Mahalanobis distance between the matched pairs in all three experiments.	102
5.14	The fraction of inliers and the threshold of accepting the estimated transformation.	103
5.15	The results of fused maps for all the three experiments.	104
6.1	An example of the heterogeneous map fusion.	107
6.2	The diagram of the system.	110
6.3	Top: Robot platform and the fused 3D map created using the two robots. Bottom: The diagram of the group of robots.	111
6.4	The density map and histogram for the two sensors.	115
6.5	Density based multiple data association strategy.	117
6.6	The process of uncertainty propagation and merging.	121
6.7	The experiment setting up: NTU Indoor, NTU Mixed, Changi Centre, NTU Forest respectively.	123

6.8	The experiment of heterogeneous map matching with different data association strategies.	125
6.9	A demonstration of multi-level map fusion result in NTU indoor experiment.	126
6.10	A demonstration of multi-level map fusion result in NTU Mix experiment.	127
6.11	A demonstration of the collaborative mapping system in real-time at exhibition center.	129
6.12	A demonstration of the collaborative mapping system in real-time at NTU forest.	130
6.13	The average numbers of voxels to be associated in our algorithm during the four experiments.	132
6.14	The processing time during the four experiments.	132

List of Tables

2.1	The challenges for multiple robots	15
2.2	The types of data fusion	17
3.1	The summary of setting up of the experiments	44
3.2	The entropy after map fusion	48
5.1	Summary of the threshold parameters	91
5.2	Summary of the experiment environment.	93
5.3	The mean error in both rotation and translation estimations. (The best performance is denoted in bold.)	98
5.4	The efficiency comparison of the algorithms (Seconds)	101
5.5	Average entropy of fused map for different update rule. (The best performance is denoted in bold.)	104
6.1	Summary of the experiment environment.	123
6.2	The root mean error in all experiments. (The best performance is denoted in bold.)	131

Chapter 1

Introduction

1.1 Background and Motivation

In the field of robotics and artificial intelligence, autonomous robot has been a highly active research area, which enables a robot to perform behaviors with a high degree of autonomy. With the continuing development in the field of mobile robotics, in particular, using multiple robots to autonomously explore and map an environment has caught the attention of many robotics researchers and practitioners. Utilizing a group of robots have found significant advantages compared to their single robot counterpart: robustness to single robot failure, quicker exploration of environments in time critical search and rescue missions, execution of tasks of high complexity, reduction of human risks, and costs associated to disaster response. Accurate online localization and mapping is a crucial competency for enabling collaboration between multiple mobile robots. Up to now, there are several review papers that have summarized the existing results in this area [2, 3, 4, 5]. Employing multiple robots introduces a new set of challenges such as multi-robot mapping [6, 7, 8], multi-robot localization [9, 10, 11], distributed system [12, 13, 14] and coordinated exploration [15, 16, 17, 18], to name a few. Among the above mentioned research hotspots

in this area, collaborative mapping[19, 20, 21], which is to combine and distribute the data perceived by individual robots to maintain a comprehensive understanding of the environment, is one of the most critical topics which has been investigated extensively in the past decade (see [2] for an in-depth survey).

Detailed collaborative mapping of unknown environments is a major research problem in mobile robotics that has garnered significant attention in recent years with the improvements in 3D sensors, 3D mapping algorithms, and processing systems. Due to limited sensing capabilities and communication bandwidth, each robot only has partial information of the surroundings. Since the operating environment gets larger, accuracy, efficiency and robustness of collaborative mapping become a critical factor. In order to allow each robot to make decisions and plan tasks from a holistic viewpoint, sharing and fusing map data perceived by each robot among all robots to maintain a comprehensive understanding of the environment is a fundamental requirement for multi-robot coordination. This is however a difficult task as stated by [2], which has identified ten major challenges to achieve online Collaborative Simultaneous Localization and Mapping (C-SLAM).

Many challenges exist in collaborative mapping and attempts have been made from different aspects to address them, such as estimating the relative transformation between the maps [22, 23], accounting for the uncertainty of the relative transformation [7], integrating the information from neighboring robots [24], dealing with complexity issues[13], manage the communication [19], updating maps in the global perspective [25], and many more. This work is motivated by the fact that there are several challenges in addressing some of the related issues of collaborative mapping. The major challenge in these systems is to design a general framework for collaborative mapping, which covers map uncertainty modeling, probabilistic map matching, transformation evaluation, map merging and communication management:

1. *General framework for collaborative mapping.* The research of exploiting multi-robot map fusion framework in both single robot and multi-robot levels has

not been studied extensively. In general, map fusion on both levels is composed of estimating the relative transformation and integrating the partial map information, which are usually tackled separately. In summary, previous approaches have not provided comprehensive analysis of the integrated map fusion system on map uncertainty modeling, probabilistic map matching, transformation evaluation, accurate map merging and communication management.

2. *Probabilistic map matching.* Most of the existing map matching algorithms belong to the scope of iterative dense registration, which is designed for raw sensor data registration on point/voxel-wise level. Due to the intrinsic nature difference between raw sensor data and 3D occupancy grid maps, the direct use of dense registration algorithms in 3D occupancy grid map matching results in sub-optimal voxel associations thus sub-optimal matching. To increase the accuracy and convergence, available information like geometry information, and occupancy probability should be incorporated in the matching process.
3. *Transformation evaluation and inconsistency detection.* The majority of the research is focused on generating accurate maps and post-processed optimization. However, the generated map does not guarantee to be foolproof and may generate inconsistent maps. Fusing erroneous local maps results in an inconsistent global map, which renders the accurate execution of autonomous tasks infeasible. Therefore detecting inconsistency in both local maps and fused global maps is vital for multi-robot missions. The capability to detect inconsistency in the generated maps allows robots either to perform an efficient optimization algorithm, or to return to a safe location for recovery without getting trapped.
4. *Collaborative mapping using heterogeneous sensors.* For a group of robots in arbitrary (structured/semi-structured/unstructured) environments, the robot might be equipped with multiple sensors. To utilize their complementary characteristics, map generated with heterogeneous sensors need to be fused.

Existing approaches have relied on point set registration based algorithms for map matching, where hard one-to-one correspondence relationship is assigned. Since the intrinsic characteristics (density/range/field of view) of the map produced by heterogeneous sensors are very different, a more flexible and general map registration algorithm is required and should be independent of sensor types.

1.2 Thesis Objectives and Approaches

The main objective of this research is proposing the collaborative robots 3D mapping framework so as to improve the accuracy, efficiency and robustness. The framework will act as the theoretical basis for multi-robot mapping missions, which enables the team of robots to work with heterogeneous sensors in arbitrary (structured/semi-structured/unstructured) environments. The proposed solutions need to overcome key practical issues such as uncertainty modeling, probabilistic map matching, transformation evaluation, information integration under limited communication bandwidth and computational power.

As the thesis is concerned about the objectives mentioned above, a systematic probabilistic fusion framework for collaborative robots 3D mapping is proposed. More specifically, occupancy iterative closet point (OICP) is proposed as a probabilistic 3D map fusion algorithm, which combines positional coordinates and occupancy probability. To detect the inconsistency during collaborative mapping, submap-based inconsistency detection strategy is proposed to evaluate the misalignment error on local and global map levels. To provide a comprehensive analysis of the map uncertainty modeling on high level geometric features and low level voxels, a novel hierarchical probabilistic map fusion framework is proposed for 3D occupancy grid maps. For robots equipped with heterogeneous sensors, a general probabilistic framework for multi-robot 3D mapping is developed and is independent of sensor

types and SLAM algorithms.

1.3 Contributions

This thesis investigates the collaborative mapping problem for multiple mobile robots in unknown environments. Efforts are made from the following perspectives: first of all, a probabilistic 3D map fusion algorithm occupancy iterative closet point (OICP) is proposed for 3D map registration. Secondly, submap-based inconsistency detection strategy is proposed to evaluate the misalignment error on local and global map levels during collaborative mapping missions. Thirdly, a novel hierarchical probabilistic map fusion framework for 3D occupancy grid maps is studied. Fourthly, a general probabilistic framework for multi-robot 3D mapping is proposed, which is independent of sensor types and SLAM algorithms. Extensive simulations and real robot experiments have been conducted to validate the proposed approaches.

1. *Probabilistic point set registration approach for fusion of 3D occupancy grid maps* (Chapter 3).

Firstly, probabilistic information is incorporated in the registration algorithm, which is called Occupancy Iterative Closet Point (OICP). OICP combines the strength of ICP with the associated probabilistic information of 3D occupancy map, which tends to match voxels with similar occupancy probability. Secondly, an environment measurement model (EMM) is applied to decide whether to accept or reject the computed relative transformation. Then a relative entropy filter is proposed to fuse the map by combining occupancy probability from both maps which also decreases the overall global map uncertainty. Experimental results show that the proposed algorithm outperforms existing algorithms with improved accuracy.

2. *Robust submap-based probabilistic inconsistency detection for multi-robot mapping* (Chapter 4).

A novel multi-level approach is introduced that detects the inconsistency of maps generated from two levels, i.e., single robot and multi-robot level. The inconsistency of the system is modeled as a probabilistic distribution, where submap-based inconsistency testing is performed on three types of inconsistency defined on both single robot and multi-robot level. In both levels, map inconsistency measurements are computed by modeling the misalignment errors between submap pairs into a truncated Gaussian distribution, and the resultant mean and variance are used to detect the inconsistency. Besides, the detected inconsistency is further validated through a 3D map registration process. The fused global map is deemed consistent only if all submap pairs of all robots pass the inconsistency checks. The algorithm is evaluated using submaps from challenging environments in both indoor and outdoor, which illustrates its usefulness and robustness for multi-robot mapping tasks.

3. *Hierarchical probabilistic fusion framework for matching and merging of 3D occupancy maps* (Chapter 5).

The hierarchical probabilistic fusion system is proposed to address the integrated 3D occupancy map fusion problem, which is composed of three modules: uncertainty modeling, map matching and map merging. Based on the in-depth analysis and propagation of the 3D map uncertainty model, the two-level Probabilistic Map Matching (PMM) algorithm is developed by combining both high-level structural edge and low-level occupancy voxel features. The statistical testing model is applied to evaluate the matching result, where the acceptance threshold is auto-tuned based on the overlapping between the maps. Then, the individual maps are merged by combining occupancy probability based on a relative entropy filter. The results verify the accuracy and efficiency of the proposed 3D occupancy map fusion approach, and exhibit the improved convergence in these scenarios.

4. *A general probabilistic framework for collaborative 3D mapping using heterogeneous sensors* (Chapter 6).

Firstly, a general probabilistic framework is proposed to address the integrated map fusion problem, which is independent of sensor types and SLAM algorithms. The framework provides theoretical basis for computing the relative transformations (map matching) and merging probabilistic map information (map merging). Secondly, the probabilistic map matching algorithm is proposed that automatically decides the numbers of voxels to be associated for maps produced by heterogeneous sensors with different physical properties. Thirdly, a time-sequential map merging framework that makes fusing maps from a distributed multi-robot system efficiently, where uncertainty embedded in the map and caused by the relative transformation are propagated and integrated. The proposed approach is evaluated in real world experiments in various environments with heterogeneous sensors, which shows its efficiency and generality in 3D map fusion for distributed multi-robot mapping missions.

1.4 Thesis Organization

The remainder of this thesis is organized as follows.

- Chapter 2 gives an extensive survey of collaborative mapping methods. The main approaches published in the last ten years with regard to our common thread are reviewed.
- Chapter 3 details the probabilistic point set registration approach for fusion of 3D occupancy grid maps. This chapter first presents Occupancy Iterative Closet Point algorithm to register the maps. Then, the evaluation of the computed relative transformation and the fusion of probabilities is presented. The accuracy of the proposed algorithm is evaluated using maps generated from both simulated and real environments.

- Chapter 4 elaborates robust submap-based probabilistic inconsistency detection for multi-robot mapping. The topological probabilistic structure for submap-based probabilistic inconsistency detection is presented. More detailed submap-based inconsistency modeling and detection is analysed and formulated. The accuracy of the proposed method is evaluated using submaps from challenging environments in both indoor and outdoor, which illustrates its usefulness and robustness for multi-robot mapping tasks.
- Chapter 5 explains hierarchical probabilistic fusion framework for matching and merging of 3D occupancy maps. The uncertainty of 3D occupancy map data at voxel level and structural level are analysed in depth. The two-level probabilistic map matching framework based on the uncertainty modeling is formulated. Then, the statistical testing for transformation estimation and the relative entropy-based map merging approach are proposed. The results verify the accuracy and efficiency of the proposed 3D occupancy map fusion approach, and exhibit the improved convergence in these scenarios.
- Chapter 6 introduces a general probabilistic framework for collaborative 3D mapping using heterogeneous sensors. The chapter formulates the joint probability of the global map and the set of relative transformations between the local maps. Then, it also proposes a time-sequential map merging framework to share map information among robots. The proposed approach is evaluated in real-time experiments in various environments with heterogeneous sensors, which shows its robustness and generality in 3D map fusion for multi-robot mapping missions.
- Chapter 7 summaries this thesis. The contributions of this work together with the expected future work are provided.

Chapter 2

Technical Background and Literature Review

In this chapter, an extensive survey related to existing approaches of collaborative mapping is provided. Among these solutions, map matching (finding relative transformation), integrating the information from neighboring robots, dealing with complexity issues, managing the communication network, and updating maps in the global perspective are the majority. The main approaches published in the last ten years with regard to multi-robot mapping method are reviewed.

The remaining part of the chapter proceeds as below. In Section 2.1, single-robot SLAM and multi-robot SLAM problems are reviewed and main approaches to collaborative mapping are summarized. Section 2.2 summarizes the state of the art of the map matching solutions, where registration of raw sensor, homogeneous map matching and heterogeneous map matching are thoroughly reviewed. Sec 2.3 describes the distributed map sharing, map inconsistency detection and probabilistic information fusion, which are categorized into the scope of the map merging problem. Sec 2.4 concludes the chapter.

2.1 Collaborative Mapping

This section first explains the single-robot SLAM problem, which gives the background information of multi-robot SLAM. While the single-robot SLAM is challenging enough, moving to a platform of multiple robots adds another layer of challenge. Challenges in multi-robot SLAM are investigated, and solutions for these challenges are reviewed. In a multiple-robot environment, robots must incorporate all available data to construct a consistent global map, which acts the basis for multi-robot missions. Collaborative mapping has benefits such as performing missions faster and being robust to failure, where the challenges of maintaining such a complex system are also summarized.

2.1.1 Single-Robot Simultaneous Localization and Mapping

This section presents various approaches existed to solve single robot SLAM problem. The reason for this section is to summarize the available solutions and to have a better understanding of multi-robot coordination problem. SLAM describes a problem of placing a robot in an unknown environment without GPS information, which has to map the environment and localize itself with respect to the map. Both the trajectories of the robot and the map should be estimated online by using the external sensors like cameras, lasers, radars, or internal sensors like IMUs, wheel encoders. Solving the SLAM problem and providing a reliable online algorithm is a key technology for autonomous robots.

Assuming that robot poses are $x_{1:k} = \{x_1, x_2, \dots, x_k\}$, where x_k is the current post of the robot at time k . The map at time k is denoted by the set $m_k = \{m_{k,1}, m_{k,2}, \dots, m_{k,l_k}\}$, and l_k represents the number of voxels(landmarks) at time k . The observations made by the robot are $z_{1:k} = \{z_1, z_2, \dots, z_k\}$, and $z_k = \{z_{k,1}, z_{k,2}, \dots, z_{k,n_k}\}$ is the sensor observation set perceived at time k , while n_k presents the number of observations. The control sequences are $u_{1:k} = \{u_1, u_2, \dots, u_k\}$, and

Figure 2.1: An illustration of the SLAM problem, where simultaneously estimate both robot and landmark locations is required [1].

u_k is the control command applied to the robot at time k .

An online SLAM problem is defined to only estimate the posterior over the current position x_k and map m_k , which is:

$$p(x_k, m_k | z_{1:k}, u_{1:k}, x_0) \quad (2.1)$$

Based on some initial job in [26], extended kalman filter (EKF) is proposed in [27, 28, 29]. EKF models the feature map as vectors and linearize the nonlinear function to achieve efficiency. However, solving the state vector will be highly computational expensive and the linearization also cause large errors if the environment is highly non-linearized. Then Fast SLAM 1.0 and 2.0 are proposed in [30] and [31], respectively, which applies Monte Carlo Sampling with particles to recursively estimate the non-linear and non-Gaussian environment. However, Fast SLAM has to maintain a numerous number of particles and is prone to impoverishment by resampling. In addition to EKF and Fast SLAM, a notable approach called sparse extended information filters (SEIF) is proposed in [32], which can update equations and execute in constant time irrespective of the size of the map. For all the SLAM methods mentioned above, they all have to deal with the data association problem, which means to determine the correspondence between measurements and landmarks. The data association problem is always solved with a measurement model [33], scan matching [34] or an image feature extraction algorithm [35]. To incorpo-

rate the data association problem to the Bayesian filter, a random finite set(RFS) based on probability hypothesis density(PHD) filter is proposed in [36].

Full SLAM problem involves a lot of computational power, and may not process online. Then, a full SLAM problem is to estimate the trajectories of the robot at each time step $x_{1:t}$ and the posterior over the map m , which is:

$$p(x_{1:k}, m_k | z_{1:k}, u_{1:k}, x_0) \quad (2.2)$$

A full SLAM algorithm has to optimize the overall posterior of robot trajectories and map, and minimize the error caused by measurements and control input. In [37], a full SLAM algorithm called graph SLAM is proposed, which optimize the overall trajectories and landmarks by nonlinear minimization offline. The graph SLAM transfers the posterior into a graph network, and then reduce the variables to get a sparse matrix for minimization solving. In [38], an incremental smoothing and mapping approach is proposed based on fast incremental matrix factorization to perform full SLAM online. As shown in [39], graph SLAM is extended to large scale using aerial images as prior information. To achieve global convergence, loop closure detection is one of the key challenge. Loop closure means when a robot revisits a place, the robot is able to recognize. In large environment, a new edge will be added to the graph when a loop closure is detected, which will largely reduce the error caused by autonomous exploration. In [40], a switchable constraint is proposed for robust loop closure detections against outliers like data association errors and false positive. [41] extends loop closure to incremental searching and multi-sessions with the applying of g2o [42], which is a famous graph slam package.

Visual SLAM has a long history. Early visual SLAM systems are originated from conventional SLAM, where sequential filtering is used [43]. Given robot control signals and feature measurements, by assuming that the state-space estimation and measurements follow Gaussian distribution, a probability framework is formulated to deal with errors from different sources (e.g. motion model and observation model).

Current visual SLAM takes SLAM as graph optimization process based on bundle adjustment and loop closing technique. Double window and multi window optimization which are well-suited for different environments are also explored [44]. Instead of taking all the frames and all the features into account, the concept of keyframe” is commonly used to maintain the sparsity of graph. A frame is selected as keyframe only if the overlap between current frame and previous keyframe is lower than a threshold [45]. The connections between all the nodes of the graph are decreased dramatically compared with filtering approach. Real time implementation is no longer too difficult [46]. A very detailed comparison study on filtering and optimization approach has been given in [47].

2.1.2 Multi-Robot Simultaneous Localization and Mapping

Since single robot SLAM is difficult enough, enabling a group of robots to perform tasks autonomously brings challenges at a higher level. Coordination among multiple robots enable them to perform difficult tasks in challenging environments more efficiently and reliably, some of the applications include: earthquake surveillance [8], forest mapping [10], detection and tracking of moving objects [15], intelligent transportation [17] underwater mapping [6]. The approaches reviewed in last section is the base for multi-robot SLAM, and additional challenges include robot-wise communication, relative position estimation, loop closure detection, uncertainty updating and data fusion, which is summarized in Tab.2.1.

Assuming that robot poses are $x_{1:k}^{(r)} = \{x_1^{(r)}, x_2^{(r)}, \dots, x_k^{(r)}\}$, and $x_k^{(r)}$ represents the pose of the robot r with in its local coordinate frame at time k . The map at time k is given as $m_k^{(r)} = \{m_{k,1}^{(r)}, m_{k,2}^{(r)}, \dots, m_{k,l_k^{(r)}}^{(r)}\}$, where $l_k^{(r)}$ represents the number of voxels (landmarks) at time k . Then the global map M at time k is given as $M_k = \{m_k^{(1)}, m_k^{(2)}, \dots, m_k^{(r)}\}$. The sensor observations made by each robot are given as $z_{1:k}^{(r)} = \{z_1^{(r)}, z_2^{(r)}, \dots, z_k^{(r)}\}$, and $z_k^{(r)} = \{z_{k,1}^{(r)}, z_{k,2}^{(r)}, \dots, z_{k,n_k^{(r)}}^{(r)}\}$ denotes the sensor observation set perceived from robot r at time k , while $n_k^{(r)}$ is the number of observations at this

Figure 2.2: An example of multiple-robot SLAM with two robots [2].

time stamp. The control inputs given to the robots are $u_{1:k}^{(r)} = \{u_1^{(r)}, u_2^{(r)}, \dots, u_k^{(r)}\}$, and $u_k^{(r)}$ represents the control command given to robot r at time k . Then, a full multi-robot SLAM problem is to estimate the continuous trajectories of each robot r at time step $x_{1:k}^{(1,\dots,r)}$ and the update global map M_k , which is:

$$p(x_{1:k}^{(1,\dots,r)}, M_k | z_{1:k}^{(1,\dots,r)}, u_{1:k}^{(1,\dots,r)}, x_0^{(1,\dots,r)}) \quad (2.3)$$

To extend single robot SLAM to multi-robot SLAM, various solutions based on EKF have been proposed, such as merge sensor and navigation information from multiple robots in [48, 17, 49]. Extension from a single robot to multiple robots using EKF is straightforward, however, it suffers from linearization error and exponentially growing computational burden. Multi-robot based on particle filters is proposed in [50], which starts with not known prior information and integrates all data into a single common map. The particle filters based approach is further improved to deal with limited communication in large scale environment [51]. Alternatively, a graph SLAM based multi-robot coordination algorithm is proposed in [52], which is based on stochastic estimation and central convergence is proved. In [18], the

Table 2.1: The challenges for multiple robots

Type	Descriptions
Communications	The way of robots communicating with each other, it can be centralized and distributed.
Relative position	When multiple robots are performing tasks, the robot need to know the position of other robots in a global coordinate frame
Complexity	The complexity of operating multiple robots, it is important to enable the robots to perform in real time with computational power limitation.
Data fusion	One of the most important reason of applying multiple robots is to fuse the measurements generated by individual robots and provide the global and full map of the explored environment.

author utilizes a cost function to enable a team of robots to explore an unknown environment in a minimum cost of reaching a target point. In [11], a neural network-based multi-robot SLAM is proposed. To handle dynamic environment with multiple robots, multi-robot coordination based on random finite set is proposed [53].

Aforementioned approaches mainly work in centralized manner, distributed multi-robot coordinate brings the challenges of data sharing, relative localization, loop closing and communication management. In [13], the author presents a distributed data fusion network for robots to share map information with neighbors under the limited communication bandwidth. In [7], distributed real-time cooperative localization and mapping is achieved using an uncertainty-aware expectation maximization approach. In [54], the paper addresses the problem of determining the relative pose between robots. Specifically, the author proves the minimum number of distance measurements required for determining all six possible solutions for 2D robot SLAM is three. In [55], the paper shows the loop closure detection with multiple robots. In [56], the author presents a map fusion algorithm of multi-robot slam using reinforcement learning. Submap based SLAM has been studied [57] [58], which combines with loop closure detection [59].

2.1.3 Multi-Robot Map Fusion

Detailed 3D mapping of unknown environments is a major research problem in mobile robotics that has garnered significant attention in recent years with the improvements in 3D sensors, 3D mapping algorithms, and processing systems [2]. Utilizing multiple robots to conduct the mapping is going to be inevitable. Many modern autonomous mobile robotic applications ranging from environmental mapping [2, 21], surveillance [8] missions to autonomous vehicles require a detailed 3D map of the environment as a prerequisite for their optimal operation. Since the operating environment gets larger, efficiency and robustness of map generation become a critical factor [9, 15].

(a) Data fusion with all the raw sensor data [21]

(b) Data fusion with topology structure [14]

(c) Data fusion with 2D grid map [24]

(d) Data fusion with 3D Occupancy map

Figure 2.3: Data fusion at different levels

Table 2.2: The types of data fusion

Data Type	Descriptions
Raw sensor data	fuse whatever the sensor observed, like point cloud, images, odometry
Volumetric maps	fuse the maps generated by the robots using the sensor measurement, which can be 2D map and 3D map
Topology Structure	fuse the topology structure extracted from the maps, it can be triangulation, voronoi

The major challenge in these systems is to fuse the maps generated by individual robots (i.e., local maps) into a globally consistent map (i.e., a global map). This is further exacerbated when the robots are required to operate in a networked environment with real or semi-real time mapping requirement, where the local maps are to be transferred over the network for map fusion. Equipped with various sensors, the laser sensor data and camera image data are very different at the raw sensor data level [60], where accurate transformation between the maps need to be calculated. Under the condition of limited bandwidth and coverage of communication channels [13], large amount of data is hard to transfer in real time [19]. Real or semi-real time global mapping with limited network bandwidth requires a careful consideration for the choice of map type that balances the requirements of detailed 3D mapping, support for transfer over limited communication and their utility for robust map fusion [24]. Some researches have been done to address data fusion among multiple robots which can be summarized into three different types (see Tab.2.2 and Fig. 2.3). Raw sensor data [20, 61], volumetric maps [62, 25, 22] and topological maps [63, 23] are some of the different types of maps that can be used to represent the 3D environment.

While raw sensor data provides more detailed information about the environment, they are extremely challenging to store, manage and process and even more so to transfer over networks with limited connectivity. Therefore, the point cloud is not an ideal choice of map type for real-time multi-robot mapping missions. However, the rich details in these maps have encouraged the development of 3D map fusion algorithms [20], albeit the off-line high computational effort. Topological maps, on

the other hand, are easier to process and transfer over the network due to their lower memory footprint. The literature contains map fusion algorithms involving 2D topological maps [24, 63, 23]. However, the high level of abstraction in these maps do not lend well for detailed 3D environment mapping and hence for 3D map fusion. Metric grid maps such as volumetric maps [64, 65] can represent the 3D environments in sufficient detail without sacrificing the memory or processor usage. Combined that with the availability of compression schemes that allow real-time transferring over limited bandwidth makes them the ideal choice of map type for multi-robot mapping missions. Octomap [65] is a volumetric map type with a high compression ratio that can represent any type of 3D environment, hence is selected as the choice of map type in this research.

2.2 Map Matching

The objective of map matching is to estimate the relative transformation between the robots as shown in Fig.2.4. The two robots build local maps with coordinate frame L_a and L_b , then the local maps should be transformed into global coordinate. Over the last years, various research groups have developed efficient solutions to the map matching problem. Map matching can be performed at different levels, and should consider the trade-off between efficiency and quality. These techniques range from 2D grid map matching [63, 66, 67], to sparse feature descriptors [68, 69], to full 3D dense occupancy map registration [70, 71]. Here, the work focuses on map matching and makes the assumption of the local map generated by each robot is consistent. As the majority of the initial solutions to 3D map fusion is inspired by registration algorithms, sensor data based dense registration algorithms are reviewed first. Then, the approaches of map matching are reviewed, where the strengths and limitations are carefully discussed.

Figure 2.4: An example of relative position between two robots.

2.2.1 Registration of Sensor Data

To establish the point-wise correspondence, dense registration algorithms are directly operated on the data obtained by range sensors like Kinect, laser scanners. The objective is to find a 3D rigid transformation to register the consecutive sensor scans. The well known Iterative Closet Point (ICP) [72] is the most significant algorithm, which iteratively searches for corresponding pairs and computes the transform that minimizes the Euclidean distance. Since the introduction of ICP, many variants based on ICP have been proposed to improve the aspects like pre-filtering [73, 74], data association [75, 76, 77, 78], outlier removal [79, 80] and error function optimization [81, 82, 83, 84].

Prior to registration, scan data is usually preprocessed to extract features like planar patches [73], or down-sampling [74] to increase the efficiency. To find the data association: ICP assumes the points in two sources have physical spatial correspondence, however, this is violated in the presence of sensor noise and limited sensor density. To model the sensor noise, a probabilistic version is proposed in [85], where points are represented by a normal distribution. In [76], probabilistic data association is applied to align two point clouds with different densities. To deal with the sensor

data with low sampling resolution, [86] represents the point cloud by a line cloud by random generation of Collar Line Segments (CLS). While purely using position information might fall victim to correspondence ambiguity, additional information extracted from the point cloud like local plane structure [77], curvature&intensity [87], color information [78] are incorporated as multi-channel sources.

Once the data association established, tuning the outlier filter is a critical task for the success of the alignment. To filter out the outliers, each match is weighted by the inverse of the distance between the points involved in the matching process in [79]. In [80], outlier filtering is automatically tuned with respect to the overlap variations. Based on the true correspondence, many ICP variations use a closed form solution to iteratively compute the alignment based on cross-correlation of point-cloud [81]. To allow more generic minimization function other than simply Euclidean distance, nonlinear optimization framework is introduced to add additional constraints and dimensions based on $L2$ norm distance [82], such as more penalization along the planar patch direction [83] or edge direction [84].

2.2.2 Homogeneous Map Matching

Considering the efficiency and accuracy, matching the occupancy map is more suitable than raw sensor data and abstract topology. Firstly, a compact map is generated by compressing raw sensor data, as shown in Fig.2.5, the large amount of data has been compressed about ninety percent. Secondly, although some information will lost when map is generated, the state-of-art three dimensional mapping algorithm would preserve the most of the sensor data, such as octomap shown in Fig.2.5. In other words, sharing the occupancy/landmark maps [88], or just sharing the compact map is preferred.

For 2D grid maps, topology-based approaches like hough transform [89] and Voronoi diagram [23] are applied to represent occupancy map on the abstract level. To calcu-

Figure 2.5: Raw sensor data of a plane and Octomap generated from the data

late the transformation, a variety of standard topology map attributes are evaluated to compute the subgraph-isomorphisms in [67]. In [66], the author presents a hierarchical SLAM approach which uses the spectral information to register the local submaps. Besides, the probabilistic Voronoi matching is introduced to incorporate the motion uncertainty in [24]. For feature-based maps, Delaunay triangulation based topology network is introduced to match the features in [14]. In [61], structural descriptors are extracted to match LIDAR point-cloud and sparse vision keypoint maps. Similarly, a histogram cross-correlation technique is proposed to perform map matching at both local and global levels in [68]. [69] feature-based map merging with dynamic consensus on information increments.

With the increasing availability of high-quality 3D sensors, the dense registration algorithms have been employed to match maps composed of raw sensor data. In [19], probabilistic hypothesis density(PHD) SLAM is employed to match sparse features in simulation. CSHOT [90] descriptor and ICP [72] are integrated for submap-based map matching in [22]. For back-end optimization, [20] and [71] integrate raw sensor data to construct submaps and optimize constrained graph structures. However, aforementioned 3D map matching approaches are based on raw sensor data instead of probabilistic maps.

Since 3D occupancy grid map is generated by compressing the raw sensor data with a sensor measurement model [65], observation noise and motion uncertainty have been embedded into the map. The raw sensor data generated from the simulated environment are highly compressed to occupied voxels. Hence, the available characteristics of the occupancy map is modeled and combined into the registration

Figure 2.6: The matching of maps generated by heterogeneous sensors.

process. The fusion of 3D volumetric maps generated using 3D probabilistic map is first proposed in [70], which utilizes ICP [72] to calculate the relative transformation by minimizing point-wise Euclidean distance. In [91], it is proven that registration algorithm combining invariant features will converge if the weight parameter is non-increasing. A recent paper [92] reviews the 3D scan matching methods up to date, concluding that Normal Distribution Transform (NDT) [85] provides most accurate results compared to others. NDT models the point cloud into a collection of Gaussian distributions with covariance and mean, and minimizes the sum of $L2$ norm. Since in our case the points are evenly distributed with a distance of the size of a voxel, the Gaussian distribution may not represent the environment explicitly compared to rich sensor data, so the error metric loses its original superiority. Based on this observation, probabilistic map information is incorporated to improve the accuracy of the estimated transformation in [25].

2.2.3 Heterogeneous Map Matching

For a group of robots in arbitrary (structured/semi-structured/unstructured) environments, single sensor is far from enough in mapping accuracy and robustness [60, 93]. Maps generated with heterogeneous sensors need to be matched to combine information with a higher information content [76] (see Fig. 2.6). Inspired by

the probabilistic characteristic of expectation maximization(EM), several works consider more generalized scenario of correspondence between point sets with varying density. In [79], the author proposed an EM based ICP to improve the registration precision by considering the gaussian noise between the point sets. In [94], the author proposed a learning anisotropic ICP, which doesn't require manual tuning and update the parameters continually. In [76], the author further exploits the data association relationship and proposes an algorithm to align a sparse point cloud with a dense one, where each point is associated with a set of points. Then, EM based data association is extended to include semantic information and features, as well as internal measurements in [95].

For calibration on sensor level, research has been done in the field of extrinsic calibration between a 3D lidar, visual camera and thermal camera in the past decades. [96] first proposes to use checkerboard to calibrate a 2D lidar and a camera. Traditionally, one checkerboard should be put at different positions and orientations to collect multiple images, which is laborious. To alleviate the problem, [97] proposes to put several checkerboards at various positions and orientations at a time, so that taking only one shot is enough. They also make the calibration toolbox publicly available [98]. Recently, [99] proposes a novel checkerboard and is particularly suitable for the sparse 3D lidar. Following [99], [100] makes a progress by improving the detection accuracy of the circles. However, [100] only applies to a stereo camera. Meanwhile, a successful work on calibration of 3d-thermal-depth can be found in [101] [102]. However, the calibration on sensor level requires real time transferring raw sensor data among the robots, which is very challenging for real multiple robots applications.

The map matching algorithm of the thesis is similar to [76], however, the author only hard codes the K nearest neighborhood and does not discuss how to automatically tune the multiple data association strategy to balance the accuracy and efficiency. In [61], the author proposed a structural descriptors for matching point-cloud maps and vision keypoint maps. However, the paper presents a dense point-cloud map

by rotating the 2D laser (Hokuyo) in off-line, 3D laser (Velodyne) usually tends to produce sparse data in real time. In addition, the structural descriptors are usually sensitive to parameters and requires fine tuning. In [60] the registration method is based on the matching of corresponding planar segments that are extracted from the point clouds. An off-line approach for a globally optimization is also presented. In [93], the author compares the following descriptors a) local features: key-points or segments; b) descriptors: FPFH, SHOT, or ESF; and c) transformation estimations: RANSAC or FGR for registration of point clouds produced by Lidar and camera.

2.3 Map Merging

2.3.1 Map Inconsistency Detection

In registration algorithm, there is no guarantee that a global minimum will be found. A wrong transformation would lead to an inaccurate global map, which in turn is detrimental for robot exploration. Over the last decades, various SLAM approaches have been proposed to address the consistency problem. Especially for the Graph SLAM, which optimize the overall trajectories and sensor data to achieve a consistent global map. To increase the convergence of mapping, graph SLAM back-end has been proposed to minimize the accumulated error [103]. Based on that work, variants using dynamic covariance scaling [104] and switchable constraints [40] have been addressed.

However, it's still not yet possible to measure at what level the consistency of the map generated. In [105], an environment measurement model is proposed to verify the accuracy of transformation by calculating the percentage of inliers. In [75], the author analyses the sources of error for ICP and proposes a closed-form algorithm to estimate the covariance of ICP. In statistics, the paper models the histogram of the misalignment error metric into a truncated Gaussian distribution to evaluate the

Figure 2.7: An example of the inconsistency map, where the inconsistency is denoted in red.

consistency in [106]. The research of identifying whether a map is consistent has not been extensively studied except the works proposed in [107] [108] [109]. In [107], the author applies a Bayesian representation of map posterior to revise the past wrong data association. Then it is extended to a multi-scan scenario with a cascaded map inconsistency statistical test in [108]. Based on that work, [109] developed a robot homing strategy whenever the map is detected to be inconsistent. These works pioneer the map inconsistency detection problem, and the use of this knowledge to improve autonomous mapping missions. However, these methods focus on sensor data level in a 2D environment with a single robot, which is infeasible to be extended to a multi-robot scenario in full 3D mapping missions.

Since it's reasonable to assume the submap is locally consistent [110], submap-based mapping strategy shows its superiority. In large environment, submap-based approach provides incremental [111] and efficient [112] approach for robot mapping. For multiple robots coordination [71], submap is even more suitable under the constraints of limited communication bandwidth and coverage. As submaps are

Figure 2.8: Merging two features.

generated over a short motion window of the robot, it is reasonable to assume that they are locally consistent [110]. To the extent of our knowledge, this is the first application of robust submap-based inconsistency detection for multi-robot mapping.

2.3.2 Probabilistic Information Fusion

To distribute data among robots, a serial sequential strategy to communicate with robots is proposed in [17]. In DDF-SAM [13], a robust distributed communication framework is proposed, however solely provide simulations results under simplifying assumptions. In [113], the author proposed a distributed scheme for multiple data-gathering robots under limited communication and buffer size constraints. In [21], proposed an efficient and robust algorithm, an offboard ground station takes sensor data from all MAVs and fuses them into a detailed map. In [113], the author proposed a distributed scheme for multiple data-gathering robots under limited communication and buffer size constraints. In [114] [115], the author presents a calibration method based on raw sensor data.

When a robust and accurate transformation is calculated when the map is received, the source map is transformed to the coordination frame of the target map by a transformation function. Then, the key issue is to integrate the probabilistic in-

formation in partial maps into a global consistent map. In many of the previous methods, once the transformations between map coordinate frames are estimated, the final maps are generated by simply stitching the overlapped data [22] or averaging the occupancy probability value of corresponding pairs on voxel-wise level [70], where uncertainty propagation and merging is ignored. To measure the uncertainty of probabilistic map, Shannon entropy is used to measure the uncertainty of map in robot exploration in [116]. In [24], the paper introduces an entropy filter to reject fusion leading to a higher entropy, but is overconfident on the single map without taking both maps into consideration. Since the merging process should not introduce inconsistency for the global map, the Kullback-Leibler(KL) divergence based relative entropy filter is adopted to fuse occupancy probability consistently in [25].

In general, there are two types of uncertainty need to be integrated when performing map merging, which are uncertainty embedded in the map and uncertainty caused by the the relative transformation. To propagate the transformation uncertainty, the covariance of transformation function is derived in [22]. In [75], the author proposed an closed-form estimate of ICP's covariance. Then, the author calculates the covariance of transformed point by deriving the Jacobian of the transformation function in [24]. As for the map uncertainty, it can be represented by the occupancy probability in [65]. Given the uncertainty generated, the probability distribution should be merged to update the covariance. In [12], two 2D Gaussian distributions are multiplied and results in a compounded Gaussian distribution. Alternatively, the maps are fused on the voxel-wise level by directly taking the average of the probabilities in [70]. In [25], the relative transformation is evaluated based on a probabilistic measurement model and map dissimilarities are combined with relative entropy filter. Aforementioned approaches either assumes accurate transformation and only considers merging map uncertainty [70] [25] [12], or ignores the map uncertainty and focus on transformation uncertainty [24] [22]. In addition, none of those approaches tightly couple the map matching output with map merging, where two sub-problems are solved separately.

2.4 Conclusions

This chapter presents the relevant background of collaborative mapping and gives an extensive survey. The aspects of collaborative mapping approaches which include dense registration, map matching (homogeneous/heterogeneous), map sharing, inconsistency detection and information integration are carefully examined. Although, tremendous success have been made, a systematic collaborative mapping framework is still not studied in depth, especially in unstructured environments with heterogeneous sensors. Overall, most of the approaches just address the collaborative mapping from a specific viewpoint. However, they have not provided a comprehensive analysis of the systematic probabilistic fusion framework on feature extraction, map uncertainty modeling, probabilistic map matching, transformation evaluation and accurate map merging. By a close inspection it can be seen that the map matching approaches either falls to the scope of raw sensor data registration, or fails to model the nature of the 3D occupancy map. In addition, a full map merging process should consist of transformation evaluation, uncertainty derivation and information integration. Few of previous approaches address this issue in an adequate manner and most of the focus is given to integration of the information. The common thread of the following chapters is to develop a systematic probabilistic fusion framework for collaborative robots 3D mapping that can address these challenges.

Chapter 3

Probabilistic Point Set

Registration Approach for Fusion of 3D Occupancy Grid Maps

3.1 Introduction

With the continuing development in the field of mobile robotics and SLAM, in particular, the use of mobile robots to autonomously explore and map an environment using multiple robots has caught the attention of many robotics researchers and practitioners. Coordination among multiple robots [53] enable them to perform difficult tasks in challenging environments more efficiently and reliably, such as search and rescue missions.

One of the major challenges in such missions is to fuse the data generated by individual robots, which can be performed at different levels. Under the condition of limited communication and memory, a compact map generated by compressing raw sensor data is preferred. Hence, it's vital to fuse the partial maps generated by individual robots into a more consistent global map, which is further compounded

with 3D volumetric maps where the map fusion algorithms have to process large amounts of 3D map data.

The research in map fusion has not been studied extensively, especially for 3D volumetric maps. Majority of the existing approaches focus on feature matching and topology matching for 2D maps. As the availability of high-quality 3D sensors increase, Octomap based on Octree data structure has been widely used to model the environment with a full probabilistic 3D model [65]. In order to get an accurate and systematic map fusion algorithm, two requirements are set.

Relative Transformation

Since the thesis is working on the compressed map level, all the characteristics of the map should be utilized to compute an accurate transformation. In addition, point-wise correspondence should be provided for probability fusion.

Probability Fusion

Many of the existing researches pay much attention to compute relative transformation, however place less emphasis on how to fuse the maps. Due to different view and SLAM error, individual maps will have varying probability even for the same object. To fuse the maps into a global one, dissimilarities between individual maps should be measured to fuse the maps and decrease the total uncertainty.

This chapter addresses the above requirements and presents a novel method with the following contributions:

Firstly, probabilistic and surface information are incorporated in the registration algorithm, which is called Occupancy Iterative Closet Point (OICP). OICP combines the strength of ICP with the associated probabilistic and surface information of 3D occupancy map, which tends to match voxels with similar occupancy probability.

Secondly, an environment measurement model (EMM) is applied to decide whether to accept or reject the computed relative transformation. Then a relative entropy filter is proposed to fuse the map by combining occupancy probability from both maps which also decreases the overall global map uncertainty.

The chapter is organized as follows: Section II introduces the Occupancy Iterative Closet Point algorithm. Section III details the evaluation of the computed relative transformation and the fusion of probabilities. Section IV presents the outcome of the experiments conducted using simulated and real data. Section V concludes the chapter and discusses the future work.

3.2 OICP Algorithm

In this section, occupancy iterative closet point(OICP) is presented to incorporate occupancy probability and surface information of 3D occupancy map, which computes the relative transformation between partial maps under the condition of given consistent partial occupancy maps and initial position estimation.

Figure 3.1: An example of Octomap and its related point cloud

Figure 3.2: Sensor model of Octomap

Fig. 3.1 illustrates an example of Octomap and the associated point cloud extracted from the map. Octomap is a three-dimensional voxel map based on octrees, providing a volumetric representation of space which is a compact, probabilistic, efficient model for a variety of robotic applications including flying robots and ground robots. The measurements are integrated using a sensor model to provide the probability of grid. Then a grid cell probability is updated using a Bayesian process [117]. A sensor model of a single measurement with the probability distribution is shown in 3.2.

Given an Octomap M_r , the point cloud m_r extracted from Octomap is defined as

$$m_r = (m_r^1, m_r^2, \dots, m_r^i, \dots)$$

where $(m_{rx}^i, m_{ry}^i, m_{rz}^i, L_{m_r}^i, d_{m_r})$ is a tuple that includes

$$m_r = \begin{cases} (m_{rx}^i, m_{ry}^i, m_{rz}^i) & \text{position of the extracted voxel center} \\ L_{m_r}^i & \text{the occupancy probability log-likelihood} \\ d_{m_r} & \text{The size of the voxel in Octree} \end{cases}$$

In the registration process, the source point cloud will be transformed by matrix T iteratively to minimize the error metric with target point cloud, while target point

cloud remains fixed. Here, point cloud m_r is assigned as target point cloud and m_s as source point cloud. The transformation matrix T , consisting of rotation matrix R and translation matrix t . The transformation matrix T can be written as the form of affine matrices in 3.1.

$$T = \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0 & I_{1 \times 1} \end{bmatrix} \quad (3.1)$$

The OICP can be divided into two stages. For the first stage, point-to-plane ICP is applied to make use of the smooth surface information of Octomap using 3.2, where η is the surface normal of target point cloud m_r .

$$f = \min_T \sum_{i,j} ||(Rm_s^j + t - m_r^i) \cdot \eta||^2 \quad (3.2)$$

In the second stage, voxel occupancy probability is incorporated in order to achieve accurate transformation with point-wise correspondence. The iterative registration using invariant features is proposed in [91], it is proven that a weighted linear combination of positional and feature distances will converge if the weight parameter is non-increasing. The work is proved in a common framework. Specifically, at each time step of map fusion, the occupancy is regarded as an invariant feature. Hence, the weighted linear combination is utilized to combine positional distance and feature distance into a single point-wise distance metric, where each point is represented by its three positional coordinates and the attached one dimensional occupancy probability coordinate. Points are matched in this 4 dimensional space.

The point-wise distance is described by the summation of Euclidean distance $d_e(m_r^i, m_s^j)$ and feature distance $d_o(p(m_r^i), p(m_s^j))$. The weight parameter ω_o is defined to balance the contribution between positions and features. The general form for the distance

metric is defined in 3.3.

$$d = d_e(m_r^i, m_s^j) + \omega_o d_o(p_{m_r^i}, p_{m_s^j}) \quad (3.3)$$

For Euclidean distance, the classical ICP is applied to calculate the relative transformation by minimizing point-wise Euclidean distance using 3.4. ICP iteratively revises the transformation needed to minimize the distance from the source point cloud m to the target point cloud n . ICP will be stop when a defined threshold is achieved, for example, the maximum number of iterations.

$$d_e = ||Rm_s^j + t - m_r^i||^2 \quad (3.4)$$

For feature distance, the occupancy probability is set as an invariant feature. The probability of voxel m_r^i given observation $z_{1:t}^i$ is updated using log-odds notation in Octomap [33]. Although occupancy probability will recursively update when new sensor measurements are observed, it can be viewed as invariant features at each time step.

$$p(m_r^i | z_{1:t}^i) = \left(1 + \frac{1 - p(m_r^i | z_t^i)}{p(m_r^i | z_t^i)} \frac{1 - p(m_r^i | z_{1:t-1}^i)}{p(m_r^i | z_{1:t-1}^i)} \frac{p(m_r^i)}{1 - p(m_r^i)} \right)^{-1} \quad (3.5)$$

Using log-odds notation, equation 3.5 can be simplified to

$$L(m_r^i | z_{1:t}^i) = L(m_r^i | z_{1:t-1}^i) + L(m_r^i | z_t^i) \quad (3.6)$$

Note that log-odds can be easily converted to probabilities using 3.7.

$$L(m_r^i | z_{1:t}^i) = \log \frac{p(m_r^i | z_{1:t}^i)}{1 - p(m_r^i | z_{1:t}^i)} \quad (3.7)$$

Then, the feature distance is defined as:

$$d_o = ||p_{m_r^i} - p_{m_s^j}||^2 \quad (3.8)$$

Combining Euclidean distance and feature distance, occupancy iterative closet point is formulated in 3.9. Here the Euclidean distance is defined as the distance between center points of the voxel, and feature distance is represented by the difference in associated occupancy probabilities.

$$f_{oicp} = \min_T \sum_{i,j} ||Rm_s^j + t - m_r^i||^2 + \omega_o ||p_{m_s^j} - p_{m_r^i}||^2 \quad (3.9)$$

The OICP iteratively computes the transformation which minimizes the proposed error metric between corresponding points. And the trade off between positional and feature value is computed by weight parameter ω_o . The weight parameter is a scale factor to determine the contribution of positional and probability value. To model this weight, the uncertainty of positional error caused by mismatching and probability error caused by sensor noised are estimated. The OICP algorithm can be summarized as follows:

3.2.1 Uncertainty in Occupancy Probability

As Octomap is generated using noisy pose estimates provided by SLAM outputs and noisy sensor data, there is always an inherent level of uncertainty associated with them. When the uncertainty of the map is high, less confident is given on the map. On the contrary, when the uncertainty of the map is low, more weight is assigned to the map. Therefore, Shannon entropy is introduced to measure the uncertainty of the probabilistic voxel m_r^i using 3.10 . $H(m_r^i)$ is defined to represent the entropy of probabilistic voxel m_r^i .

$$H(m_r^i) = -p_{m_r^i} \log_2 p_{m_r^i} \quad (3.10)$$

Figure 3.3: The value of shannon entropy with the change of probability value

The occupancy probability of $p_{m_r^i}$ is within the range of $[0, 1]$, where

$$p_{m_r^i} \begin{cases} \in [0, 0.5) & \text{the voxel is free} \\ = 0.5 & \text{unknown} \\ \in (0.5, 1] & \text{the voxel the occupied} \end{cases}$$

The higher the probability, the more certainty the voxel is occupied. The lower probability, the more certainty the voxel is free. Here, the situation of voxel is occupied, which means $p_{m_r^i} \in (0.5, 1]$. The value of $H(m_r^i)$ with the change of $p_{m_r^i} \in (0.5, 1]$ is shown in Fig.3.3, which is a monotonic decreasing function. More specifically,

$$\begin{cases} H(m_r^i)_{min} = 0 & \text{when } p_{m_r^i} = 1 \\ H(m_r^i)_{max} = 0.5 & \text{when } p_{m_r^i} = 0.5 \end{cases}$$

When a voxel is occupied certainly with $p_{m_r^i} = 1$, the entropy achieves minimum with 0. When the minimum value is achieved, it means the voxel is occupied certainly with no uncertainty. On the contrary, entropy reaches maximum value with 1 when the occupancy probability $p_{m_r^i} = 0.5$. When $p_{m_r^i} = 0.5$, it means it does not have

any information of the voxel, the probability of the voxel is free or occupied is the same.

Then, the overall map uncertainty $H(m_r)$ is represented as the summation of probabilistic voxels:

$$H(m_r) = \frac{1}{n} \sum_{i=1}^n H(m_r^i) \quad (3.11)$$

3.2.2 Uncertainty in Positional Value

For registration, there is always a misalignment error between positional coordinates at each step. The misalignment error is hard to model due to the unknown amount of mismatch point between target scene and source scene. For the purpose of quantifying the error, it is assumed to be a Gaussian noise model.

As shown in Fig. 3.4, let d_{TS} be the true distance between target point m_r^i and source point n_s . At each iteration, point m_s^j iteratively attempts to find its ground truth point m_r^i , it would match to its nearest neighboring point m_r^c with a distance d_{CS} . So, there is always an offset between ground truth distance d_{TS} and the closet point distance d_{CS} .

In [91], conditional expectation is applied to model the error due to misalignment in each iteration. The author assumes that the points around the ground truth point m_r^i are continuously distributed. However, the points are discretized with equal distance d_{m_i} in our case.

To model the true distance d_{TS} between point m_r^i and point m_s^j using closet point distance d_{CS} , the registration error is assumed to be independent of noise and equal in three directions.

$$E(d_{TS}^2) = \sigma_{TS}^2 = \frac{1}{3}\sigma_{TSx}^2 = \frac{1}{3}\sigma_{TSy}^2 = \frac{1}{3}\sigma_{TSz}^2 \quad (3.12)$$

$$f(d_{CS}^2|d_{TS}) = \frac{1}{4\pi d_{TS}^2} \quad (3.13)$$
$$\begin{aligned}
E(d_{CS}^2|d_{TS}) &= \int_S d_{CS}^2 \cdot f(d_{CS}^2|d_{TS}) ds \\
&= \int_S ((d_{TS} \cos \Phi)^2 + d_E^2) \cdot \frac{1}{4\pi d_{TS}^2} ds \\
&\approx \frac{1}{3} d_{TS}^2
\end{aligned} \tag{3.14}$$

In conditional expectation, $E[E(X|Y)] = E(X)$, so the expectation of d_{TS}^2 can be calculated as:

$$E(d_{TS}^2) \approx E(3E(d_{CS}^2|d_{TS}) - 2d_m^2) = 3d_{CS}^2 \quad (3.15)$$

The term $-2d_m^2$ indicates the error caused by discretized point cloud. Since mean square error (MSE) is the global estimate of d_{CS} , $d_{CS} \approx MSE$. As each dimension is identical, $\sigma_{TSx}^2 \approx MSE^2$.

Hence, we get the final form for the weight as:

$$\omega_o = k \cdot \frac{\sigma_{TSx}}{H} \approx k \cdot \frac{MSE}{H_{m_r}} \quad (3.16)$$

where k is positive constant factor. As the registration iteratively minimizes the error function, MSE should be non-increasing. Then weight ω_o is non-increasing with invariant H , and should tend to zero as the OICP converges to a perfect match. When the positional error is large, more weight is given to features. As the two maps get closer, the algorithm tends to rely more on positional value.

3.3 Transformation Evaluation and Probability Fusion

The transformation matrix T transforms source map m_s from local coordinate frame into an target map m_r coordinate system. Therefore, the voxel $m_s^j \in m_s$ is transformed to the coordinate frame of m_r by the transformation matrix T , which is

$${}^r m_s^j = T \cdot m_s^j \quad (3.17)$$

In theory, the voxel ${}^r m_s^j$ should coincide with its closet voxel m_r^i in map m_r . Due to the sensor noise in map and the misalignment error caused by OICP, an error between the corresponding pairs is always expected. Hence, a transformation evaluation criteria is proposed to measure the misalignment and further fuse the map with a relative entropy filter.

3.3.1 Transformation Evaluation

Although the registration algorithm will achieve convergence, there is no guarantee that a global minimum will be found and there is a high chance that the algorithms would fall into a local minimum.

In [105], an environment measurement model (EMM) is proposed to verify the transformation by calculating the Mahalanobis distance between matched points. Then the inliers percentage is set as a threshold to decide whether to accept or reject the transformation. For the statistic test, the null hypothesis assumption is that after performing the transformation, the individual maps should be aligned correctly. Let's consider the matching between voxels under mahalanobis distance. An example of the registration error can be shown in Fig.3.5. The center represented by circle and square are two features. For each pair of such matching features, a distance should be calculated to determine if the two voxels actually represent the same object.

Then Mahalanobis distance between point m_r^i and ${}^r m_s^j$ can be written as:

$$f(m_r^i, {}^r m_s^j) = f(m_r^i; {}^r m_s^j, \Sigma_{ij}) \quad (3.18)$$

When operating on the Octomap representation, the sensor noise has already been incorporated into the occupancy probability. Hence, we utilize the inverse of occupancy probability to represent the spherical covariance of a voxel.

Figure 3.5: An example of the registration error between matched features

Since the covariance is equal in three directions, rotation matrix has no impact on Σ_i and Σ_j

$$\Sigma_{ij} = R^T \Sigma_j R + \Sigma_i = \frac{1}{p_{m_r^i}} + \frac{1}{p_{m_s^j}} \quad (3.19)$$

Then for each correlated pairs, Mahalanobis distance can be computed as follows, where d is a normalizing factor:

$$f_{m_{ij}} = -\frac{({}^r m_s^j - m_r^i)^T \cdot ({}^r m_s^j - m_r^i)}{d \cdot (p_{m_r^i}^{-1} + p_{m_s^j}^{-1})} \quad (3.20)$$

Note that the covariance defined here is only a scale factor and reveals that Euclidean distance is a special case of Mahalanobis distance. That covariance equals to identity matrix, when $p_{m_r^i} = p_{m_s^j} = 1$. Then 3.20 becomes

$$f = ||{}^r m_s^j - m_r^i||^2 \quad (3.21)$$

The inliers I are decided by setting a threshold of Mahalanobis distance, and the rest are counted as outliers O . Then a confidence level is applied as a criteria for

rejecting the wrong estimates by computing the overall fraction of inliers in 3.22.

$$q = \frac{I}{I + O} \quad (3.22)$$

It is shown in the experiments that the applying of environment measurement model(EMM) effectively rejects high erroneous estimates that would largely undermine the quality of the fused map.

3.3.2 Relative Entropy Filter

Under sensor noise and motion uncertainty, the same object may have different probabilities in separated maps. This is called duplicate landmarks in feature map. However, in voxel map, the features and objects have been discretized into many voxels. It's hard count how many voxels consisting of an object. However, its vital to establish a strategy to consider the dissimilarities and fuse the voxels into the global probability map.

To merge pair-wise voxels, the separation level between the voxels has to be measured and evaluated. The test is based on Kullback-Leibler(KL) divergence, which is a commonly used relative entropy filter to measure the difference between two probability. Here the occupancy probability $p_{m_r^i}$ and $p_{r_{m_s^j}}$ are modeled as discrete random variables(RVs), and the difference between the two RVs are computed as:

$$KL(p_{m_r^i} || p_{r_{m_s^j}}) = p_{m_r^i} \log \frac{p_{m_r^i}}{p_{r_{m_s^j}}} + (1 - p_{m_r^i}) \log \frac{1 - p_{m_r^i}}{1 - p_{r_{m_s^j}}} \quad (3.23)$$

$KL(p_{m_r^i} || p_{r_{m_s^j}})$ represents the difference between the two probabilities.

If KL is within a certain threshold, the dissimilarity are modeled as random errors and still fuse the corresponding pairs by taking the average

$$p_{fuse}^i = \frac{p_{m_r^i} + p_{r_{m_s^j}}}{2} \quad (3.24)$$

To update the map, the the corresponding pairs are deleted and replaced by the newly fused voxel.

If KL is beyond the threshold with $p_{m_r^i} > p_{r_{m_s^j}}$, it means the voxel with lower probability $p_{r_{m_s^j}}$ doesn't have enough information for the environment and will be rejected, then we only trust the voxel with higher probability and let $p_{fuse}^i = p_{m_r^i}$.

The overall algorithm for transformation evaluation and probability fusion is summarized as follows:

Algorithm 1 Transformation Evaluation and Probability Fusion

Require: transformation matrix T , map m_r and m_s

Ensure: the local maps can be fused into a global map

1. $\Sigma_{ij} = R^T \Sigma_j R + \Sigma_i$
 2. Calculate mahalanobis distance $f_{m_{ij}}$
 3. Calculate inlier percentage q
- if** $q > q_{thres}$ **then**
transformation is accepted, **continue**
if $KL(p_{m_r^i} || p_{r_{m_s^j}}) < KL_{thres}$ **then**
 $p_{fuse}^i = \frac{p_{m_r^i} + p_{r_{m_s^j}}}{2}$
else if $p_{m_r^i} > p_{r_{m_s^j}}$ **then**
 $p_{fuse}^i = p_{m_r^i}$
else
 $p_{fuse}^i = p_{r_{m_s^j}}$
end if
-

3.4 Experimental Results

Experiments conducted using both simulated and real data are presented in this section. For simulation, a robot equipped with a simulated 3D depth sensor traversed the environment twice to create two maps in Gazebo simulator [118] using RTAB [119]. A Pioneer 3 AT robot was tele-operated twice in the WKW lobby at Nanyang Technological University to build two maps. The robot was equipped with a Hokuyo Laser range finder for pose estimation using Gmapping [120] and an Asus Xtion PRO for 3D perceptions, as shown in Fig. 3.6. The individually generated

Table 3.1: The summary of setting up of the experiments

Type	Environment	SLAM Method	Sensor	Voxel Size
Simulation	Gazebo simulator	RTAB	Asus Xtion	0.1m
Real world Experiment	WKW lobby	GMAPPING	Asus Xtion	0.1m

(a) Pioneer robot

(b) Partial map generated from WKW Lobby

Figure 3.6: Pioneer robot and partial map generated from WKW Lobby

(a) Simulation

(b) WKW Lobby

Figure 3.7: Initial transformation for point cloud generated from Octomap

Octomaps were post-processed using C++ and Matlab to generate the fused maps. The size of Octomaps in both experiments was set to be $0.1m$. A summary of the setting up can be shown in Tab.3.1

As mentioned before, the focus of this chapter is on the fusion stage with an initial relative transform estimation. To simulate the availability of noisy initial relative transformations, two maps were generated with different initial transformations both in position and rotation. Fig. 3.7 illustrates the point clouds extracted from the generated individual Octomaps with an erroneous initial transformation. The dimension was about $50m * 30m * 10m$ for both maps.

3.4.1 Registration Results

In this section, the performance of OICP compared to ICP, point-to-plane ICP and normal distribution transform(NDT) is presented. As a baseline for comparison, standard ICP, point-to-plane ICP and NDT implementations from the Point Cloud Library (PCL) were used [121] and parameters were set according to the environments. The number of iterations for ICP, point-to-plane ICP and NDT is set to be 50. As mentioned before, the point clouds extracted from Octomap has a smoother local planes compared to raw sensor data. For OICP, point-to-plane ICP was performed for 15 iterations as initial registration stage to make use of this surface information, then ICP with occupancy probability formulated in Chapter.3 was performed 35 iterations to get the point-wise correspondence.

The comparison of accuracy of registration algorithms is summarized in Fig. 3.8. The translation error and rotation error are the Euclidean norm of and Euler norm of the difference between the ground-truth and the output.

OICP outperforms NDT, point-to-plane ICP, as well as map registration algorithm ICP used in [70]. Although point-to-plane ICP unable to provide point-wise correspondence, it makes use of the smooth plane and achieves better accuracy compared to default ICP based fusion. However, point-to-plane ICP still has drifts in translation and rotation. OICP combines surface and probability information to compute the point-wise correspondence with similar probability to generate the most accu-

Figure 3.8: Mean Error in both rotation and translation for the experiments, blue color represents the results conducted in simulation, while the results conducted in WKW lobby is shown in red

rate relative transformation. We should make special notice to NDT, which produces large errors, especially in translation. The reason might be the Gaussian distribution extracted from discretized sparse point cloud can't represent the environment explicitly compared to rich sensor data.

As described before, point-to-plane ICP and NDT can't provide point-wise correspondence. Hence only ICP and OICP are discussed in the following experiments. The registered results are shown in Figs. 3.9 and 3.10. A large offset in translation is shown for the result of ICP, and OICP produces approximately a perfect alignment. For both results, rotation errors are considered to be relatively accurate.

3.4.2 Transformation evaluation

After computing a transformation, the accuracy is evaluated by calculating the percentage of inliers. Since ICP is deterministic with only coordinates, the covariance of each point is set to be identity matrix $I_{3 \times 3}$. For map fusion, a strict criteria is set to guarantee the accuracy: the inliers threshold is set as 0.15 Mahalanobis distance and the fraction of rejecting a transformation is set to be below 60%. Fig. 3.11

Figure 3.9: Registration result of WKW Lobby

Figure 3.10: Registration result of simulated environment

illustrates the comparison of the average Mahalanobis distances between OICP and ICP in the two experiment settings. The box plot shows a box that encloses the middle 50% of the data, with the median value marked by a line and the mean value marked by a small square.

The inlier fraction based on the threshold is shown in Fig. 3.12. The inlier percentage of ICP is lower than OICP because ICP didn't converge to perfect alignment. Based on the selected threshold, OICP transformation generates an accurate and reliable map registration result compared to pure ICP based map registration approach in [70].

Figure 3.11: Average Mahalanobis Distance

Figure 3.12: The percentage of inliers

Table 3.2: The entropy after map fusion

	Taking The Average	Relative Entropy Filter
Simulation	0.16	0.13
WKW Lobby	0.24	0.2

3.4.3 Probabilistic Map Fusion

After determining the correct transformation for map registration, a relative entropy filter is used to achieve map fusion by considering the dissimilarities between matched voxels. The relative entropy threshold is set as 0.1 in KL-divergence. In comparison, the fusion strategy in [70] is performed by directly taking the average of probability for all the matched voxels. In simulated environment, the entropy for map fusion by directly taking the average is 0.16. And the entropy of using relative entropy filter is 0.13. For real experiment in WKW Lobby environment, the entropy of directly taking the average is 0.24, and is 0.2 by using relative entropy filter. The results are summarized in Tab.3.2.

The entropy of WKW Lobby is higher than simulation due to real world noise and uncertainty. The experiment indicates that our map fusion strategy is able to combine probabilities of both maps effectively to further decrease the total uncertainty. Figs. 3.13 and 3.14 show the final results of the fused 3D occupancy grid maps of

Figure 3.13: The fused map of simulated environment

Figure 3.14: The fused map of WKW Lobby

simulation and WKW Lobby.

3.5 Conclusions

Most of the existing map matching algorithms ignore the probability information in the map. In this chapter, a probabilistic 3D map fusion algorithm Occupancy iterative closet point(OICP) was proposed. The OICP algorithm combines occupancy probability in conjunction with surface information to register point cloud extracted from Octomap. The weight between positional error and map uncertainty is controlled by modeling errors using Shannon entropy and positional misalignment. EMM is applied to evaluate the quality of transformation by rejecting low inlier fraction. The final map fusion is then achieved with a relative entropy filter which integrates the measurements of both maps and decreases the uncertainty of the global map. Experiments on simulated and real world environments were conducted and the evaluations confirmed that the proposed OCIP method is able to produce more consistent global 3D occupancy grid maps compared to existing methods in the literatures.

This chapter is based on the assumption that the local maps generated by single robot is consistent. However, this assumption may not hold true in real applications.

In Chapter 4, inconsistency detection on both single robot and multi-robot levels will be presented. In addition, OICP ignores the structure information in the map and only relies on the point cloud extracted. In Chapter 5, a two-level probabilistic map matching algorithm is proposed to combine structure edge information and individual voxel information.

Chapter 4

Robust Submap-Based Probabilistic Inconsistency Detection for Multi-Robot Mapping

4.1 Introduction

In Chapter 3, the point set registration approach for fusion of 3D occupancy grid maps is studied. In addition, an environment measurement model (EMM) is applied to detect the inconsistency. In collaborative mapping missions, global map generation is generally achieved by fusing the local maps generated by individual robots [25]. Fusing erroneous local maps results in an inconsistent global map, which renders the accurate execution of autonomous tasks infeasible. Therefore detecting inconsistency in both local maps and fused global maps is vital for multi-robot missions. The development of map inconsistency detection strategies has not received much attention, where the majority of the research is focused on generating accurate

Figure 4.1: An example of consistent and inconsistent map

maps and post-processed optimization. However, map generated does not guarantee to be foolproof and may generate inconsistent maps, especially in large environments and over long operating times. In addition, post-processed optimization is computationally expensive, where a proper triggering time and quantitative inconsistency measurements are required. Hence, there is a gap between the map generation and the map optimization, which is the inconsistency detection of the map. The capability to detect inconsistency in the generated maps allows robots either to perform an efficient optimization algorithm in order to continue the mission, or to return to a safe location for recovery without getting trapped.

This chapter presents a novel strategy that detects the inconsistency of map generated from two levels, i.e., single robot and multi-robot level. The inconsistency of the system is modeled as a probabilistic distribution, where submap-based inconsistency testing is performed on three types of inconsistency defined on both single robot and multi-robot level. In both levels, map inconsistency measurements are computed by modeling the misalignment errors between submap pairs into a truncated Gaussian distribution, and the resultant mean and variance are used to detect the inconsistency. Besides, the detected inconsistency is further validated through a 3D map registration process. The fused global map is deemed consistent only if all submap pairs of all robots pass the inconsistency checks.

Main contributions of this work are listed below:

- A novel multi-level topological framework is proposed to model the inconsistency of full 3D maps on both single robot and multi-robot level.
- A submap-based probabilistic algorithm is developed to efficiently detect and verify the three types of inconsistency (i.e., sequential, local and global).

The rest of this chapter is organized as follows: Section II presents the topological probabilistic structure. Section III details submap-based inconsistency modeling and detection. Section IV presents the experiments. Section V concludes the chapter.

4.2 Topological Structure of Map Inconsistency Detection

This section presents the structure of submap-based inconsistency measurement, which is shown in Fig. 4.2. The submap-based inconsistency measurement is defined as a topological structure, where the node is defined as the submap and the edge is defined as the submap-wise inconsistency. Then it further explains the three types of inconsistency on single robot and multi-robot levels.

4.2.1 Definition of Node

The node is defined as a submap and notation is denoted in Eq.(4.1)

$${}_i^x V = \{{}_i^x m\}_{i \in n_x, x \in \mathcal{R}} \quad (4.1)$$

where n_x stands for the set of known submaps generated by robot x , and $\mathcal{R}$ represents the set of robots $\{a, b, c, \dots\}$. Submap $\{{}_i^x m\}$ is generated when certain conditions have met and is added to the topology as a new node. In this chapter, a submap is represented as a 3D probabilistic occupancy voxel map [65]. Considering the

Figure 4.2: Topological structure of map inconsistency detection.

limited size of submap will lead to less drift, while still providing sufficient submap size for map registration. The criterion that triggers the creation of a new submap is based on the condition that the robot moved for a certain distance or rotated for a certain angle. Based on submaps generated, the local map and global map on a higher level are defined below.

4.2.1.1 Local map generated by a single robot

Local map $^x m$ generated by a single robot x consists of submaps $\{^x_i m\}$ with $i \in n_x$, which is shown as the chain of node with the same color in Fig. 4.2. The nodes with different colors represents the submaps of different robots. The three types of edges show the inconsistency at different levels.

$$^x m = \{^x_i m, i \in n_x\} \quad (4.2)$$

4.2.1.2 Global map generated by multi-robot

Global map M that integrates 3D occupancy local maps ${}^x m$ with $x \in \mathcal{R}$ is defined as:

$$M = \{{}^x m, x \in \mathcal{R}\} \quad (4.3)$$

4.2.2 Definition of Consistency

Map inconsistency is defined as a measurement of alignment between the submaps. On that basis, consistency indicates that submaps have been registered properly, while inconsistency implies a large offset between two submaps. The inconsistency between two submaps is represented by the edge, which defines the inconsistency between two submaps on single robot level and multi-robot level.

4.2.2.1 single robot level

The submaps i, j within the local map ${}^a m$ connected by an edge ${}^i_j \Delta$ is denoted as single robot level inconsistency, which has two types: sequential inconsistency and local inconsistency.

Sequential inconsistency

Sequential inconsistency is defined as the probability between consecutive submaps ${}^a_i m$ and ${}^a_{i-1} m$ and is defined in Eq.(4.4). Sequential inconsistency always happens when two consecutive submaps didn't align properly due to SLAM error like a sharp turn occurs or a sudden drift in odometry sensor.

$${}^i_{i-1} \Delta = p({}^a_i m | {}^a_{i-1} m)_{\{i=2:n_a\}} \quad (4.4)$$

Local inconsistency

local inconsistency is defined in Eq.(4.5), which calculates the probability of ${}^a_i m$ over the past submaps ${}^a_j m$ generated by robot a .

$${}_j^i \Delta = p({}^a_i m | {}^a_j m)_{\{i=k+1:n_a, j=1:i-k\}} \quad (4.5)$$

The calculation is not performed from $i - k$ to $i - 1$ submaps because ${}^a_i m$ always has small overlapping with those submaps. Local inconsistency always happens when the robot in current submap ${}^a_i m$ didn't recognize the previously traveled submap ${}^a_j m$.

4.2.2.2 Multi-robot level

The local maps ${}^a m$, ${}^b m$ connected by an edge ${}_b^a \Lambda$ is denoted as multi-robot level inconsistency. Global inconsistency is defined as probability of submap ${}^b_j m$ in robot b over all the submaps ${}^a_i m$ in robot a , which is defined in Eq.(4.6). Global inconsistency always happens when two robots can not recognize the same place they visited.

$${}_b^a \Lambda = p({}^b m | {}^a m) \propto \prod_{i=1:n_a, j=1:n_b} p({}^b_j m | {}^a_i m) \quad (4.6)$$

4.2.2.3 Overall system

With the three types of inconsistency defined above, the inconsistency measurement for the whole system can be formulated. Here, the inconsistency of the system is

denoted in Eq.(4.7).

$$\begin{aligned}
& \prod_a \underbrace{p(\overset{a}{i}m|\overset{a}{j}m)}_{\text{single robot level}} \cdot \prod_{a,b} \underbrace{p(\overset{b}{j}m|\overset{a}{i}m)}_{\text{multi-robot level}} \\
&= \prod_a \left(\underbrace{p(\overset{a}{i}m|\overset{a}{i-1}m)}_{\text{pairwise consistency}} \underbrace{p(\overset{a}{i}m|\overset{a}{j}m)}_{\text{local consistency}} \right) \cdot \prod_{a,b} \underbrace{p(\overset{b}{j}m|\overset{a}{i}m)}_{\text{multi-robot level}}
\end{aligned} \tag{4.7}$$

Since the three types of inconsistency defined can be represented at submap-level, the inconsistency measurement can be modeled as a Gaussian distribution $f(x; \mu, \sigma^2)$.

4.3 Submap-Based Inconsistency Detection

In this section, the submap-based inconsistency measurement is calculated by modeling the histogram of inconsistency distance into a truncated Gaussian distribution. Then submap-based inconsistency testing is performed and verified by applying 3D map registration.

4.3.1 Submap-wise Inconsistency Distance

A proper inconsistency distance should be defined to describe the discrepancy between submaps. Here, $\overset{x}{i}m$ and $\overset{x}{j}m$ are two submaps. The edge connects $\overset{x}{i}m$ and $\overset{x}{j}m$ can be any of the three types of inconsistency. The distance applied here is the Occupancy Iterative Closet Point(OICP) distance defined in [25]. The OICP distance is an error metric specially designed for 3D occupancy grid map that combines the Euclidean distance $d_e(m_i, n_j)$ and occupancy probability distance $d_f(m_i, n_j)$. The inconsistency distance between a pair-wise matched voxels is defined in Eq.(4.8).

(a) The pair-wise submaps are misaligned with an error (b) Uniform histogram is modeled into gaussian distribution

Figure 4.3: An example of misaligned submaps and the histogram of inconsistency distance

The details of OICP distance can be found in [25].

$$d(\overset{x}{m}_i, \overset{x}{m}_j) = d_e(\overset{x}{m}_i, \overset{x}{m}_j) + \omega_o d_o(\overset{x}{m}_i, \overset{x}{m}_j) \quad (4.8)$$

Where $d(\overset{x}{m}_i, \overset{x}{m}_j)$ is the OICP distance between two submaps. The histogram of OICP distance over all the matched voxels is computed to show the distribution of the inconsistency distances. Since inconsistency is defined as a probability distribution, the histogram is further modeled into probability distribution to describe the discrepancy between submaps.

4.3.2 Modeling of Probability Distribution

For simplification, the submaps $\overset{x}{m}_i$ and $\overset{x}{m}_j$ are assumed to be straight lines and the probability of each voxel equals to 1. Histograms of perfect alignment and misalignment are modeled into Gaussian distributions $f(x; \mu, \sigma^2)$.

4.3.2.1 Modeling of Perfect Alignment

Assuming two submaps are noise free and perfectly aligned, which means the distance metric $d(i^x m, j^x m)$ between all corresponding voxels equals to zero. Then the histogram is subject to a Dirac delta distribution, which is the limit the normal distribution when $\sigma^2 \rightarrow 0$.

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}}; \mu = 0, \sigma^2 \rightarrow 0, \quad (4.9)$$

However, in real environment the map will be noisy and alignment will have a distance residual, the σ^2 will always greater than zero.

4.3.2.2 Modeling of Misalignment

Considering the inconsistency situation for straight lines, the two lines will intersect at a point. Here, the two lines intersect at the start point, as shown in Fig. 4.3(a). The parallel lines are not considered since submap overlapping is a basic requirement for inconsistency detection. The histogram of distance distribution complies a uniform distribution and is shown in Fig. 4.3(b).

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b \\ 0 & \text{for } x \leq 0 \text{ or } x \geq b \end{cases} \quad (4.10)$$

The uniform histogram can also be modeled by a Gaussian distribution with mean μ and variance σ^2 as shown in Fig. 4.3(b).

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}; \mu = \frac{a+b}{2}, \sigma^2 = \frac{(b-a)^2}{12} \quad (4.11)$$

Based on the observations, the simplification reveals that μ and σ^2 can be used to

describe the discrepancy. In general, large μ and σ^2 indicate a large misalignment error.

4.3.3 Truncated Gaussian Distribution

As shown above, the histogram can be modeled into Gaussian distribution $X \sim N(\mu, \sigma^2)$. Since the distance is absolute value, the Gaussian distribution can be modified to truncated Gaussian distribution in the interval of $[a, b]$. Given a normal distribution $X \sim N(\mu, \sigma^2)$ with the probability density function(pdf) $\Phi(x)$ and cumulative density function(cdf) $\Psi(x)$. Then, $X \sim N(\mu, \sigma^2)$ conditioned on interval $a \leq x \leq b$ is derived:

$$f(x; \mu, \sigma^2; a, b) = f_t(x; \mu_t, \sigma_t^2) = \frac{\Phi(x)}{\Psi(b) - \Psi(a)} \quad (4.12)$$

Let $\alpha = \frac{a-\mu}{\sigma}$, $\beta = \frac{b-\mu}{\sigma}$ where truncated mean μ_t and truncated variance σ_t^2 are calculated as follows:

$$\mu_t = \mu + \sigma \frac{\Phi(\alpha) - \Phi(\beta)}{\Psi(\beta) - \Psi(\alpha)} \quad (4.13)$$

$$\sigma_t^2 = \sigma^2 \left[1 + \frac{\alpha\Phi(\alpha) - \beta\Phi(\beta)}{\Psi(\beta) - \Psi(\alpha)} - \left(\frac{\Phi(\alpha) - \Phi(\beta)}{\Psi(\beta) - \Psi(\alpha)} \right)^2 \right] \quad (4.14)$$

Let $a = 0$, as distance is always not less than 0, and $b = d$, where d is the maximum search distance of OICP registration algorithm. On the interval of $[0, d]$, μ_t describes the average misalignment distance, and σ_t^2 describes the degree of misalignment divergence.

Figure 4.4: The environment of the experiment

4.3.4 Inconsistency Testing

In this part, a test is performed to detect the three types of inconsistency. Given all the submaps $\{m_i^x\}_{i \in n_x, x \in \mathcal{R}}$, the overlapping between submaps is firstly tested. Then the histogram of misalignment distances is modeled into truncated Gaussian distribution. An $N \times N$ square matrix Γ is generated that contains Γ_{ij} value defined in Eq.(4.15), where $N = \sum_{x \in \mathcal{R}} n_x$ is the total number of submaps. Here, the sparsity of matrix Γ depends on the pairs of overlapping submaps.

$$\Gamma_{i,j} = \frac{1}{\mu_t \cdot \sigma_t^2} \quad (4.15)$$

Inconsistency testing is performed for each $\Gamma_{i,j}$, a large value of $\Gamma_{i,j}$ indicates consistency, while a low value implies that large misalignment compared with the perfect map alignment. A global map is consistent if all tests are successful. For inconsistent submaps, they share a common area, however, are not aligned properly. Hence, a novel map registration algorithm is applied [25] to align the inconsistent submaps and generate a $N \times N$ matrix Θ . By comparing matrix Γ with Θ , the large difference verifies the detected inconsistency. An overall algorithm is shown below.

Algorithm 2 Submap-Based Inconsistency Detection

Require: The generated submaps $\{^x m\}_{i \in n_x, x \in \mathcal{R}}$
Ensure: Detecting sequential, local and global inconsistency

```

for  $\{^x m\}_{i \in n_x, x \in \mathcal{R}}$  with overlapping pairs do
  for single robot level do
    Sequential inconsistency:  ${}^i_{i-1} \Delta = p({}^a m | {}^a_{i-1} m)_{\{i=2:n_a\}}$ 
    Local inconsistency:  ${}^i_j \Delta = p({}^a m | {}^a_j m)_{\{i=k+1:n_a, j=1:i-k\}}$ 
  end for
  for multi-robot level do
    Global inconsistency:  ${}^i_j \Lambda = p({}^b m | {}^a_i m)_{\{i=1:n_a, j=1:n_b\}}$ 
  end for
end for
Generating inconsistency matrix  $\Gamma_{N \times N}$ 
for  $\Gamma_{(i,j)} > 0$  do
  Generating matrix  $\Theta$  after performing map registration
end for
Verified inconsistency matrix  $\Gamma^v = \Theta - \Gamma$ 

```

4.4 Experimental Results

Experiments performed using two robots in indoor and outdoor environments are presented in this section. Two robots were tele-operated in both indoor and outdoor environment at Nanyang Technological University. The robot was equipped with a Hokuyo Laser range finder for pose estimation using Gmapping [120] and a Velodyne VLP-16 for 3D perceptions, as shown in Fig. 4.4. The individually generated 3D grid maps are post-processed using C++ and Matlab to analyze their inconsistency. Submaps are generated at 10m intervals. The resolution of the 3D occupancy grid map in both experiments is set to be 0.1m.

4.4.1 Indoor Environment

The ground truth map and the inconsistent map are shown in Fig. 4.5. For local map ${}^a m$ is consistent while the local map ${}^b m$ can't close the loop when it comes back to the starting point. The ground truth of the map was generated with a very high number of particles and by moving the robot extremely slowly and smoothly.

Figure 4.5: The ground truth and inconsistent map generated in indoor environment.

For experiments, the number of particles was decreased to the default value of 30. The two separated local maps are fused to generate the global map and the result is shown in Fig. 4.5(b). For local map a_m , 8 submaps were generated. And 9 submaps were generated for the local map b_m .

Square matrix $\Gamma_{17 \times 17}$ is generated to describe the inconsistency as shown in Fig. 4.6(a). As for single robot level inconsistency detection, $\Gamma_{[1:8,1:8]}(\Gamma_{aa})$ shows the inconsistency between submaps $a_{1:8}m$, and $\Gamma_{[9:17,9:17]}(\Gamma_{bb})$ shows the inconsistency between submaps $b_{1:9}m$. For multi-robot level inconsistency, $\Gamma_{[9:17,1:8]}(\Gamma_{ab})$ represents the global consistency between submaps $b_{1:9}m$ and submaps $a_{1:8}m$. The two map sessions started and ended at the same position and is marked with a star in Fig. 4.5(b).

Figure 4.6: The large difference of the color in the same pixel verifies the inconsistency.

Note that only sequential inconsistency detection is performed, so the matrix is not symmetric.

4.4.1.1 Single robot level

For local map am and bm , the sequential consistency $_{i-1}^i\Delta$ are accepted. As shown in matrix Γ , all the pixels representing sequential inconsistency is with high values. Local inconsistency for map am is also accepted, which means the local loop is closed. For example, the high value of $\Gamma_{(8,1)}$ and $\Gamma_{(7,1)}$ indicates last two submaps $^am_7, ^am_8$ aligned properly with first submap am_1 . However, the low value for $\Gamma_{(13:16,10)}$, $\Gamma_{(15,9)}$, and $\Gamma_{(17,9)}$ indicate the local inconsistency detected in bm . For example, $\Gamma_{(14,10)}$ shows the inconsistency between bm_6 and bm_2 . And $\Gamma_{(17,9)}$ shows the misalignment between first submap bm_1 and last submap bm_9 .

4.4.1.2 Multi-robot level

The measured global inconsistency is presented in lower left part in Fig. 4.6(a), and is based on the overlapping area between am and bm in Fig. 4.5(b). Pixel $\Gamma_{(9,1)}$ is

with a high value, because robot a and b started at the same position and aligned properly. Then $\Gamma_{(17,1)}$ and $\Gamma_{(16,2)}$ are with a low value, due to the misalignment between last two submaps b_9m , b_8m and start position of a_1m and a_2m . The low values of $\Gamma_{(16,8)}$ and $\Gamma_{(17,8)}$ also detects the inconsistency between submaps $({}^b_8m, {}^a_8m)$ and $({}^b_9m, {}^a_8m)$.

4.4.1.3 Map Registration

As mentioned before, inconsistency is caused by misalignment between submaps. Here, a map registration algorithm is applied to verify the potential inconsistency detected. Square matrix Θ is generated and is shown in Fig. 4.6(b), which shows that the previously inconsistent submaps have been aligned after performing submap registration.

At single robot level, the sequential inconsistency remains unchanged, which validated our measurements. For ${}^a m$, the local inconsistency is also validated. For ${}^b m$, the local inconsistency value of $\Theta_{(14,10)}$, $\Theta_{(15,10)}$, $\Theta_{(17,9)}$ have increased sharply, which indicates these pairs have been aligned after map registration. At multi-robot level, the global inconsistency value $\Theta_{(16,8)}$, $\Theta_{(17,1)}$ and $\Theta_{(17,8)}$ have all been detected with a big change. An example of $\Theta_{(17,8)}$ is shown in Fig. 4.7, which is a detailed explanation of the inconsistency between submap b_8m and a_8m .

4.4.1.4 Verified Inconsistency Matrix

The verified inconsistency matrix Γ_v is generated by identifying the large differences before and after map registration in corresponding consistency matrix Γ and Θ . As illustrated in Fig. 4.8, local inconsistency is verified in local map ${}^b m$ for $\Gamma_{(14,10)}^v$, $\Gamma_{(15,10)}^v$ and $\Gamma_{(17,9)}^v$. Global inconsistency between ${}^a m$ and ${}^b m$ is verified for $\Gamma_{(17,1)}^v$, $\Gamma_{(16,8)}^v$ and $\Gamma_{(17,8)}^v$. The shade of color indicates the confidence level of the inconsistency detected. Inconsistency detected in $\Gamma_{(13,10)}^v$, $\Gamma_{(16,10)}^v$ and $\Gamma_{(15,9)}^v$ have been

Figure 4.7: The top row shows the detected global inconsistency between submap b_m (blue color) and a_m (red color), while the bottom row verifies the inconsistency after performing map registration

rejected due to the slight change after performing map registration. This is usually caused by low overlapping areas.

4.4.2 Outdoor Environment

The outdoor experiment was performed in an open car park in NTU. As can be seen from Fig. 4.4(b), the environment was quite challenging with moving cars and people. Fig. 4.10 shows the inconsistent map and inconsistency detected. The start and end position are marked with stars. For local map a_m , 13 submaps were generated. And 17 submaps were generated for the local map b_m .

Since there is no local loop for the local maps a_m and b_m , only sequential inconsistency and global inconsistency is detected. The verified inconsistency matrix Γ^v is shown directly to describe the inconsistency in Fig. 4.9.

Figure 4.8: The matrix Γ^v shows the verified inconsistent pairs by identifying the difference between Fig.4.6(a) and Fig. 4.6(b) in indoor scenario.

Single robot Level

The inconsistency of robot a is shown in $\Gamma_{[1:13,1:13]}^v$. Pairwise inconsistency is detected in $\Gamma_{(3,2)}^v$, $\Gamma_{(6,5)}^v$ and $\Gamma_{(13,12)}^v$. For local map $^b m$, the inconsistency is shown in $\Gamma_{[14:30,14:30]}^v$ for submaps $^b m_{1:17}$. Sequential inconsistency is detected in $\Gamma_{(24,23)}^v$ and $\Gamma_{(29,28)}^v$. Three pairs of inconsistent submaps are marked with ellipses in Fig. 4.10.

Multi-Robot Level

$\Gamma_{[14:30,1:13]}^v$ represents the global inconsistency between submaps $^b_{1:17}$ and $^a_{1:13}$. Global inconsistency is detected in $\Gamma_{(30,1)}^v$, $\Gamma_{(30,2)}^v$ and $\Gamma_{(28,1)}^v$. In addition, $\Gamma_{(14,12)}^v$, $\Gamma_{(14,13)}^v$, $\Gamma_{(15,12)}^v$ and $\Gamma_{(15,13)}^v$ represents the inconsistency detected between $(^b_1 m, ^a_{12} m)$, $(^b_1 m, ^a_{13} m)$, $(^b_2 m, ^a_{12} m)$ and $(^b_2 m, ^a_{13} m)$. The global inconsistency detected is shown in the overlapping area between $^a m$ and $^b m$ in Fig. 4.10. Due to the unstructured environments and moving objects, the map is quite inconsistent. Our method is able to detect the inconsistency accurately.

4.5 Conclusions

In this chapter, the problem of measuring the inconsistency of maps generated by multi-robot mapping missions is addressed. As there is no available approaches,

we proposed a multi-robot map inconsistency detection strategy that evaluates the inconsistency on local and global levels. The inconsistency is measured by modeling the submap-wise misalignment error metric distribution into a truncated Gaussian distribution. In addition, a map registration algorithm is applied to verify the detected inconsistency. Our method successfully detects the inconsistency in the challenging indoor and outdoor environment. More importantly, as shown in Fig.(4.8) and Fig.(4.9), the inconsistency of different types(ie. sequential, local and global) are detected with a probability distribution. The proposed submap-based framework is demonstrated to reduce the problem complexity and to accurately detect inconsistency in the global maps.

The developed approach can be applied for robust multi-robot operation. Inconsistent maps generated during multi-robot mapping process can be detected and actions can be executed. For example, it can be the uncertainty of post-processed optimization or the signal of resetting mapping mission.

Chapter 5

Hierarchical Probabilistic Fusion Framework for Matching and Merging of 3D Occupancy Maps

5.1 Introduction

In Chapter 3, 3D volumetric map fusion algorithm occupancy iterative closet point (OICP) is proposed by extending the well known iterative closest point (ICP) algorithm. Algorithmically, it belongs to the scope of iterative dense registration algorithm [72, 77], which is designed for raw sensor data registration. Due to the intrinsic nature difference between raw sensor data and 3D occupancy grid maps, the direct use of dense registration algorithms in 3D occupancy grid map matching results in sub-optimal voxel associations thus sub-optimal matching. Since the research is based on the probabilistic map level, this chapter provides a comprehensive analysis of the map uncertainty modeling on high level geometric features and low level voxels. An example of the proposed framework is shown in Fig. 5.1. Top: Husky A200 robot platform equipped with Velodyne VLP-16 and the 3D edges ex-

Figure 5.1: An example of the proposed hierarchical probabilistic fusion framework.

tracted from the map. Middle: two individual maps of the WKW environment with the common area (black circle). Bottom: the fused global map and the zoomed in image of the overlapping area.

Generally, the *multi-robot 3D occupancy grid map fusion* consists of estimating the relative transformation between the local maps (i.e., map matching) and merging the occupancy map information (i.e., map merging), which are usually tackled independently of each other in this particular order [88, 62, 25]. Existing 3D occupancy map matching approaches utilize some form of dense registration algorithm to estimate the relative transform between maps, where the algorithm proposed in Chapter 3 also falls within this scope. This is typically realized by converting the occupancy grid map into a point cloud, by representing each lowest level voxel in the 3D map by its center coordinate, and then employing a point set registration algorithm [62]. The map merging step consists of verifying the accuracy of the estimated transformation and fusing the occupancy probabilities of the matching voxels [24]. When a map matching algorithm returns an accurate transformation, the majority of map merging approaches merely resort to either stitching the overlapped voxels or to

averaging the occupancy probabilities of matching voxels. While this two-step partitioning of 3D occupancy map fusion into a geometric problem and a probability fusion problem simplifies the solution and has seen significant development in recent years, it also introduces some issues that result in sub-optimal fusion results which need to be resolved. Besides, more detailed evaluation of these algorithms in the form of convergence and sensitivity to local map overlap changes warrants further examination and are the focus of this article.

Dense scan registration algorithms inspired initial solutions to map matching. These algorithms, such as ICP [72], NDT [81], GICP [77], OICP [25], are introduced to operate on raw scan data which are essentially samples of continuous surfaces with some noise. In the mapping process, sensor noise and robot localization errors are arbitrary digitized to generate 3D voxel maps with discontinuous and multiple voxel thick surfaces. Converting these voxel surfaces into the equivalent point clouds will not restore the continuous surfaces of samples as illustrated in Fig. 5.2. During the mapping process, motion uncertainty and sensor noise are accumulated and updated to a discontinuous occupancy grid map in the right image. Additionally, occupancy grid map formulation assumes voxel-wise independence to make the map update process computationally tangible. Ignoring the correlations among neighboring voxels with structural information also results in sub-optimal matching results.

To address the above challenges, the main objectives of the chapter are to develop a hierarchical probabilistic map fusion framework from a systematic viewpoint with the following contributions (see Fig. 5.3):

- The hierarchical probabilistic fusion system is proposed to address the integrated 3D occupancy map fusion problem, which is composed of three modules: uncertainty modeling, map matching and map merging.
- Based on the in-depth analysis and propagation of the 3D map uncertainty model, the two-level Probabilistic Map Matching (PMM) algorithm is developed by combining both high-level structural edge and low-level occupancy

Figure 5.2: The left image shows two consecutive scans of a continuous surface collected by the sensor.

voxel features.

- The statistical testing model is applied to evaluate the matching result, where the acceptance threshold is auto-tuned based on the overlapping between the maps. Then, the individual maps are merged by combining occupancy probability based on a relative entropy filter.

The rest of the chapter is organized as follows: Section II then introduces our first contribution of modeling the uncertainty of 3D occupancy map data at voxel level and structural level as related to map fusion. The two-level probabilistic map matching framework based on the uncertainty modeling is detailed in Section III. Section IV presents the proposed statistical testing for transformation estimation and the relative entropy-based map merging approach. Section V presents the results of the experiments conducted using both simulated and real data. Section VI concludes the chapter with a discussion on future work.

5.2 System Overview

The overview of the system architecture is depicted in Fig. 5.3. The focus of this chapter is to design the hierarchical system for map fusion, which can be divided into three modules, i.e., uncertainty modeling, map matching and map merging. The input of the system comes from the perception level, where a robot equipped

Figure 5.3: The diagram of the system.

with various sensors is utilized to map the environment. Prior to matching the maps, the individual voxel uncertainty and structural edge uncertainty are modeled and integrated to propagate the local uncertainty covariance of the 3D probabilistic map. Then, the map matching is performed with the hierarchical structure, where the matching starts from structural information layer and proceeds to full dense voxel map. The last step performs statistical testing to evaluate the quality of the transformation, where relative entropy filter probabilistic merging strategy is applied for successful estimations.

5.3 Map Uncertainty Modeling

This section exhibits the probabilistic mapping process and an efficient structural information extraction algorithm. Furthermore, the process of propagating and merging individual voxel uncertainty and structural edge uncertainty is presented.

5.3.1 Individual Voxel Uncertainty

Given an 3D occupancy voxel map G_r , and the information extracted from each voxel $(m_{rx}^i, m_{ry}^i, m_{rz}^i, L_{m_r}^i, d_{m_r})$ is defined as a tuple that includes the position of the extracted voxel center $(m_{rx}^i, m_{ry}^i, m_{rz}^i)$, the occupancy probability value $p_{m_r}^i$ and the voxel size d_{m_r} . Based on the independence assumption, the probability of a leaf node d_{m_r} given observation $z_{1:t}^i$ is updated by applying the Bayes filter (see Eq.(5.2)). The occupancy probability will update recursively when new sensor measurements are observed.

$$p(m_r^i | z_{1:t}^i) = \left(1 + \frac{1 - p(m_r^i | z_t^i)}{p(m_r^i | z_t^i)} \frac{1 - p(m_r^i | z_{1:t-1}^i)}{p(m_r^i | z_{1:t-1}^i)} \frac{p(m_r^i)}{1 - p(m_r^i)} \right)^{-1} \quad (5.1)$$

$$\propto \underbrace{p(m_r^i | z_t^i)}_{\text{sensor model}} \cdot \underbrace{p(m_r^i | z_{1:t-1}^i)}_{\text{previous estimation}} \quad (5.2)$$

The updating of Eq.(5.2) depends on the current observation $p(m_r^i | z_t^i)$, the previous estimation $p(m_r^i | z_{1:t-1}^i)$ and the prior probability $p(m_r^i)$. The term $p(m_r^i | z_t^i)$ is the sensor model that represents the sensor uncertainty. Fig. 5.4 illustrates an example of the sensor uncertainty and motion uncertainty, where the color bar represents the occupancy probability.. Fig. 5.4(a) and 5.4(b) show the map generated in the same environment with different sensors, which represents the uncertainty caused by the specification of the sensors. Since the raw sensor data registration only

Figure 5.4: An example of the sensor uncertainty and motion uncertainty in the 3D map.

considers the sensor uncertainty, what makes probabilistic map matching different from sensor data dense matching is the term $p(m_r^i | z_{1:t-1}^i)$. The previous estimation term $p(m_r^i | z_{1:t-1}^i)$ models the accumulation of previous observations and motion uncertainty. As shown in Fig. 5.4(c), large error with several thicknesses of walls is introduced when a sharp turn happens.

In probabilistic updating process, the only output is the occupancy probability value $p(m_r^i | z_{1:t}^i)$, where sensor readings and previous estimations are integrated into occupancy probability. To model the covariance of a voxel, the inverse of occupancy probability is utilized to represent the spherical covariance of a voxel. The Gaussian distribution $\mathcal{N}_{p_i}(\mu_{m_r^i}, \Sigma_{p_i})$ is located at the center of the voxel and identical in each

direction, as shown in Fig. 5.6.

$$\Sigma_{p_i} = \begin{bmatrix} p_{m_r^i}^{-1} & 0 & 0 \\ 0 & p_{m_r^i}^{-1} & 0 \\ 0 & 0 & p_{m_r^i}^{-1} \end{bmatrix} \quad (5.3)$$

5.3.2 Structural Edge Uncertainty

The edge voxels in a 3D occupancy grid map are the voxels in the 3D map that exhibit a sudden large change in local curvature (i.e. sudden bend in the surface) and are extracted following the steps described in Alg. 3. First the surface normals of the occupied voxels are computed (lines 2-6). This is done by performing Principal Component Analysis (PCA) on the local neighborhood N_i of each occupied voxel m_r^i , defined by the radius r_N (line. 4). The output of this operation are the eigen values $\Lambda = \{\lambda_1, \lambda_2, \lambda_3\}$ in decreasing order and the corresponding principal component vectors $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$. The surface normal is $\mathbf{v}_3$ which is the principal component corresponding to the smallest covariance (i.e. hence smallest eigen value λ_3) (line. 5) [122]. Voxels with surface normals that on average has an angle cost larger than a given threshold κ to the surface normals within their neighboring voxels are selected as candidate edge voxels (line. 8). The average angle cost of a surface normal n_i with respect to its neighboring surface normals is computed using Eq. 5.4, where the angle cost between two normal vectors is computed using their cross product.

$$\overline{Cost}_\theta(n_i) = \frac{\sum_{m_r^j \in N_i} |\mathbf{n}_j \times \mathbf{n}_i|}{|N_i|} \quad (5.4)$$

A validity check is performed (lines. 9-20) in order to detect valid edge voxels from the candidate edge voxels. During this check, vectors are traced along the surface

Algorithm 3 Structural Edge Voxel Detection

Require: M^O : The occupied voxels of the 3D grid Map

```

1:  $E = \phi$ 
2: for each  $m_r^i \in M^O$  do
3:    $N_i = \{m_r^j : m_r^j \in M^O \text{ and } \text{distance}(m_r^i, m_r^j) < r_N\}$ 
4:    $(\Lambda, V) = \text{PCA}(\mathcal{N}_i)$ 
5:    $n_i = V[3]$ 
6: end for
7: for each  $m_r^i \in M^O$  do
8:   if  $\frac{\sum_{m_r^j \in \mathcal{N}_i} |\mathbf{n}_j \times \mathbf{n}_i|}{|\mathcal{N}_i|} > \kappa$  then
9:     create  $W = (\mathbf{u}, \mathbf{v}, \mathbf{n}_i)$  right handed local coordinate frame at  $m_r^j$ 
10:    for  $\theta = \{0, \delta, \dots, \pi - \delta, \pi\}$  do
11:       $\mathbf{v}_1^W = (\cos(\theta), \sin(\theta), 0, 1)^T$ 
12:       $\mathbf{v}_2^W = -v_1^W$ 
13:       $\mathbf{v}_1^M = T_W^M v_1^W, \mathbf{v}_2^M = T_W^M v_2^W$ 
14:       $\mathbf{t}_1^M = \text{TraceVectorAlongSurfaceVoxels}(m_r^i, v_1^M, d_T)$ 
15:       $\mathbf{t}_2^M = \text{TraceVectorAlongSurfaceVoxels}(m_r^i, v_2^M, d_T)$ 
16:      if  $\alpha_{\min} < \arccos(\mathbf{t}_1^M \cdot \mathbf{t}_2^M) < \alpha_{\max}$  then
17:         $E = E \cup \{m_r^i\}$ 
18:        break
19:      end if
20:    end for
21:  end if
22: end for
23:  $E = \text{StatisticalOutlierRemoval}(E)$ 
24: return  $E$ 

```

Figure 5.5: The extraction of structural edge.

from each candidate voxel m_r^i in all directions of the local plane up to a distance d_T . Steps for initial trace vector generation for each m_r^i is given in lines. 11-13 where T_W^M is the transformation matrix from local frame W to global map frame M . t_1^M is the resultant vector after initial vector v_1^M is traced along the voxel surface from m_r^i up to d_T distance (line. 14). Value of 0.5m for d_T is used. If the resultant vectors t_1^M, t_2^M of any two opposing initial trace vectors v_1^M, v_2^M falls within the angle range $(\alpha_{min}, \alpha_{max})$, signifying a sudden bend in the local surface, m_r^i is detected as an edge voxel. Angle range is typically set to $(80^\circ, 100^\circ)$. The detected voxels in E are statistically filtered (line. 23) to remove any remaining outliers to generate the edge voxels of the 3D map. Output of the edge extraction process is depicted in Fig. 5.5. The left image shows the input 3D occupancy grid map generated in an indoor hallway, then the middle image presents the computed normals and detected 3D edge voxels from the 3D map. Finally, the outliers are removed and outputs the 3D edge voxels in the right image.

Figure 5.6: The process of combining individual voxel covariance (left) and structural edge covariance (right), where the middle image shows the merged result (middle).

Given the detected edge voxels, local uncertainty Σ_{τ_i} of each edge voxel m_r^i is derived using the neighboring edge voxels. Then, the spatial Gaussian model $\mathcal{N}_{\tau_i}(\mu_{m_r^i}, \Sigma_{\tau_i})$ is located at the center of the edge voxel $\mu_{m_r^i}$. By performing eigen decomposition, covariance Σ_{τ_i} is factorized into $\Sigma_{\tau_i} = \vec{v}_i \lambda_i \vec{v}_i^T$. The eigen values $\lambda_i = \text{diag}(\lambda_i^1, \lambda_i^2, \lambda_i^3)$ in ascending order correspond to eigen vectors $\vec{v}_i = [v_i^1, v_i^2, v_i^3]$, where the edge vector τ_i equals to $\vec{v}_i^3$. The edge covariance is shown in Fig. 5.6.

$$\Sigma_{\tau_i} = \begin{bmatrix} \vec{v}_i^1 & \vec{v}_i^2 & \vec{v}_i^3 \end{bmatrix} \cdot \begin{bmatrix} \lambda_i^1 & & \\ & \lambda_i^2 & \\ & & \lambda_i^3 \end{bmatrix} \cdot \begin{bmatrix} \vec{v}_i^1 & \vec{v}_i^2 & \vec{v}_i^3 \end{bmatrix}^T \quad (5.5)$$

5.3.3 Local Uncertainty Propagation

The significance of the structural information reconstruction comes at that, although the occupancy map is assumed to be voxel-wise independent, the continuous structural information is recovered by extracting the edges from the noisy map data.

Hence, with the individual voxel covariance $\mathcal{N}_{p_i}(\mu_{m_r^i}, \Sigma_{p_i})$ and structural edge model $\mathcal{N}_{\tau_i}(\mu_{m_r^i}, \Sigma_{\tau_i})$ available, the two uncertainty models are integrated into an unified

representation of the local uncertainty model $\mathcal{N}(\mu_{g_i}, \Sigma_{g_i})$:

$$\mathcal{N}(\mu_{g_i}, \Sigma_{g_i}) \sim \mathcal{N}(\mu_{m_r^i}, \Sigma_{p_i}) \cdot \mathcal{N}(\mu_{m_r^i}, \Sigma_{\tau_i}) \quad (5.6)$$

Where Eq.(5.6) is the weighted averaging algorithm with the updated mean and covariance:

$$\mu_{g_i} = (\Sigma_{p_i}^{-1} + \Sigma_{\tau_i}^{-1})^{-1}(\Sigma_{p_i}^{-1}\mu_{m_r^i} + \Sigma_{\tau_i}^{-1}\mu_{m_r^i}) = \mu_{m_r^i} \quad (5.7)$$

$$\Sigma_{g_i} = (\Sigma_{p_i}^{-1} + \Sigma_{\tau_i}^{-1})^{-1} \quad (5.8)$$

The process of integrating the uncertainty is shown in Fig. 5.6. The benefit of the uncertainty modeling and merging comes at two parts. Firstly, it accurately models the local uncertainty of the map with a properly designed Gaussian ellipse, which preserves the most valuable information in the environment. Secondly, the fusion combines the individual voxel covariance and structural edge covariance to represent the final uncertainty.

5.4 Two-Level Probabilistic Map Matching

To address 3D volumetric map registration, the two-level Probabilistic Map Matching (PMM) architecture is proposed. We start with formulating the two-level structure of the map matching algorithm, which starts from high-level edge features and moves to low-level full voxel matching. Then, map matching is further solved by decomposing it into establishing the probabilistic data association and optimizing the rigid transformation based on the data association.

5.4.1 The Formulation of Two-Level Probabilistic Map Matching Problem

5.4.1.1 Mathematical Definition

Problem *Given a set of 3D occupancy grid maps, each map can be denoted as $G_r = \{m_r, g_r\}$, where the point cloud $m_r = \{m_r^i\}_1^N$ and geometry features $g_r = \{g_r^i\}_1^L$ are extracted from the map G_r . The objective is to estimate the relative transformation T that maximizes the overlap between two maps.*

For dense registration of two maps G_r and G_s , the common solution is to fix one map G_r as the model map, and another map G_s as the scene map, where the rigid transformation T transforms the map G_s to the coordinate frame of G_r . The maximization of the overlapping between G_r and G_s is formulated as:

$$T = -\min_T \log p(G_r|T, G_s) \quad (5.9)$$

where $p(G_r|T, G_s)$ is the maximum likelihood estimation (MLE) that aims to find the most likely relative transformation T by matching the two partial maps G_r and G_s . The estimated matrix $T \in SE(3)$ is a rigid transformation matrix with rotation matrix $R \in SO(3)$ and translation vector $t \in \mathbb{R}^3$. The transformation T is iteratively minimized until a certain criteria is met.

5.4.1.2 Two-Level Map Matching Formulation

Here, the maps G_r and G_s are defined as 3D occupancy grid map and derive the structure for two-level matching. For the model map G_r , let full point cloud $m_r = \{m_r^i\}_{i=1}^{N_{m_r}}$ and geometry feature $g_r = \{g_r^i\}_{i=1}^{N_{g_r}}$. In this chapter, the geometry feature g_r^i refers to the coordinate of the voxel that belongs to the extracted edge. Instead of simply considering T as a fixed unknown parameter, it is modeled as a 6D

random variable and estimated with maximum likelihood estimation (MLE). The registration of G_r and G_s formulated in Eq.(5.9) is computed in two levels, where edge matching and full map voxel matching are tackled sequentially in Eq.(5.10).

$$T = \min_T \left(\underbrace{-\log p(g_r|T, g_s)}_{\text{Level 1: Edge Matching}} - \underbrace{\log p(m_r|T, m_s)}_{\text{Level 2: Full Map Matching}} \right) \quad (5.10)$$

The two-level structure works sequentially. Firstly, in the first level, the extracted edges are matched by utilizing the weighted linear combination of positional distance, occupancy probability distance and edge direction distance into an overall distance metric. Then, the probabilistic quadratic function is solved with the non-linear optimization algorithm. In the second level, full voxel maps are matched to achieve accurate transformation and derive the corresponding pairs for probability merging. Here, the chapter assumes voxel-wise correspondences to be independent, the probability density function in Eq.(5.10) can be simplified into Eq.(5.11), where $m_s^{c_i}$ and $g_s^{c_j}$ are the candidate corresponding voxels of m_r^i and g_r^j , respectively.

$$T = \min_T \left(\underbrace{-\sum_j \log p(g_r^j|T, g_s^{c_j})}_{\text{Level 1: Edge Matching}} - \underbrace{\sum_i \log p(m_r^i|T, m_s^{c_i})}_{\text{Level 2: Full Map Matching}} \right) \quad (5.11)$$

The problem of estimating the transformation matrix is often solved in two steps: data association and nonlinear optimization. Since the data association is unknown, the problem in Eq.(5.11) is iteratively estimated. First, the data association step efficiently estimates the hidden corresponding pairs by evaluating the similarity of the local descriptor. Then, the optimization step solves the rigid transformation based on the previous step by marginalizing the likelihood function.

5.4.2 Probabilistic Data Association

To estimate transformation T , data association between the model voxel m_r^i (g_r^j) and scene voxel $m_s^{c_i}$ ($g_s^{c_j}$) needs to be established, which is usually estimated by performing unidirectional nearest neighborhood search of Euclidean distance. With the presence of map noise, its hard to find the true correspondence with purely coordinate information [72]. In [87, 78], geometrical features such as curvature, surface normal and point cloud density are introduced to find the accurate correspondence relationship.

In our previous work [25], occupancy probability is incorporated to establish voxel-to-voxel correspondences between all overlapped partial maps in $4D$ space. With the structural information extracted from Sec.5.3, it will provide additional information to restore the corresponding relationship between two occupancy grid maps. Here, edge matching is augmented to $7D$ local descriptor in the first level with $\{m_i^x, m_i^y, m_i^z, p_{m_i}, \tau_i^x, \tau_i^y, \tau_i^z\}$, which includes the center coordinate of voxel ($3D$), occupancy probability ($1D$) and edge direction ($3D$).

The first level of the registration applies full $7D$ local descriptor with the distance metric defined in Eq. (5.12), the distance metric $d(g_r^i, g_s^j)$ is described by Euclidean point distance $d_e(g_r^i, g_s^j)$, occupancy probability distance $d_o(g_r^i, g_s^j)$ and edge geometry distance $d_\tau(g_r^i, g_s^j)$. The weights for the corresponding pair search are inversely proportional to the variance of the particular channel of the source information. The weighting vector $\omega = \{\omega_o, \omega_\tau\}$ is introduced to determine the importance of the occupancy probability and geometry distance relative to the positional distance.

$$\begin{aligned} d(g_r^i, g_s^j) &= d_e(m_r^i, m_s^j) + \omega_o d_o(p_{m_r^i}, p_{m_s^j}) + \omega_\tau d_\tau(\tau_r^i, \tau_s^j) \\ &= \|m_r^i - T \oplus m_s^j\|^2 + \omega_o \|p_{m_r^i} - p_{m_s^j}\|^2 + \omega_\tau \|\tau_r^i - T \oplus \tau_s^j\|^2 \end{aligned} \quad (5.12)$$

When it moves to the full map matching level, the structure information is not

available. In this case, the dimension of descriptor reduced to $\{m_i^x, m_i^y, m_i^z, p_{m_i}\}$ in 4D space, which is the same as [25]. The error metric defined in Eq.(5.13) denotes the error metric of full map matching, where $\omega_\tau = 0$.

$$\begin{aligned} d(m_r^i, m_s^j) &= d_e(m_r^i, m_s^j) + \omega_o d_o(p_{m_r^i}, p_{m_s^j}) \\ &= \|m_r^i - T \oplus m_s^j\|^2 + \omega_o \|p_{m_r^i} - p_{m_s^j}\|^2 \end{aligned} \quad (5.13)$$

The weighting vector $\omega = \{\omega_o, \omega_\tau\}$ measures the importance of the occupancy probability and geometry distance relative to the positional distance. The weight parameters $\{\omega_o, \omega_\tau\}$ are scale factors to determine the contribution of occupancy probability and geometry distance. Since weight ω_o has been decided in Chapter. 3, the focus here is to estimate ω_τ . To model the two weights ω_τ , the uncertainties of structural information are estimated as described below.

5.4.2.1 The Uncertainty of Structural Information

For structural spatial covariance Σ_{τ_i} , the corresponding eigenvalues of $\lambda^i = \text{diag}[\lambda_1^i, \lambda_2^i, \lambda_3^i]$ is in ascending order. The eigenvalues denote the length of the principal coordinate axes, which are the structural uncertainties along three directions. More specifically, the largest eigenvalue λ_3^i corresponds to the edge vector τ_i . Here, Eq.(5.14) is defined to represent the uncertainty of the edge.

$$\lambda_\tau = \sum_i^n \frac{\lambda_3^i}{n(\lambda_1^i + \lambda_2^i + \lambda_3^i)} \quad (5.14)$$

When the edge is perfect without any noise on the perpendicular plane, $\lambda_\tau = 1$. And the extreme case is the three eigenvalues are equal, and the three axes have the same length, then $\lambda_\tau = \frac{1}{3}$. In the second level of full voxel map registration, $\lambda_\tau = 0$ as no structural information is available.

Figure 5.7: An example of data association established between two L-shape datasets.

5.4.2.2 Overall Weights

With the uncertainties of occupancy probability, structural information and positional error estimated in Eq.(3.11), (5.14), (3.15), the weights ω_o, ω_τ defined in Eq.(5.12), (5.13) are denoted as:

$$\omega_o = k_1 \cdot \frac{\sigma_{TSx}}{H_m} \approx k_1 \cdot \frac{MSE}{H_{m_r}} \quad (5.15)$$

$$\omega_\tau = k_2 \cdot \sigma_{TSx} \cdot \lambda_\tau \approx k_2 \cdot MSE \cdot \lambda_\tau \quad (5.16)$$

where k_1 and k_2 are the normalization factors. As the registration iteratively minimizes the error function, MSE is non-increasing. Since weights ω_o and ω_τ are constant in the matching process, the MSE should tend to zero as the matching converges to a perfect match. When the positional error is large, we should rely more on occupancy probability value and geometric information (only for the first level). As the two maps get closer, the algorithm tends to rely more on positional value.

To show the superiority of our 7D distance metric compared to purely Euclidean distance, an example of data association between two L-shape edges is presented in Fig. 5.7. In Fig. 5.7(a), it only considers the Euclidean distance. While in Fig. 5.7(b), the 7D augmented descriptor is applied. The full line red box shows the wrong data

association estimated in Fig. 5.7(a), which is removed in Fig. 5.7(b). While the black box in the upper image shows the missed data association, which is found and connected with 7D descriptor in Fig. 5.7(b). As can be seen, the data association using only Euclidean distance based closet neighborhood search falls into the false match, which might lead to wrong convergence in the optimization process. In contrast, the 7D distance metric combined with the augmented information filters out the wrong correspondences and finds correct pairs. The results prove that combining edge information, occupancy probability, and point coordinate tend to find accurate data association.

5.4.3 Error Metric Optimization

Based on the two-level structure, the registration strategy starts from the geometry structural edge layer and proceeds to the dense grid voxel layer. The optimization problem is to minimize the error function defined in Eq.(5.11) with the data associations estimated.

Starting with the edge optimization, robust correspondences have been established, and fewer voxels will be optimized. Applying the prolonged shape of the local uncertainty covariance Σ_{g_i} located along the edge direction will penalize two edges not lying in the same direction. The applying of local uncertainty covariance reduces the number of iterations compared to Euclidean distance based optimization algorithm. Then it forces the two edges to have the same direction with positional coordinate aligned, which enhances the convergence basin to inaccurate initial estimation and improves overall registration results.

$$T = \min \sum_j (g_r^j - T \oplus g_s^{c_j})^T (\Sigma_{g_r^j} + R^T \Sigma_{g_s^j} R)^{-1} (g_r^j - T \oplus g_s^{c_j}) \quad (5.17)$$

Figure 5.8: Given the same datasets with two different rigid transformations.

Based on the edge registration outputs, the dense voxel-wise correspondences will be established to refine the estimation. Here, the uncertainty model involved is the individual voxel covariance Σ_{p_i} . Since the covariance is equal in three directions, rotation matrix has no impact on Σ_{p_i} . Then the error function is simplified to the norm of Euclidean distance with the scale factor (see Eq.(5.18)). Since not all the corresponding error terms should have the same weight, larger weight is assigned to the associations with higher occupancy probability.

$$T = \min \sum_i (p_{m_r^i}^{-1} + p_{m_s^{c_i}}^{-1})^{-1} \|m_r^i - T \oplus m_s^{c_i}\|^2 \quad (5.18)$$

5.5 Transformation Evaluation and Probability Merging

In this section, the statistical testing of accepting or rejecting the transformation is presented. Then, a relative entropy filter is proposed to measure the map dissimilarities and merge individual maps into a global map.

5.5.1 Transformation Evaluation

In Chapter. 3 an environment measurement model (EMM) is applied to verify the transformation by calculating the Mahalanobis distance between matched points. Here, point cloud map m_s has been transformed to the coordinate frame of m_r and is denoted as ${}^r m_s$. Since the edge matching is a middle process, the evaluation is focused on the voxel-wise correspondences between full map voxels. The statistical evaluation of map m_r given ${}^r m_s$ is formulated in Eq.(5.19).

$$p(m_r | {}^r m_s) = \prod_i^{N_{m_r}} p(m_r^i | {}^r m_s^{c_i}) \quad (5.19)$$

A proper inconsistency distance should be defined to describe the discrepancy between submaps. As illustrated in Eq.(5.18), the probability of each correlated pair can be computed as follows:

$$p(m_r^i | {}^r m_s^{c_i}) = \exp \left(- \frac{(m_r^i - {}^r m_s^{c_i})^T \cdot (m_r^i - {}^r m_s^{c_i})}{p_{m_r^i}^{-1} + p_{m_s^{c_i}}^{-1}} \right) \quad (5.20)$$

The inliers are decided by setting a threshold of Mahalanobis distance. Then, a confidence level is applied as the criteria for rejecting the wrong estimates by computing the overall fraction of inliers. The critical thing is to set a proper threshold for the ratio to accept the transformation. To address this issue, the parameter Θ

is introduced to measure the overlapping between the two registered maps. The value is calculated by dividing the number of matched pairs N_{match} by the number of voxels on two maps.

$$\Theta = \frac{1}{2} \cdot \left(\frac{N_{match}}{N_{m_s}} + \frac{N_{match}}{N_{m_r}} \right) \quad (5.21)$$

Since accepting a wrong rigid transformation will lead to a diverged global map, a conservative acceptance rate is set to be 60%. Besides, as the overlapping decreases, the acceptance rate is enhanced with special care for extreme low overlap cases. The relationship between the overlapping metric Θ and the acceptance threshold of the transformation is defined as follows:

- $\Theta < 20 \%$, the inlier percentage should $> 80 \%$
- $20 \% < \Theta < 60 \%$, the inlier percentage should $> 0.9 - \frac{\Theta}{2} \%$
- $\Theta > 60 \%$, the inlier percentage should $> 60 \%$

This simple relationship leads to satisfactory performance in our experiments. An example in Fig. 5.8 shows the applying of statistical testing efficiently rejects high erroneous estimates that would largely undermine the quality of the fused map. Statistical testing is performed to decide whether to accept the transformation. The inliers are denoted with black, while the outliers are red. In Fig. 5.8(c), the acceptance rate 70% is computed using the overlapping parameter. In the example, the overlapping area is denoted with the red box, and overlapping metric Θ is 40%. Then the inlier threshold should be $0.9 - \frac{\Theta}{2} = 0.7$. In both scenarios, there are six corresponding pairs established. In Fig. 9a, there are four outliers and two inliers, while the six matching pairs are all inliers in Fig. 9b. Since the inlier acceptance rate is 70%, Fig. 9c shows the final results of the transformation evaluation.

Table 5.1: Summary of the threshold parameters

Notation	Location	Value
κ	Alg. (3)	set as 0.5
$d(g_r^i, g_s^j)$	Eq.(5.12)	set as 2
$d(m_r^i, m_s^j)$	Eq.(5.13)	set as 0.5
inlier percentage	Eq.(5.20)	depending on overlapping metric Θ
$KL(o_{m_r^i} o_{m_s^{c_i}})$	Eq.(5.22)	set as 0.1

5.5.2 Relative Entropy Filter

We introduce $o_{m_r^i}$ as a discrete random variable (RV) with two states : occupied and free, where $p(o_{m_r^i} = \text{occupied}) = p_{m_r^i}$, $p(o_{m_r^i} = \text{free}) = 1 - p_{m_r^i}$, and the difference between the two RVs are computed as:

$$KL(o_{m_r^i} || o_{m_s^{c_i}}) = p_{m_r^i} \log \frac{p_{m_r^i}}{p_{m_s^{c_i}}} + (1 - p_{m_r^i}) \log \frac{(1 - p_{m_r^i})}{(1 - p_{m_s^{c_i}})} \quad (5.22)$$

As mentioned in Chapter. 3, Kullback-Leibler (KL) divergence is used relative entropy filter to measure the difference between two random variables. $KL(o_{m_r^i} || o_{m_s^{c_i}})$ represents the difference between the two probabilities. If KL is within a certain threshold, the dissimilarity is seemed as random errors and the corresponding pairs are fused with weights

$$p_{fuse}^i = \frac{p_{m_r^i}}{p_{m_r^i} + p_{m_s^{c_i}}} p_{m_r^i} + \frac{p_{m_s^{c_i}}}{p_{m_r^i} + p_{m_s^{c_i}}} p_{m_s^{c_i}} \quad (5.23)$$

If KL is beyond the threshold with $p_{m_r^i} > p_{m_s^{c_i}}$, it means the voxel with lower probability $p_{m_s^{c_i}}$ does not have enough information for the environment and will be rejected, only voxel with higher probability is considered and let $p_{fuse}^i = p_{m_r^i}$.

5.6 Experimental Results

Experiments conducted in simulated and real-world environments are summarized in this section. Both qualitative and quantitative results are presented in the following aspects:

- a) Evaluation of how the proposed approach improves the accuracy in both translation and rotation estimation compared with the baseline algorithms, where fewer iterations required to achieve the same accuracy.
- b) Exploration of the effect of reducing the overlap between the two maps showing that extracting the structural information increases convergence.
- c) Showing that the efficiency of the proposed algorithm is significantly improved compared to the baseline algorithms, proving that real-time performance is guaranteed.
- d) Presenting that our algorithm increases the robustness under initial perturbations, where statistic results show the improved performance on accuracy and convergence.
- e) Evaluation of how the statistical testing accepts the correct transformation and merges the map using relative entropy filter.

As mentioned before, the research is focused on the fusion stage with an initial relative transform estimation. The initial relative transformation between the coordinate frames of different robots can be extracted during the deployment phase of the mission if such information is available. Else it can be extracted later on through coordinated rendezvous actions [17].

To simulate the noisy initial relative transformations, two maps with different initial transformations both in translation and rotation were generated.

Table 5.2: Summary of the experiment environment.

Dataset	Dimension(x*y*z)	Map Size	Edge Size	Environment Type
Corridor Dataset	20m*5m*3m	46000 voxels	3500 voxels	Structural, static
WKW Dataset	40m*15m*3m	67071 voxels	4300 voxels	Semi-stru, dynamic
Willow Dataset	50m*40m*3m	76000 voxels	3200 voxels	Structural, static

5.6.1 Evaluation Protocol

Experiment Environment: Experiments conducted using simulated and real data are compared by operating the robot to collect data and generate maps in three different scenarios in Nanyang Technological University (see Tab.5.2). Each robot was equipped with a Hokuyo Laser range finder for pose estimation using Gmapping, Asus Xtion Pro RGB-D camera for 3D perception. In the simulation dataset Willow, the robot was tele-operated to explore environment twice from the same starting point. The Willow is a standard ROS/ Gazebo office world with varying room sizes and corridors. In the real environments Corridor and WKW, the recorded datasets were divided into two segments to generate two different maps with some overlap. The corridor is a long corridor in the building, while the WKW is the lobby of a building with walking people and clutter furniture/objects. The ground truth transformation between the maps was acquired by recording the experimental setup. The resolution of the 3D occupancy grid map was set to be 0.1m.

Error Metric: The chapter compares the estimated transformation T_e to the ground truth T_g . Then the error ΔT is calculated as $\Delta T = T_e \cdot T_g^{-1} = \{\Delta R, \Delta t\}$. The 3D translation error $e_t = \|\Delta t\|$ and 3D rotation error $e_r = \|\Delta R\| = \arctan \Delta R$ are the Euclidean norm of and Euler norm of the difference between the ground-truth and the output. Another metric MSE (Mean Squared Error) is the mean squared Euclidean distance between the matched pairs.

Comparison Baseline: To demonstrate the superiority of the two-level PMM algorithm, the comparison is conducted on two levels. Firstly, the ICP based edge matching is compared against the first level of PMM. Then, the full PMM is com-

pared against ICP [72], point-to-plane (P2P) ICP [123], Generalized ICP (GICP) [77] and Normal Distribution Transform (NDT) [124] based dense registration algorithm. The implementation of the compared algorithms are from the standard Point Cloud Library [125].

5.6.2 Edge Matching Analysis

One of the novelties of the proposed first level PMM comes at the extraction of structural information and modeling into a probabilistic registration problem. An alternative way to register extracted edges is to directly apply point-to-point ICP algorithm, which ignores the local uncertainty of each edge voxel. Here, the first level of PMM is compared against the ICP edge matching on two datasets. The initial perturbations were randomly generated for the two experiments, and the registration results of convergence rate and matching error are exhibited.

5.6.2.1 Registration Accuracy

The experiment was conducted in two datasets by extracting the edges from the full 3D map. Fig. 5.9(a) shows the ICP edge matching trapped into local minimum after several iterations. The reason is due to the wrong data association established, as shown in the black line in the figure. Therefore, the algorithm fails to converge to the right global minimum. Compared with ICP edge matching, the first level of PMM makes full use of the constraint on x and y directions and combines the local uncertainty information. Hence, the correct correspondences are established and gradually converged to the desired position.

In the WKW dataset, both algorithms converge to the correct minimum (see Fig. 5.9(b)). However, the algorithm proposed requires fewer iterations compared with ICP edge matching. In the experiment, our algorithm converges to the local minimum after around 15 iterations, while the classical ICP requires more than 25 iterations. In

(a) Registration results of the edges extracted from Corridor dataset. The black lines show the correspondences established between the two edge maps.

(b) Registration results of the edges extracted from WKW dataset. Both algorithms converge to the correct position, where the first level of PMM converges much faster.

Figure 5.9: Comparison between edge registration algorithms, where first level of PMM is compared against ICP based edge registration algorithm.

summary, our algorithm increases the efficiency of convergence speed by nearly 40%. The reason is that the proposed optimization algorithm penalizes more on the offset between the edge direction and increases the convergence speed.

5.6.2.2 Sensitivity to Overlapping

Since the maps are viewpoints independent and the assumption of large overlapping in dense registration does not hold true here. The influence on different overlapping parameters is also tested. The original overlapping rate between the two maps in Fig. 5.9(b) is 45%. Then, fixing the scene map to remove part of the map data by decreasing 1m gradually on x-direction of the scene map for nine iterations ($x = -9m$). The removing of the map data leads to the decrease of the overlapping rate to 10% when $x = -9m$. In Fig. 5.10, the change of rotation and translation error with the dropping of the overlapping rate is summarized, where our algorithm shows a larger convergence basin. In contrast, the classical ICP based approach has

Figure 5.10: Sensitivity to the change of overlap in WKW dataset, where ICP edge matching and first level PMM are compared.

a large sliding error in the x-direction. The result shows proposed algorithm can make the most use of the map information and increase the region of attraction of the correct minimization direction.

5.6.3 Full Map Matching Analysis

For the three experiments above, an exhaustive comparison between ICP map matching, P2P ICP map matching, GICP map matching, NDT map matching, first level PMM and full PMM has been carried out to test the results of registration accuracy and sensitivity to different initial inputs. The results of registration on three different scenarios are compared against the ground truth relative transformation. The scene map was transformed away from the ground truth and used as input to test the convergence. For each testing pairs, the offsets between $-2m$ and $2m$ in steps of 1m (along x and y-axes) were introduced. In addition, the maps were rotated from -15° to 15° (in steps of 7.5°). Thus, a total of 125 test transformations were generated and applied to the maps. For all algorithms, the threshold for nearest neighbor search is $2m$, and the number of iterations is set to be 30.

(a) Corridor dataset translation error

(b) WKW dataset translation error

(c) Willow dataset translation error

(d) Corridor dataset rotation error

(e) WKW dataset rotation error

(f) Willow dataset rotation error

Figure 5.11: Summary of convergence of the evaluated methods under 125 initial perturbations. Translation error (top) and rotation error (bottom).

Table 5.3: The mean error in both rotation and translation estimations. (The best performance is denoted in bold.)

Dataset	ICP		P2P ICP		GICP		NDT		First PMM		Full PMM	
	Euler	Meter	Euler	Meter	Euler	Meter	Euler	Meter	Euler	Meter	Euler	Meter
Corridor	31.39	1.63	19.97	1.51	10.58	1.71	13.63	2.56	12.06	0.62	9.37	0.57
WKW	5.77	1.95	6.31	1.78	5.32	2.84	5.51	2.32	6.90	1.47	5.60	1.34
Willow	6.54	3.60	4.97	2.78	5.28	2.72	8.07	5.13	8.88	2.54	4.99	2.00

5.6.3.1 Sensitive to Initial Perturbations

The boxplot presented in Fig. 5.11 supports that the full PMM algorithm outperforms to those of ICP, P2P ICP, GICP, and NDT. The boxplot encloses the middle 50% of the data, with the median value marked by a line and the mean value marked by a green rhombus. The red cross marks indicate outliers, i.e., data values below one standard deviation above and below the mean of the data.

For the first scenario, the translation and rotation errors of the algorithms except full PMM present a significant variance, indicating that the optimization process often falls into the local minimum. The reason is that the corridor environment has low translational features in the y-direction, where other algorithms fail to utilize the y-direction constraint in the optimization process. In the second environment, all algorithms present acceptable performance. Thanks to the abundant translational features on x-direction and y-direction, all the algorithms can relatively converge to the correct transformation. In the third environment, large translation errors are found for all the algorithms. This is due to locally similar geometries, where the small office rooms are very similar. The rotation errors are much lower, suggesting that in the structured environment the translation errors are the primary concern.

5.6.3.2 Convergence Analysis

Tab.5.3 summarizes the mean error in both rotation/translation estimations, the full PMM outperforms other algorithms in most of the cases. The exception is the rotation estimation of P2P ICP slightly exceeds full PMM in Willow dataset, and

(a) The success rate in Corridor dataset

(b) The success rate in WKW dataset

(c) The success rate in Willow dataset

Figure 5.12: Summary of success rates of the evaluated matching methods over all datasets. Translation estimation (top) and rotation estimation (bottom).

NDT slightly exceeds full PMM in WKW dataset. Besides the performance of full PMM, the GICP achieves the second best results. Since not all the outputs are successful, the criteria for a correct registration was chosen as a translation error of $0.6m$ or less and a rotation error of 6° or less.

The performance summarized in Fig. 5.12 exhibits the successful fraction of all three datasets. The trend is clear that the smallest errors in Tab.5.3 exhibit the highest percentage of success. Fig. 5.12(a) shows poor performance on ICP and P2P ICP, and the most likely cause is the wrong data association due to the low constraint on y-direction. Fig. 5.12(b) shows that P2P ICP, GICP, and full PMM achieve comparable convergence basin. In the Willow dataset (see Fig. 5.12(c)), all the algorithms witness a large offset on translation estimation, which is caused by the highly symmetric structure of the small office rooms. Overall, Full PMM demonstrates superior or comparable performance over existing algorithms.

5.6.3.3 Efficiency Analysis

The overall computation time of full PMM is compared against the ICP, P2P ICP, GICP and NDT methods and results are shown in Tab. 5.4. The results indicate that ICP, P2P ICP, and full PMM require a significantly lower computation time compared to NDT and GICP. The high computation time of NDT is due to computing the required normal distributions from the octomap input in addition to the fusion step. GICP adopting a more similar approach to PMM, generates the second best map registration accuracy of all methods, however it's higher computation time can be attributed to the use of only the voxel level registration requiring more computation efforts compared to our proposed two level PMM. Therefore, it can be claimed that the high efficiency of the full PMM compared to all other methods stems from the use of the two-level structure, where edges are registered in the first level with fast convergence with high accuracy which causes the voxel based registration at the second level to converge quickly as well, making the entire process more efficient.

Table 5.4: The efficiency comparison of the algorithms (Seconds)

Datasets	ICP	P2P ICP	GICP	NDT	Full PMM
Corridor	3.14	2.90	24.08	136.42	2.13
WKW	10.35	8.13	46.66	233.33	5.31
Willow	5.59	4.47	40.27	217.38	3.83

5.6.4 Statistical Testing and Map Merging

In the standard library, GICP and NDT cannot provide point-wise correspondence. We only discuss ICP, P2P ICP and full PMM in the following experiments. After computing the transformation, the accuracy is evaluated by calculating the percentage of inliers and merged based on relative entropy filter.

5.6.4.1 Statistical Testing and Transformation Evaluation

Fig. 5.13 illustrates the comparison of the average Mahalanobis distances among P2P ICP map matching, ICP map matching and full PMM in the three experiment settings. The figure shows the box plots that enclose the middle 50% of the data with the median value marked by a line. It is apparent that our algorithm has lower variance and mean compared with others.

Since ICP is deterministic with only coordinate position, the covariance of each point is denoted as the identity matrix $I_{3 \times 3}$. For the three experiments, a strict criterion is set to guarantee the accuracy: the inliers threshold is set as 0.2 Mahalanobis distance, and the fraction of rejecting a transformation is set according to the overlapping rate. The overlapping rate for Willow dataset, WKW dataset, and Corridor dataset are 40%, 42%, and 48%, then, the acceptance is calculated as 70%, 69%, and 66% respectively.

In Fig. 5.14, the acceptance threshold is denoted with a black line, where the inlier rate higher than the threshold is accepted. In Corridor and Willow datasets, the inlier percentages of ICP and P2P ICP are lower than full PMM since they fail

Figure 5.13: The distribution of the average Mahalanobis distance between the matched pairs in all three experiments.

to converge to perfect alignment. Based on the auto-tuned thresholds, full PMM transformation generates accurate and reliable map registration result compared with ICP and P2P ICP based map matching approach. Alternatively, both full PMM and P2P map matching algorithms produce acceptance results in the WKW environment.

5.6.4.2 Relative Entropy Filter based Map Merging

After determining the correct transformation for map registration, a relative entropy filter is applied to achieve map merging by considering the dissimilarities between matched voxels. The relative entropy threshold is set as 0.1 in KL-divergence.

Figure 5.14: The fraction of inliers and the threshold of accepting the estimated transformation.

The entropy of the resultant maps after applying the KL-divergence based merging algorithm is summarized in Tab. 5.5. In comparison, the merging strategy in [62] is performed by directly taking the average of probabilities for all the matched voxels. Besides, the algorithm is also compared with directly stitching the two maps without any post-processing.

The results are summarized in Tab.(5.5). It is shown that the relative entropy filter based map merging algorithm reduces the size of the fused map and mean of the entropy. Both relative entropy filter and taking an average algorithm to lead to compression for the size of the map compared with directly stitching. The entropy of WKW and Corridor environment is higher than Willow due to real-world noise and uncertainty. The experiment indicates that the proposed map fusion strategy

Table 5.5: Average entropy of fused map for different update rule. (The best performance is denoted in bold.)

Dataset	Directly Stitch		Taking Average		Relative Entropy	
	Size	Entropy	Size	Entropy	Size	Entropy
Corridor	40763	0.2061	32787	0.2141	32787	0.2002
WKW	101364	0.2374	82765	0.2371	82765	0.2317
Willow	95596	0.1583	75442	0.1680	75442	0.1526

Figure 5.15: The results of fused maps for all the three experiments.

can combine probabilities of both maps effectively to decrease the total uncertainty. Fig. 5.15 shows the final results of the fused 3D occupancy grid maps of three environments.

5.7 Conclusions

This chapter extends the works of Chapter 3 by proposing a novel hierarchical probabilistic map fusion framework for 3D occupancy grid maps, where the uncertainties

of the structural features and individual voxels are used during matching in a two-step approach. This increases the convergence rate of the optimization process as the use of local uncertainty prevents outliers from dominating the solution.

The work addresses the hierarchical probabilistic fusion framework for 3D occupancy maps, which is factorized into three modules: uncertainty modeling, map matching, and map merging. The chapter first clarifies the difference between raw sensor data registration and map matching in-depth. Then, the local uncertainty of maps is modeled by combining the individual voxel and structural edge covariances. To calculate the accurate relative transformation between maps, a two-level probabilistic map matching algorithm is proposed to combine structural information and voxel information. For the first level, 3D edges extracted from the maps are aligned based on the probabilistic data association strategy. Then, the full maps are correlated on voxel-wise to refine the result and derived the corresponding pairs for probability merging. Since the transformation is only a local minimum, statistical testing is performed to evaluate the quality of the transformation by rejecting low inlier fraction. The final map merging is achieved with a relative entropy filter which integrated the measurements from both maps and decreased the uncertainties. Experiments on simulated and real-world environments are conducted, and the evaluations confirms the efficacy of the proposed approach in generating more accurate relative transformation estimates between maps with a faster convergence rate and a larger convergence basin. The proposed algorithm is able to produce more consistent global 3D occupancy grid maps compared with existing methods.

In this chapter, the map fusion problem in structured environment with a specific sensor type is discussed. For a group of robots in arbitrary (structured/semi-structured/unstructured) environments, single sensor is far from enough in mapping accuracy and robustness. Maps generated with heterogeneous sensors need to be matched and merged to combine information with a higher information content. Chapter 6 will present a general probabilistic framework for collaborative 3D mapping using heterogeneous sensors

Chapter 6

A General Probabilistic Framework for Collaborative 3D Mapping using Heterogeneous Sensors

6.1 Introduction

In single robot level, with the emergence of diverse sensor types, each robot could carry heterogeneous sensors (laser scanners and vision-based sensor). The nature of the map produced by different sensors is the major challenge, i.e, density, range, sensor noise and field of view. Considering the 3D Lidar (Velodyne) scans 360° and records most of the environment without leaving any major blind spots, however, generates a less dense map owing to sensors sparseness. While the RGB-D camera (Asus Xtion) generates denser and more detailed 3D map with the surroundings, however, RGB-D camera only has limited field of view and the error grows exponentially with the increasing of the sensing range. Hence, the fusion of maps produced

Figure 6.1: An example of the heterogeneous map fusion.

by miscellaneous sensors could present more detailed information of the environment and increase the perception ability of the robot (See. Fig.6.1. Left: Robot platform Husky A200 equipped with various sensors. Middle: The map generated by Velodyne and RGB-D camera. Right: fused global map.).

In multi-robot level, considering the complexity and dimension of the environment, each robot only has partial field of view and limited sensing ability of the surroundings. In order to allow the team of robots to model the environment and plan tasks from a systematic viewpoint, sharing and fusing map data perceived by each robot among all robots act as the basis for multi-robot coordination. The research of exploiting multi-robot map fusion framework in both single robot and multi-robot levels has not been studied extensively [126]. In general, map fusion on both levels is composed of estimating the relative transformation and integrating the partial map information, which are usually tackled separately [24]. Existing approaches have relied on low level point set registration based algorithms for map matching [62], where hard one-to-one correspondences relationship are assigned. When maps produced by heterogeneous maps need to be matched, it is hard to detect correct corresponding points. Most researchers focus on estimating relative transformation and directly applying the transformation to stitch the partial maps [20], where the integrating of transformation uncertainty and map uncertainty are not considered.

In addition, they have not provided a comprehensive analysis of the integrated map fusion system on map uncertainty modeling, probabilistic map matching, transformation evaluation and accurate map merging.

In this chapter, a novel multi-level framework is presented allows for joint local and global map matching and merging, which acts as the theory basis for multi-robot mapping missions. More specifically, the multiple data association method to match heterogeneous maps; Then a time-sequential map merging algorithm is applied. We designed an efficient distributed solution and perform all computation online and on board to gain robustness w.r.t. periodic communication and robot failure. The main contributions of this chapter are listed as follows:

- A general probabilistic framework is proposed to address the integrated map fusion problem, which is independent of sensor types and SLAM algorithms. The framework provides theoretical basis for computing the relative transformations (map matching) and merging probabilistic map information (map merging).
- A probabilistic map matching algorithm is proposed that automatically decides the numbers of voxels to be associated for maps produced by heterogeneous sensors with different physical properties.
- A time-sequential map merging framework that makes fusing maps from distributed multi-robot system efficiently, where uncertainty embedded in the map and caused by the relative transformation are propagated and integrated.

6.2 Distributed Multi-Robot Map Fusion

In this section, the system architecture for multi-robot map fusion with single robot level and multi-robot level is presented, where the problem is formulated to provide

a theoretical basis for its factorization into two sub problems of map matching and map merging.

6.2.1 System Architecture

The overview of the system architecture is depicted in Fig.6.2. The focus of this chapter is to build a general framework for map fusion, which can be divided into two sub-problems, i.e., map matching and map merging. **First level** is perception level, the robot equipped with heterogeneous sensors will generate its map. For example, laser scanner tends to generate 360 degree sparse map, while rgb-d camera generates denser map with narrow field of view. **Second level** is local map level, the maps produced by different sensors are fused locally on each robot to combine the strength for each sensors. **Third level** is global map level, the robots communicate with each other to transmit and fuse the local maps into a global map. Our main focus is on the second level and third level.

6.2.2 System Framework Definition and Formulation

Problem 1 *Given a group of robots $\mathcal{R} = \{r\}_{1:n_r}$, each robot r is equipped with a set of heterogeneous sensors $\mathcal{S} = \{s\}_{1:n_s}$, the map produced by robot r using sensor s is denote as $m_{r,s}$, the problem is to estimate the fused global map M_r and the set of relative transformation $\mathcal{T}_r$ with a fully distributed network.*

Let $\mathcal{R} = \{r_1, r_2, \dots, r, \dots\}$ (the number of robots is n_r) be the set of robots in an environment. The neighboring robots of robot r is defined as $\mathcal{R}_r = \{r_n, r_n \in \mathcal{R}_r\}$, where neighboring robots $\mathcal{R}_r$ is the set of robots in communication with r and $\mathcal{R} = \{r \cup \mathcal{R}_r\}$ if the communication covers all robots. A diagram of group of robots is shown in Fig. 6.3.

In the system, input is the perception level and output is the global map level. Local

Figure 6.2: The diagram of the system.

map level is an intermediate level that acts as communication nodes and single robot fusion master. The benefits of introducing the local map level comes at following advantages, firstly: the size of the fused local map is smaller than the summation of individual sensor maps, which reduce the communication burden; secondly, the fused map on single robot level helps the localization and exploration of single robot, where the maps produced by different sensors are fused to compensate the limitation of each types of sensor; thirdly, map fusion happens at perception level brings the computational complexity of $\mathcal{O}(R^2S^2)$, while the local map level reduces it to $\mathcal{O}(R^2 + RS^2)$. Then, the multi-level framework is factorized into estimating local fused map M_r given the maps of perception level $m_{r,s}$ and estimating fused global map $M_{\mathcal{R}_r}$ conditional on available local maps M_r .

$$\underbrace{\prod_{r \in \mathcal{R}} p(m_r, T_{S_r} | m_{r,1:n_s})}_{\text{local map level}} \cdot \underbrace{p(M_r, T_{r,\mathcal{R}_r} | m_r, m_{\mathcal{R}_r})}_{\text{global map level}} \quad (6.1)$$

Figure 6.3: Top: Robot platform and the fused 3D map created using the two robots. Bottom: The diagram of the group of robots.

For local map level, each individual robot r is assumed to estimate its trajectories based on local observations and generates its partial map m_r^s , then robot r fuses the sensor maps $m_r^{1:n_s}$ to generate M_r . For global map level, robot r receives the set of local maps $M_{\mathcal{R}_r}$ from nearby robots, where $M_{\mathcal{R}_r} = \{M_{r_n}, r_n \in \mathcal{R}_r\}$.

6.2.3 Map Fusion Definition and Formulation

Problem 2 *For both local and global robot levels, the problem of map fusion problem is an joint probability estimation of relative transformation and merged map. Assuming the relative transformation estimation and map merging are conditionally independent, the map fusion problem is factorized into estimating a set of relative transformation (map matching) and generating the global map conditioned on all*

available partial maps (map merging).

To approximate the joint distribution, both local and global map fusion in Eq. (6.1) can be factorized into a product of relative transformation posterior and the fused map posterior. To simplify the derivation, only global map level is derived in detail, where it is the same for the local level. Then, Eq.(6.1) is written as follows:

$$p(M_r^{\mathcal{R}_r}, T_{r, \mathcal{R}_r} | m_r, m_{\mathcal{R}_r}) = \underbrace{p(T_{r, \mathcal{R}_r} | m_r, m_{\mathcal{R}_r})}_{\text{map matching}} \cdot \underbrace{p(M_r^{\mathcal{R}_r} | T_{r, \mathcal{R}_r}, m_r, m_{\mathcal{R}_r})}_{\text{map merging}} \quad (6.2)$$

This decouples multi-robot map fusion problem into two separate estimation problems. The key advantage of this factorization is summarized as follows: Firstly, it benefits from using MAP estimation for robots relative transformation estimation, making it possible to apply probabilistic nonlinear least square minimization models. Secondly, it benefits from utilizing a time-sequential relative entropy filter based map merging algorithm. Thirdly, the output of map matching result can be used to propagate the uncertainty for map merging, which tightly concatenates the two sub-problems.

6.3 Multi-Robot Map Matching

In this section, a general framework is presented for map matching. To address the challenge of matching maps produced by heterogeneous sensors, the estimation of relative transformation based on expectation maximization is proposed.

6.3.1 Mathematic Formulation

For robot r , it receives partial maps sequentially, which means the relative transformation is a serial processing upon receiving of new partial maps. Since the relative

transformation T_{r,r_n} is also pairwise independent of each pairs of robots, the posterior of relative transformation $p(T_{r,\mathcal{R}_r}|m_r, m_{\mathcal{R}_r})$ can be factorized into a product of the posterior of relative transformation between each pair, as shown in Eq. (6.3):

$$p(T_{r,\mathcal{R}_r}|m_r, m_{\mathcal{R}_r}) = \prod_{r_n \in \mathcal{R}_r} p(T_{r,r_n}|m_r, m_{r_n}) \quad (6.3)$$

where $T_{r,r_n} \in SE(3)$ is a rigid transformation with rotation $R_{r,r_n} \in SO(3)$ and translation $t_{r,r_n} \in R^3$. Instead of simply considering T_{r,r_n} as a fixed unknown parameter, it is modeled as a 6D random variable. Then, T_{r,r_n} can be modeled as a multivariate Gaussian distribution and solved by MAP estimation, where $T_{r,r_n} \sim (\hat{T}_{MAP}, \hat{\Sigma}_{MAP})$. To solve the MAP estimation problem, Bayes rule is applied to factorize Eq. (6.3) into Eq. (6.4), which is the product of maximum likelihood estimation and prior estimation:

$$p(T_{r,r_n}|m_r, m_{r_n}) = p(m_r|T_{r,r_n}, m_{r_n}) \cdot p(T_{r,r_n}), \quad (6.4)$$

where term $p(m_r|T_{r,r_n}, m_{r_n})$ is a common maximum likelihood estimation (MLE) problem that aims to find the most likely relative transformation T_{r,r_n} by matching the two partial maps m_r and m_{r_n} . By specifying the initialization, prior information is incorporated to evaluate the posterior even with a poor initial input, which can be given by feature descriptor correlation. If no prior information is available, the prior distribution is assumed to be a uniform distribution. In this case, $\hat{T}_{MAP} = \hat{T}_{MLE}$, where MLE is a special case of MAP estimation. To establish voxel-wise correspondences for map merging, the dense scan-matching technique is applied to solve MLE.

6.3.2 3D Occupancy Map Matching

For dense registration of two maps m_r and m_{r_n} , the common solution is to fix the map of current robot m_r as model map, and the neighboring map m_{r_n} as scene

map, where the rigid transformation T_{r,r_n} is estimated by transforming m_{r_n} to the coordinate of m_r and minimizing the error metric iteratively. To estimate T_{r,r_n} , data association needs to be estimated firstly, which is usually addressed by performing nearest neighborhood search of Euclidean distance or Mahalanobis distance. For pure registration purpose, the matches are not parameters of interest and are usually ignored after registration.

In our previous work [25], a novel map registration algorithm is applied called occupancy iterative closest point(OICP), which establishes voxel-to-voxel correspondences between all overlapped partial maps. Here, the data association issue is further exploited with different cases. For the partial map, $m_r = \{m_r^i\}_{i=1}^{N_{m_r}}$, $m_{r_n} = \{m_{r_n}^j\}_{j=1}^{N_{m_{r_n}}}$, where N_{m_r} and $N_{m_{r_n}}$ are the number of voxels of partial maps. Since the voxel-wise correspondences are usually assumed to be independent, the probability density function of $p(m_r|m_{r_n}, T_{r,r_n})$ can be factorized into Eq.(6.5).

$$p(m_r|m_{r_n}, T_{r,r_n}) = \prod_i p(m_r^i|m_{r_n}^{c_i}, T_{r,r_n}) \quad (6.5)$$

where $m_{r_n}^{c_i} = \{m_{r_n}^j, j \in c_i\}$ is the set of candidate corresponding voxels of m_r^i . The problem of calculating the corresponding points $m_{r_n}^{c_i}$ of m_r^i is often referred as data association. To estimate the hidden data association, Expectation-maximization (EM) is introduced, which is an algorithm well-suited for models containing latent variables and estimating them iteratively by marginalizing the likelihood function.

There are two steps for EM: the E-step efficiently estimates the hidden variables by evaluating the expectation, while the M-step solves the rigid transformation based on E-step using optimization algorithms. Here, we define a binary matching matrix J to represent the estimation of hidden data association between the corresponding voxels, where $J_{i,j} = 1$ if m_r^i corresponds to $m_{r_n}^j$ and equals 0 otherwise. For each row i , the matched voxels of m_r^i is the set of $J_{i,c_i} = \{J_{i,j} = 1, j \in c_i\}$. The EM form

Figure 6.4: The density map and histogram for the two sensors.

of Eq.(6.5) is rewritten as:

$$\begin{aligned}
 p(m_r | m_{r_n}, T_{r,r_n}) &= \sum_J p(m_r, J | m_{r_n}, T_{r,r_n}) \\
 &= \prod_i \sum_{c_i} p(m_r^i, J_{i,c_i} | m_{r_n}^{c_i}, T_{r,r_n}) \\
 &= \prod_i \sum_{c_i} \underbrace{p(J_{i,c_i} | m_r^i, m_{r_n}^{c_i}, T_{r,r_n})}_{\text{E step: data association}} \cdot \underbrace{p(m_r^i, J_{i,c_i} | T_{r,r_n}, m_{r_n}^{c_i})}_{\text{M step: optimization}} \quad (6.6)
 \end{aligned}$$

The negative log-likelihood to be minimized with EM is given by

$$f(T_{r,r_n}) = \min_{T_{r,r_n}} \sum_i \sum_{c_i} p(J_{i,c_i} | m_r^i, m_{r_n}^{c_i}, T_{r,r_n}) \cdot \ln p(m_r^i, J_{i,c_i} | T_{r,r_n}, m_{r_n}^{c_i}) \quad (6.7)$$

6.3.3 E-step

- **Homogeneous Sensors:** Since the density of map is similar for the same type of sensors, one-to-one hard assignment is established for computational efficiency. For each voxel m_r^i , there will be one and only one voxel $m_{r_n}^j$ correspond to it, where $J_{i,c_i} = J_{i,j}$.

$$\forall m_r^i \in m_r, \exists! m_{r_n}^j \in m_{r_n} \Rightarrow J_{i,j} = \omega_{ij}$$

- **Heterogeneous Sensors:** If heterogeneous sensors need to be aligned, one-to-one matching might fail due to wrong matches or insufficient correspondences. Hence, multiple soft correspondences should be established, where each voxel m_r^i corresponds to multiple voxels in $m_{r_n}^{c_i}$

$$\forall m_r^i \in m_r, \exists m_{r_n}^{c_i} \in m_{r_n} \Rightarrow J_{i,c_i} = \{J_{i,j} = \omega_{ij}, j \in c_i\}$$

The problems comes at two hands: (1) automatically determining the policy for the candidate correspondences searching. If it's the case of heterogeneous sensors, how to determine the value of c_i to balance the efficiency and accuracy. (2) For heterogeneous sensors, the candidate $J_{i,j} \in J_{i,c_i}$ should have different weight to contribute to the objective function.

As can be seen in Fig.6.4, the density of rgb-d camera is higher than laser scanner. In addition, the histogram of rgb-d map complies a Gaussian distribution, which means the density is distributed more evenly. However, the histogram of laser is more like the Gamma distribution. Based on the observation, a local density map based automatic tuning strategy is proposed for probabilistic data association, as shown in Algorithm 4. For the given maps, the algorithm firstly generates their density maps (depicted in Fig.6.4). If the mean of the density maps are similar, one-to-one correspondence is applied. If not, the numbers of correspondences (c_i) to be established are determined by the ratio of local density value. As shown

Figure 6.5: Density based multiple data association strategy.

in Fig.6.5, the density of rgbd-d map is distributed evenly, the numbers of voxels matched increases with the decreases of laser map's density. Each voxel in laser map matches different value to c_i in rgb-d map, where the histogram and density map shows the automatically tuned value of c_i .

6.3.4 M-step

Solving M-step is to minimize the log-likelihood of Eq.(6.6) based on the data association established in E-step. Here, probabilistic weights are assigned to each pair of correspondences. There are several different ways of updating the rotation and

translation, for example, Levenberg-Marquardt, single value decomposition.

$$f(T_{r,r_n}) = \underset{i}{\operatorname{argmin}} \sum_i \sum_j^{c_i} \omega_{i,j} \cdot \ln p(m_r^i, J_{i,j} | T_{r,r_n}, m_{r_n}^j) \quad (6.8)$$

Algorithm 4 Probabilistic Map Matching

Require: The maps $m_r = \{m_r^i\}_{i=1}^{N_{m_r}}$, $m_{r_n} = \{m_{r_n}^j\}_{j=1}^{N_{m_{r_n}}}$, and initial transformation fuction T_0

Ensure: Iteratively minimize the error function

1. generate density map d_r for m_r and d_{r_n} for m_{r_n}

for k from 1 to termination criteria **do**

for i from 1 to N_{m_r} criteria **do**

if $\operatorname{round}(\frac{\operatorname{mean}(d_r)}{\operatorname{mean}(d_{r_n})}) = 1$ **then**

 2. one-to-one matching, where $J_{i,c_i} = J_{i,j}$

else

 3. multiple data association, where $c_i = \operatorname{round}(\frac{d_r^i}{d_{r_n}^j})$

end if

 4. assign weight: $\omega_{ij} = \exp^{-(T^k * m_r^i - m_{r_n}^j)}$, $j \in c_i$

 5. iteratively optimize the error function: $\hat{T}_{r,r_n}^{k+1} = \underset{i}{\operatorname{argmin}} \sum_i \sum_j^{c_i} \omega_{i,j} \cdot \ln p(m_r^i, J_{i,j} | T_{r,r_n}^k, m_{r_n}^j)$

end for

end for

6.4 Time-Sequential Map Merging

After the relative transformation is derived, the partial map m_{r_n} is transformed to coordinate frame of m_r by transformation function $\Phi(T_{r,r_n}, m_{r_n})$, to generate ${}^r m_{r_n} = \Phi(T_{r,r_n}, m_{r_n})$. The process of combing the information of common objects from partial maps to form an enhanced global map is usually referred to as map merging and is formulated in Eq. (6.9):

$$p(M_r^{\mathcal{R}_r} | T_{r,\mathcal{R}_r}, m_r, m_{\mathcal{R}_r}) = p(M_r^{\mathcal{R}_r} | m_r, \Phi_{r_n \in \mathcal{R}_r}(T_{r,r_n}, m_{r_n})) \quad (6.9)$$

During real missions, maps are generated and transmitted sequentially. This results

in robot r receiving its neighboring robots' maps in some permutation π of $\mathcal{R}_r$. Hence, the map merging can be performed serially, where reception of a new partial map would trigger a pair-wise map merging. Then, Eq. (6.9) can be factorized into Eq. (6.10).

$$p(M_r^{\mathcal{R}_r} | m_r, \Phi_{r_n \in \mathcal{R}_r}(T_{r,r_n}, m_{r_n})) = \prod_{r_n=1}^{N_r} p(M_r^{\pi(1,r_n)} | M_r^{\pi(1,r_n-1)}, \Phi(T_{r,\pi(r_n)}, m_{\pi(r_n)})) \quad (6.10)$$

Note that $M_r^{\pi(1,r_n)}$ is the partial global map of robot r generated by fusing the maps of robots $\pi(1)$ to $\pi(r_n) \in \mathcal{R}_r$ with m_r after the reception of map $m_{\pi(r_n)}$ from neighboring robot $\pi(r_n) \in M_r^{\mathcal{R}_r}$. The final global fused map $M_r^{\mathcal{R}_r}$ of robot r , can be retrieved after the end of this serial process, hence $M_r^{\mathcal{R}_r} = M_r^{\pi(1,N_r)}$.

6.4.1 Uncertainty Propagation and Transformation

In general, there are two types of uncertainty need to be integrated when performing map merging.

1) *Map uncertainty*: in real environment, uncertainties will occur due to localization error and sensor measurement noise. In map level, those uncertainties are represented with the occupancy probability value. Since the same object can be observed in different perspectives by different robots, voxels representing this object could have different occupancy probabilities in separated maps. Hence, it is vital to consider these occupancy probability dissimilarities when fusing them into a global map.

2) *Relative transformation uncertainty*: the error of the robots relative position estimation. This uncertainty is related to the estimation results, the more accurate the estimation the smaller the uncertainty. This type of uncertainty also needs

to be considered in map merging process. The relative estimation uncertainty is propagated by the map matching algorithm and will be represented as a covariance matrix.

Our approach is to propagate the uncertainty for both single robot and multi-robot level, then the corresponding probability distributions are merged to output an improved estimate of the observed object. The assumption for the mathematic is based on that sensor errors are independent and complies a normal distribution. Assume that $q = [q_x \ q_y \ q_z \ L_q]$ denotes the position of the extracted voxel center $[q_x \ q_y \ q_z]$ and the occupancy probability L_q . The mean of q is $\mu_q = [\mu_{q_x} \ \mu_{q_y} \ \mu_{q_z}]$, the covariance is a diagonal matrix with the same entry $\Sigma_q = \text{diag}[\sigma_{q_x q_x}^2 \ \sigma_{q_y q_y}^2 \ \sigma_{q_z q_z}^2]$. Voxel q is transformed by the following function to voxel r .

$$r = f(\alpha, \beta, \gamma, t_x, t_y, t_z, q_x, q_y, q_z) = Rq + t \quad (6.11)$$

where α, β, γ are the rotation angles and t_x, t_y, t_z are the translation elements. Since the transformation function $f(\cdot)$ is nonlinear, then the transformed voxel r will be non-gaussian. To approximate r with Gaussian, the transformation $f(\cdot)$ is linearized using the first-order Taylor expansion:

$$p(r) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp \left\{ -\frac{1}{2}(r - \mu_r)^T \Sigma_r^{-1} (r - \mu_r) \right\} \quad (6.12)$$

$$\sim \mathcal{N}(r; \mu_r, \Sigma_r) \quad (6.13)$$

Where μ_r can be directly calculated using Eq.(6.11). Assume the covariance of transformation is $\Sigma_f = \text{diag}[\sigma_{t_x t_x}^2 \ \sigma_{t_y t_y}^2 \ \sigma_{t_z t_z}^2 \ \sigma_{\alpha\alpha}^2 \ \sigma_{\beta\beta}^2 \ \sigma_{\gamma\gamma}^2]$. Through deriving the Jacobian of transformation function, the covariance of transformed voxel Σ_r is estimated in Eq.(6.14).

Figure 6.6: The process of uncertainty propagation and merging.

$$\begin{aligned}
 F_{\alpha\beta\gamma \ t_x t_y t_z} &= \frac{\partial f(\cdot)}{\partial(\alpha, \beta, \gamma, t_x, t_y, t_z)}, \quad F_{q_x q_y q_z} = \frac{\partial f(\cdot)}{\partial(q_x, q_y, q_z)} \\
 \Sigma_r &= \underbrace{F_{\alpha\beta\gamma \ t_x t_y t_z} \Sigma_f F_{\alpha\beta\gamma \ t_x t_y t_z}^T}_{\text{transformaiton uncertainty}} + \underbrace{F_q \Sigma_q F_q^T}_{\text{map uncertainty}}
 \end{aligned} \tag{6.14}$$

6.4.2 Uncertainty merge

We employ our previous work on map merging [25] to execute the sequential pairwise uncertainty merging. For the two types of uncertainty, the merging process should preserve all valuable information of the partial maps while decreasing the uncertainty of the fused map. Once unidirectional alignment of maps is performed in the map matching stage to establish the correspondence, a relative entropy filter based on Kullback Leibler divergence is applied to merge map measurements that decreases the total uncertainty of the global map. We apply a three-dimensional statistical approach, where any given multi-dimensional Gaussian distributions can be

integrated by applying matrix operations. Let $\mathcal{N}_x(\mu, \Sigma)$ denote the density function of voxel x , then,

$$\mathcal{N}_c(\mu_c, \Sigma_c) \sim \mathcal{N}_r(\mu_r, \Sigma_r) \cdot \mathcal{N}_p(\mu_p, \Sigma_p) \quad (6.15)$$

Where,

$$\mu_c = (\Sigma_a^{-1} + \Sigma_b^{-1})^{-1}(\Sigma_r^{-1}\mu_r + \Sigma_p^{-1}\mu_p) \quad (6.16)$$

$$\Sigma_c = (\Sigma_r^{-1} + \Sigma_p^{-1})^{-1} \quad (6.17)$$

Then the steps are extended to the entire voxels in the map, where Algorithm 5 and Fig.6.6 show the overall process. Firstly, the source voxel q and target voxel p is represented as a 3D gaussian distribution. In line 2-4, q is transformed to r according to the given transformation function T , where covariance of transformed voxel is calculated using Eq.(6.14). Then the final merging happens in line 5 and the uncertainty of both source and target map is fused using Eq.(6.15). The results shows the merging process decreases the uncertainty, and the merged voxel is closer to the one with less uncertainty.

Algorithm 5 Uncertainty Propagation and Map Merging

Require: The maps m_r and $m_{r'}$, and transformation fuction T

Ensure: Transform the map m_r and fuse them

for all voxels q **do**

1. represent the source voxel as multi-gaussian distribution: $\mathcal{N}_q(\mu_q, \Sigma_q)$
2. transform the voxel q to r : $\mu_r = T(\mu_q)$
3. calculate Σ_r using Eq.(6.14)
4. represent the transformed voxel r with $\mathcal{N}_r(\mu_r, \Sigma_r)$
5. merge r with corresponding voxel p to get voxel c : $\mathcal{N}_c(\mu_c, \Sigma_c) \sim \mathcal{N}_r(\mu_r, \Sigma_r) \cdot \mathcal{N}_p(\mu_p, \Sigma_p)$

end for

(a) NTU Indoor

(b) NTU Mixed

(c) Changi Centre

(d) NTU Forest

Figure 6.7: The experiment setting up: NTU Indoor, NTU Mixed, Changi Centre, NTU Forest respectively.

6.5 Experimental Results

The performance of the proposed multi-robot map fusion framework is validated through extensive real world experiments. Both qualitative and quantitative results are presented. The former are obtained by visual inspection of the fused map in different experiment settings. Then, the latter are expressed by calculating the root mean square error and the running time.

Table 6.1: Summary of the experiment environment.

Dataset	Sensor	Location	Dynamics	Type
NTU Indoor	Asus Xtion&Velodyne	Indoor	Dynamic	Structured
NTU Mixed	Asus Xtion&Velodyne	Indoor&Outdoor	Static	Semi-Structured
Changi Centre	Velodyne	Indoor	Dynamic	Semi-Structured
NTU Forest	Velodyne	Outdoor	Static	Unstructured

6.5.1 Evaluation Protocol

Experiment Environment: In this chapter, four real environment experiments are conducted, where two robots are equipped with heterogeneous sensors. The resolution of the 3D occupancy grid map was set to be 0.1m. The communication between robots was established by long range Wi-Fi with limited bandwidth. The experiment settings are:

a)NTU Indoor : two Husky robots equipped with ASUS Xtion Camera and & Velodyne 3D Lidar in an indoor environment of Nanyang Technological University.

b)NTU Mixed : two Husky robots equipped with ASUS Xtion Camera and & Velodyne 3D Lidar in the indoor & outdoor mixed environment of Nanyang Technological University.

c)Changi Centre: two Husky robots equipped with Velodyne 3D Lidar during Asia Unmanned Exhibition in Changi Centre.

d)NTU Forest: two Husky robots equipped with Velodyne 3D Lidar in an open forest of Nanyang Technological University.

Fig. 6.7 and Tab. 6.1 summarize the setting and the main characteristics of each experiment.

Error Metric: The error metric applied is MSE (Mean Squared Error) , which is the mean squared Euclidean distance between the matched pairs.

Comparison Baseline: For each experiment, two matching algorithms are tested. First, the rigid transformation is computed with fixed one-to-to data association correspondence approach, i.e, Occupancy Iterative Closet Point proposed in Chapter 3. Second, the proposed probabilistic map fusion algorithm is applied to compare the results.

(a) At $t=0s$, velodyne map is very sparse. Left image shows the successful registration of multiple probabilistic data association (average associated voxels=3.57), while right shows the large offset between two maps by applying one-to-one correspondence.

(b) At $t=20s$, velodyne map gets denser. Both algorithms produce satisfying algorithm. It can be seen that the density of velodyne map is increasing. Thus, the average associated voxels of multiple data association strategy decrease to 2.15).

Figure 6.8: The experiment of heterogeneous map matching with different data association strategies.

6.5.2 NTU Indoor

The indoor environment is the second floor of a building in the university which contains challenging objects such as reflective glasses and moving people (see Fig. 6.7(a)). In order to test the algorithm's flexibility of using different sensors with varying characteristics, two robots were teleoperated to explore the environment with different sensors (Asus Xtion PRO and Velodyne VLP-16). The map fusion is implemented at two levels. First, the dense map produced by Asus Xtion and sparse map produced by Velodyne are fused to generate the local fused map. Then, the local fused

Figure 6.9: A demonstration of multi-level map fusion result in NTU indoor experiment.

map was sent to the neighbor robot to generate the global fused map.

In the local level, the Velodyne map is very sparse in the initial position (See Fig. 6.8(a)), while the Asus Xtion produced dense map at the same time. The matching results of the sparse map and dense map presented in Fig. 6.8(a) illustrates the superiority of the multiple data association strategy. At $t = 0$, each voxel in the sparse map correlates average 3.47 voxels in the dense map to establish more robust correspondences. With the increasing density of the Velodyne map, the average correlated voxels dropped to 2.15 at $t = 20s$ and two algorithms produce comparable results. In summary, despite the heterogeneity of sensors and challenging environments, the proposed method is able to produce a more consistent map which merges the sparse 3D LiDAR map with the dense RGBD map (see Fig. 6.8(b)).

In the global map, the local map produced by two robots are fused into a global consistent map. As highlighted in Fig. 6.9 (top part), the RGB-D sensor didn't fully perceive the environment at some corner of the building (circled area). However, it still generated a much denser map compared to the sparse 3D LiDAR. Considering

Figure 6.10: A demonstration of multi-level map fusion result in NTU Mix experiment.

the LiDAR scans 360° , it records most of the environment without leaving any major blind spots, however, generates a less dense map owing to sensor's sparseness. As shown in Fig. 6.9 (Top: individual map produced by heterogeneous sensors. Middle: the locally fused map by two different robots. Bottom: the global map by fusing two local maps.), the overlapped area between the two maps combines the advantages of the two sensors, which produces a much more detailed and accurate fused map.

6.5.3 NTU Mixed Ground Floor

The mixed environment depicted in Fig. 6.7(b) is at the ground floor of a building in the university, where the indoor environment is the lobby, and the outdoor environment is the affiliated car park. Two robots were tele-operated to explore the environment with different sensors (Asus Xtion PRO and Velodyne VLP-16).

Fig. 6.10 shows the fusion of maps on local level and global level. Robot 1 generates map in an indoor environment, where the Asus Xtion PRO captured most of the

information in the lobby. As highlighted in Fig. 6.10 (Top: individual map produced by heterogeneous sensors. Middle: the locally fused map by two different robots. Bottom: the global map by fusing two local maps.), the Kinect map of robot 2 missed the wall due to the limited perception range of the sensor. The velodyne map of robot 2 produces a sparse map, especially for the ground, which is due to the sensor limitation. As underlined with circles, the algorithm combines the map information from two heterogeneous maps to generate an enhanced and more consistent global map.

6.5.4 Changi Exhibition Center

The Changi Center depicted in Fig. 6.7(c) is at a convention center during the Asia Unmanned System Exhibition in Singapore, where the red arrow and trajectory is robot 1, the black represents robot 2. Two robots were utilized with the same configuration as the experiment in the mixed ground floor environment. The robots start their mapping from a common neighborhood, and explore the environment by traversing two different paths. It differs from previous two experiments since it also equips Velodyne 3D Lidar. As shown in Fig. 6.11, the fusion starts from very sparse Velodyne maps to denser Velodyne maps, where the fusion process requires the data association to dynamically adjust the numbers of voxels to be associated. In the initial position, a more thorough and extensive correspondences will be established to calculate relative transformation. Then, the average numbers of voxels to be associated is decreased as the maps grow denser. Finally, the two local maps are fused into a more consistent and enhanced global map. More specifically, the global map filled the gaps in the partial maps of the robots occurred due to loss of data and sensor limitations.

(a) $t=0s$, the initial position of the two robots. Two sparse local maps are successfully fused into a global map.

(b) $t=76s$, half of the mission. The local maps become denser, and global map are fused.

(c) $t=145s$, the end of the mission. The two robots back to the start position and the global map of the full environment is presented.

Figure 6.11: A demonstration of the collaborative mapping system in real-time at exhibition center.

6.5.5 NTU Forest Environment

Experiments conducted in the forest environment is presented in this part. Each robot was equipped with a Velodyne VLP-16 Lidar. The forest is in the university (NTU) with full 3D environment that contains trees and slope (see Fig. 6.7(d)). It differs from previous three experiments since it is the fully unstructured environment. The two robots started from nearby place, while robot 1 moved straight

(a) $t=0s$, the initial position of the two robots. Two sparse local maps are successfully fused into a global map.

(b) $t=120s$, half of the mission. The local maps become denser, and global map are fused.

(c) $t=215s$, the end of the mission. The global map of the full environment is presented.

Figure 6.12: A demonstration of the collaborative mapping system in real-time at NTU forest.

forward and robot 2 climbed to a small hill and turned back to the start position. As presented in Fig. 6.12, the map produced by robot 1 is semi-structured, while robot 2 produces fully unstructured forest map. The partial maps are fused into a consistent global map. The algorithm combines the map information from partial maps to generate an enhanced and more consistent global map.

Table 6.2: The root mean error in all experiments. (The best performance is denoted in bold.)

Datasets	Beginning		Middle		Final	
	OICP	Ours	OICP	Ours	OICP	Ours
NTU Indoor	0.3414	0.2479	0.2592	0.2357	0.2675	0.2890
NTU Mixed	0.3140	0.2791	0.3078	0.2563	0.2818	0.2542
Changi Center	0.4405	0.4284	0.4054	0.3998	0.3653	0.3550
NTU Forest	0.4393	0.4301	0.3561	0.3374	0.3430	0.3237

6.5.6 Analysis of Experiment Results

In this subsection, the quantitative results in respect of accuracy and efficiency are presented.

Tab.6.2 summarizes the RMS error for all experiments. It can be seen The trend is clear that the smallest errors in Tab.6.2 exhibit the highest performance of transformation estimation. During the collaborative mapping process, the RMS errors for all experiments are decreasing. All the algorithms witness a large offset at the beginning of the experiments, which is caused by the low density of the map. As the maps became denser, more correspondences between the voxels can be established and the errors will decrease. In can be also found that the RMS errors in Changi Center and NTU Forest are higher compared with NTU Indoor and NTU Mixed. The reason is that Changi Center and NTU Forest are highly unstructured environment, while NTU Indoor and NTU Mixed are more structured.

Our methods outperforms OICP in all of the cases. In NTU Indoor and NTU Mixed experiments, two robots were equipped with heterogeneous sensors, our algorithm dynamically adjust the data association strategy in the matching (See Fig.6.13). As a consequence of this strategy, the RMS error of our algorithm improved significantly of 10-20% compared to the one-to-one strategy OICP in the two scenarios. In unstructured Changi Center and NTU Forest, only Velodyne 3D Lidar was equipped due to the sensor limitation of Asus Xtion Camera. In these two experiments, the numbers of voxels to be associated are close to the one-to-one (See Fig.6.13). Both

Figure 6.13: The average numbers of voxels to be associated in our algorithm during the four experiments.

Figure 6.14: The processing time during the four experiments.

OICP and our algorithm achieve comparable convergence results and our algorithm still outperforms OICP. Overall, our algorithm demonstrates superior or comparable performance over OICP.

Additionally to the evaluation of the error metrics, computation times are an important factor in the real-time robotics applications. Fig. 6.14 summarizes the computational times of the four experiments when executed on an Intel Core i7-6700HQ CPU @ 2.60GHz robot onboard computer. The efficiency is reported per full calculation time of the map fusion process of every running. It can be found that all experiments were running in semi-real time. The computational time for NTU Forest is significantly higher than other scenarios due to the large scale and unstructured environment. For collaborative mapping missions, we are considering the strategy of local maps from individual robots being fused sporadically. Considering connectivity/bandwidth and computational constraints, transferring map data only when local maps are significantly updated is preferred over transferring every simple map update. Therefore, the time interval between two such fusion processes is usually in the order of minutes, where the map fusion process does not operate at high frequency. Hence execution time of several seconds (i.e. semi-real time) for map fusion is justified.

6.6 Conclusions

In this chapter, a general probabilistic framework for multi-robot 3D mapping using heterogeneous sensors is proposed, which is independent of sensor types and SLAM algorithms. The framework provides theoretical basis for computing the relative transformations (map matching) and merging probabilistic map information (map merging), which enables the team of robots to work with different sensor types on different platforms. Since maps produced by heterogeneous sensors poss diverse physical properties, a probabilistic map matching algorithm is proposed that auto-

matically decides the numbers of voxels to be associated. To distribute the maps among robots, a time-sequential map merging framework that makes fusing maps from distributed multi-robot system efficiently, where uncertainty embedded in the map and caused by the relative transformation are propagated and integrated. The proposed approach is evaluated in real-time experiments in various environments with heterogeneous sensors. The results verify the accuracy, efficiency and generality for collaborative mapping missions.

Chapter 7

Conclusions and Future Work

This chapter concludes this thesis by summarizing the main contributions and suggesting directions for future research.

7.1 Conclusions

Utilizing a group of robots for environmental mapping have found significant attention due to the inherent advantages of robustness and efficiency. This thesis aims to develop systematic probabilistic fusion framework for collaborative robots 3D mapping. The thesis first proposes the probabilistic point set registration algorithm to match and merge 3D maps. In addition, to detect the inconsistency during collaborative mapping, submap-based inconsistency detection strategy is proposed to evaluate the misalignment error on local and global map levels. In structured environment, a novel hierarchical probabilistic map fusion framework is proposed to combine the high level geometric features and low level voxels, which improves the convergence basin and robustness. For robots equipped with heterogeneous sensors in unstructured environments, a general probabilistic framework for multi-robot 3D mapping is validated in various scenarios, which is independent of sensor types and

SLAM algorithms.

The main contributions and conclusions of this thesis are summarized as follows:

1. *Probabilistic point set registration approach for fusion of 3D occupancy grid maps* (Chapter 3).

In mapping process, 3D occupancy grid maps are generated by compressing the raw sensor data. The proposed occupancy iterative closet point (OICP) algorithm combines positional coordinate in conjunction with occupancy probability to increase the matching accuracy, where the local descriptor is augmented from 3D space to 4D space. The weight between positional error and map uncertainty is controlled by modeling errors using Shannon entropy and positional misalignment. As the proposed OICP doesn't guarantee global minima, the chapter has applied environment measurement model to evaluate the quality of the transformation, where the mismatched results are rejected. The final map fusion is then achieved with a relative entropy filter which integrates the measurements of both maps and decreases the uncertainty of the global map. Experiments on simulated and real world environments were conducted and the evaluations confirmed that the proposed OCIP method is able to produce more consistent global 3D occupancy grid maps compared to existing methods in the literatures.

2. *Robust submap-based probabilistic inconsistency detection for multi-robot mapping* (Chapter 4).

In multi-robot mapping missions, fusing erroneous local maps results in an inconsistent global map, which renders the accurate execution of autonomous tasks infeasible. Therefore detecting inconsistency in both local maps and fused global maps is vital for multi-robot missions. Chapter 4 has proposed a multi-robot map inconsistency detection strategy to evaluate the inconsistency on local and global levels. The inconsistency is measured by modeling the submap-wise misalignment error metric distribution into a truncated Gaussian distribution and is verified by applying a

map registration algorithm. The proposed algorithm successfully detects the inconsistency in the challenging indoor and outdoor environments. More importantly, the inconsistency of different types (ie. sequential, local and global) are detected with a probability distribution, which can be the uncertainty of post-processed optimization or the signal of resetting mapping mission. The accuracy of the proposed method is evaluated using submaps from challenging environments in both indoor and outdoor, which illustrates its usefulness and robustness for multi-robot mapping tasks.

3. Hierarchical probabilistic fusion framework for matching and merging of 3D occupancy maps (Chapter 5).

Most of the map matching algorithms belong to the scope of iterative dense registration, which is designed for raw sensor data registration on point/voxel-wise level. Efficient and robust map fusion algorithm can be applied in structured environments is very desired, where geometric features can be combined with voxel coordinates. Chapter 5 has studied the hierarchical probabilistic fusion framework for 3D occupancy maps, which is factorized into three modules: uncertainty modeling, map matching, and map merging. The local uncertainty of the map is modeled by combining the individual voxel and structural edge covariances. Based on the uncertainty modeling, two-level Probabilistic Map Matching (PMM) algorithm is applied to calculate the relative transformation accurately. In the PMM, it first combines both high-level structural edge and low-level occupancy voxel features. Then, the structural uncertainty is used to generate a coarse matching between the maps and its result is then used to improve the voxel level map matching, resulting in a more efficient and accurate matching between maps with a larger convergence basin. The relative transformation output from PMM algorithm is evaluated based on the the overlap between the two maps. Experiments on simulated and real-world environments are conducted, and the evaluations confirm the utility of the proposed approach in generating more accurate relative transformation with a faster rate and a larger convergence basin. Although only the combination of point and edge fea-

tures has been discussed, the combination of other feature types is also possible under the hierarchical framework.

4. *A general probabilistic framework for collaborative 3D mapping using heterogeneous sensors* (Chapter 6).

For a group of robots in arbitrary (structured/semi-structured/unstructured) environments, maps generated with heterogeneous sensors need to be combined with a comprehensive information content. The varying difference of density, range and field of view of heterogeneous sensors brings the challenge for collaborative mapping. Chapter 6 has provided a general probabilistic framework for collaborative 3D mapping, which is independent of sensor types and SLAM algorithms. The framework provides the flexibility of solving map matching and map merging problem separately, which allows the team of robots to work with heterogeneous sensor types. In the thesis, the Expectation-maximization (EM) based map matching algorithm is successfully applied to register 3D Lidar map and Kinect map, where the numbers of voxels to be associated is automatically decided based on the map characteristics. To distribute the maps among robots, the uncertainties of relative localization estimation and the local map error have been propagated and integrated. Then, time-sequential map merging framework is utilized to distributively transfer the local maps between the teams of robots. The proposed approach is evaluated in real-time experiments in various environments with heterogeneous sensors, such as forest, dynamic and mixed indoor-outdoor environments. The results verify the accuracy, efficiency and generality for collaborative mapping missions. The proposed mapping framework can be utilized for complexed environment mapping for real time multiple mobile robots applications.

To conclude, this thesis has developed a systematic probabilistic fusion framework for collaborative robots 3D mapping. The thesis provides comprehensive theoretical formulations, algorithmic development, and experimental verification via simulation and realtime experiments. Although collaborative mapping has been investigated

in depth in this thesis, there are still several interesting research problems that are worthy to be studied.

7.2 Future Work

1. In chapter 3, the OICP map matching algorithm belongs to the scope of the iterative optimization algorithm, which tends to be trapped into a local minimum. Thus, more improvements on finding the initial inputs should be explored. For example, GPS information can also be fused to support the initial estimation if available. Since an erroneous initial input would further undermine the accurate matching, hypothesis testing strategy to reject the wrong estimation should be developed. In addition, the availability of a consistent global map requires periodic fusion of partial maps from multiple robots. Hence making a decision on when to fuse the maps warrants further research and would also be explored in the future.

2. The EM-based based map matching algorithm is suitable for the same type of robots equipped with heterogeneous sensors (Chapter 6). For hybrid teams robots (UGV and UAV), the large difference in viewpoint and robot kinematics brings the challenge of fusion different sensor modalities generated by heterogeneous robots. The difficulty comes at finding the corresponding regions that allow an accurate relative localization between the heterogeneous robots. Low level features like lines and points are often view-point dependent and subject to failure in repetitive environments. With the fast developing in deep learning, it is promising to conduct Convolutional Neural Networks-based approach to assign correspondence at semantic levels, which will increase the robustness and efficiency.

3. In Chapter 6, the robots only map the static environment and shares the 3D maps among the robots, where the moving dynamics such as pedestrians have been neglected. To obtain a more comprehensive of the surroundings with dynamics, deep learning algorithm can be applied detect and track the nearby human, where

the person who behaves abnormal or different from others in dynamic environments should be recognized. Then, the robots should actively plan their motion based on a high-fidelity model of the human dynamics, accounting for inertia, actuator limits and environment disturbances.

Author's Publications

Journal Papers:

1. **Yufeng Yue**, P.G.C.N. Senarathne, Chule Yang, Jun Zhang, Mingxing Wen, and Danwei Wang, "Hierarchical Probabilistic Fusion Framework for Matching and Merging of 3D Occupancy Maps" , *IEEE Sensors Journal*, vol. 18, no. 21, pp. 8933-8949, 1 Nov.1, 2018.
2. Yuanzhe Wang, Mao Shan, **Yufeng Yue**, and Danwei Wang, "Flexible Leader-Follower Formation Analysis and Tracking Control of Multiple Nonholonomic Mobile Robots Using Relative Pose Measurements", *IEEE Transactions on Control Systems Technology*, TCST.2019.2892031, 2019.
3. Chule Yang, **Yufeng Yue**, and Danwei Wang, "Probabilistic Reasoning for Unique Role Recognition Based on the Fusion of Semantic-Interaction and Spatio-Temporal Features", *IEEE Transactions On Multimedia*, TMM.2019.2875513, 2019
4. Chule Yang, Danwei Wang, Yijie Zeng, **Yufeng Yue**, and Prarinya Siritanawan, "Multimodal Information Fusion for Human-Oriented Context Awareness Using Mobile Robot", *Information Fusion*, Volume 50, Pages 126-138, 2019

Conference Papers:

1. **Yufeng Yue**, P.G.C.N. Senarathne, Chule Yang , Jun Zhang, Mingxing Wen, and Danwei Wang, "Probabilistic Fusion Framework for Collaborative Robots 3D Mapping", *21st International Conference on Information Fusion*, Cambridge, 2018.
2. **Yufeng Yue**, Chule Yang , P.G.C.N. Senarathne, Jun Zhang, Mingxing Wen, and Danwei Wang, "Online Collaborative 3D Mapping in Forest Environment", *IEEE International Conference on Robotics and Automation (ICRA) workshop*, Brisbane, 2018.
3. **Yufeng Yue**, Danwei Wang, P. G. C. N. Senarathne, and Chule Yang, "A General Framework for Distributed Multi-Robot 3-D Map Fusion," *IEEE International Conference on Robotics and Automation (ICRA) workshop*, Singapore, 2017.
4. **Yufeng Yue**, Danwei Wang, P. G. C. N. Senarathne, and Chule Yang, "Robust Submap-Based Probabilistic Inconsistency Detection for Multi-Robot Mapping," *2017 European Conference on Mobile Robotics (ECMR)*, Paris, 2017, pp. 1-6.
5. Chule Yang, Yijie Zeng, **Yufeng Yue**, Jun Zhang, and Danwei Wang, "Knowledge-Based Role Recognition by Using Human-Object Interaction and Spatio-Temporal Analysis", *2017 IEEE Conference on Robotics and Biomimetics (ROBIO)*, Macau, 2017, pp. 159-164. **Final List of Best Student Paper Award**
6. **Yufeng Yue**, Danwei Wang, P. G. C. N. Senarathne, and Diluka Moratuwage, "A hybrid probabilistic and point set registration approach for fusion of 3D occupancy grid maps," *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, 2016, pp. 1975-1980.

Bibliography

- [1] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i,” *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, June 2006.
- [2] S. Saeedi, M. Trentini, M. Seto, and H. Li, “Multiple-Robot Simultaneous Localization and Mapping: A Review,” *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.
- [3] R. N. Darmanin and M. K. Bugeja, “A review on multi-robot systems categorised by application domain,” in *2017 25th Mediterranean Conference on Control and Automation (MED)*, July 2017, pp. 701–706.
- [4] A. Khamis, A. Hussein, and A. Elmogy, *Multi-robot Task Allocation: A Review of the State-of-the-Art*. Cham: Springer International Publishing, 2015, pp. 31–51.
- [5] A. Gautam and S. Mohan, “A review of research in multi-robot systems,” in *2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS)*, Aug 2012, pp. 1–5.
- [6] A. Elibol, J. Kim, N. Gracias, and R. Garcia, “Efficient image mosaicing for multi-robot visual underwater mapping,” *Pattern Recognition Letters*, vol. 46, pp. 20 – 26, 2014.

- [7] J. Dong, E. Nelson, V. Indelman, N. Michael, and F. Dellaert, "Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5807–5814.
- [8] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar *et al.*, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.
- [9] A. T. Rashid, M. Frasca, and A. A. Ali, "Multi-robot localization and orientation estimation using robotic cluster matching algorithm," *Robotics and Autonomous Systems*, vol. 63, pp. 108 – 121, 2015.
- [10] M. Hussein, M. Renner, M. Watanabe, and K. Iagnemma, "Matching of ground-based lidar and aerial image data for mobile robot localization in densely forested environments," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1432–1437.
- [11] S. Saeedi, L. Paull, M. Trentini, and H. Li, "Neural network-based multiple robot simultaneous localization and mapping," *Neural Networks, IEEE Transactions on*, vol. 22, no. 12, pp. 2376–2387, 2011.
- [12] A. W. Stroupe, M. C. Martin, and T. Balch, "Distributed sensor fusion for object position estimation by multi-robot systems," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 2, 2001, pp. 1092–1098 vol.2.
- [13] A. Cunningham, V. Indelman, and F. Dellaert, "Ddf-sam 2.0: Consistent distributed smoothing and mapping," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 5220–5227.
- [14] A. Cunningham, K. M. Wurm, W. Burgard, and F. Dellaert, "Fully distributed scalable smoothing and mapping with robust multi-robot data association," in

- Robotics and Automation (ICRA)*, 2012 IEEE International Conference on, May 2012, pp. 1093–1100.
- [15] P. Trautman, J. Ma, R. M. Murray, and A. Krause, “Robot navigation in dense human crowds: Statistical models and experimental studies of humanrobot cooperation,” *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.
- [16] J. Gregoire, M. Čáp, and E. Frazzoli, “Locally-optimal multi-robot navigation under delaying disturbances using homotopy constraints,” *Autonomous Robots*, Dec 2017.
- [17] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, “Distributed multirobot exploration and mapping,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1325–1339, Jul 2006.
- [18] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, “Coordinated multi-robot exploration,” *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 376–386, 2005.
- [19] P. C. Niedfeldt, A. Speranzon, and A. Surana, “Distributed map fusion with sporadic updates for large domains,” in *2015 IEEE International Conference on Robotics and Automation*, May 2015, pp. 2806–2813.
- [20] T. M. Bonanni, B. D. Corte, and G. Grisetti, “3-D Map Merging on Pose Graphs,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1031–1038, April 2017.
- [21] D. Scaramuzza, M. C. Achtelik *et al.*, “Vision-Controlled Micro Flying Robots: From System Design to Autonomous Navigation and Mapping in GPS-Denied Environments,” *IEEE Robotics Automation Magazine*, vol. 21, no. 3, pp. 26–40, Sept 2014.

- [22] C. Brand, M. J. Schu, and H. Hirsch, "Submap matching for stereo-vision based indoor/outdoor SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2015, pp. 5670–5677.
- [23] Saeedi, Sajad and Paull, Liam and Trentini, Michael , "Map merging for multiple robots using Hough peak matching," *Robotics and Autonomous Systems*, vol. 62, no. 10, pp. 1408–1424, 2014.
- [24] S. Saeedi, L. Paull, M. Trentini, and Seto, "Group mapping: A topological approach to map merging for multiple robots," *Robotics & Automation Magazine, IEEE*, vol. 21, no. 2, pp. 60–72, 2014.
- [25] Y. Yue, D. Wang, P. Senarathne, and D. Moratuwage, "A hybrid probabilistic and point set registration approach for fusion of 3D occupancy grid maps," in *2016 IEEE International Conference on Systems, Man, and Cybernetics*, Oct 2016, pp. 1975–1980.
- [26] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The international journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.
- [27] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to non-linear systems," in *AeroSense'97*. International Society for Optics and Photonics, 1997, pp. 182–193.
- [28] 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*, vol. 17, no. 3, pp. 229–241, Jun 2001.
- [29] S. Huang and G. Dissanayake, "Convergence and consistency analysis for alman filter based slam," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1036–1049, Oct 2007.

- [30] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, “Fastslam: A factored solution to the simultaneous localization and mapping problem,” in *Aaai/iaai*, 2002, pp. 593–598.
- [31] M. Montemerlo and S. Thrun, “Fastslam 2.0,” *FastSLAM: A scalable method for the simultaneous localization and mapping problem in robotics*, pp. 63–90, 2007.
- [32] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, “Simultaneous localization and mapping with sparse extended information filters,” *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.
- [33] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. 2005. MIT Press, 2005.
- [34] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, “6d slam with approximate data association,” in *Advanced Robotics, 2005. ICAR’05. Proceedings., 12th International Conference on*. IEEE, 2005, pp. 242–249.
- [35] P. C. Ng and S. Henikoff, “Sift: Predicting amino acid changes that affect protein function,” *Nucleic acids research*, vol. 31, no. 13, pp. 3812–3814, 2003.
- [36] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, “A random-finite-set approach to bayesian slam,” *Robotics, IEEE Transactions on*, vol. 27, no. 2, pp. 268–282, 2011.
- [37] S. Thrun and M. Montemerlo, “The graph slam algorithm with applications to large-scale mapping of urban structures,” *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.
- [38] M. Kaess, A. Ranganathan, and F. Dellaert, “isam: Incremental smoothing and mapping,” *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1365–1378, 2008.

- [39] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard, “Large scale graph-based slam using aerial images as prior information,” *Autonomous Robots*, vol. 30, no. 1, pp. 25–39, 2011.
- [40] N. Snderhauf and P. Protzel, “Switchable constraints for robust pose graph slam,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 1879–1884.
- [41] Y. Latif, C. Cadena, and J. Neira, “Robust loop closing over time for pose graph slam,” *The International Journal of Robotics Research*, p. 0278364913498910, 2013.
- [42] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “g 2 o: A general framework for graph optimization,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3607–3613.
- [43] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [44] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige, “Double window optimisation for constant time visual slam,” in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2352–2359.
- [45] M. Fanfani, F. Bellavia, and C. Colombo, “Accurate keyframe selection and keypoint tracking for robust visual odometry,” *Machine Vision and Applications*, vol. 27, no. 6, pp. 833–844, Aug 2016. [Online]. Available: <https://doi.org/10.1007/s00138-016-0793-3>
- [46] H. Lim, J. Lim, and H. J. Kim, “Real-time 6-dof monocular visual slam in a large-scale environment,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 1532–1539.

- [47] H. Strasdat, J. Montiel, and A. J. Davison, “Visual slam: Why filter?” *Image and Vision Computing*, vol. 30, no. 2, pp. 65 – 77, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0262885612000248>
- [48] J. W. Fenwick, P. M. Newman, and J. J. Leonard, “Cooperative concurrent mapping and localization,” in *Robotics and Automation, 2002. Proceedings. ICRA ’02. IEEE International Conference on*, vol. 2. IEEE, 2002, pp. 1810–1817.
- [49] A. I. Mourikis and S. I. Roumeliotis, “Predicting the performance of cooperative simultaneous localization and mapping (c-slam),” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1273–1286, 2006.
- [50] A. Howard, “Multi-robot simultaneous localization and mapping using particle filters,” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, 2006.
- [51] L. Carlone, M. K. Ng, J. Du, B. Bona, and M. Indri, “Rao-blackwellized particle filters multi robot slam with unknown initial correspondences and limited communication,” in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 243–249.
- [52] L. A. Andersson and J. Nygard, “C-sam: Multi-robot slam using square root information smoothing,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 2798–2805.
- [53] D. Moratuwage, D. Wang, A. Rao, N. Senarathne, and H. Wang, “Rfs collaborative multivehicle slam: Slam in dynamic high-clutter environments,” *IEEE Robotics Automation Magazine*, vol. 21, no. 2, pp. 53–59, June 2014.
- [54] X. S. Zhou and S. I. Roumeliotis, “Robot-to-robot relative pose estimation from range measurements,” *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1379–1393, 2008.

- [55] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, “Multiple relative pose graphs for robust cooperative mapping,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3185–3192.
- [56] P. Dinnissen, S. N. Givigi Jr, and H. M. Schwartz, “Map merging of multi-robot slam using reinforcement learning,” in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 53–60.
- [57] K. Konolige, E. Marder-Eppstein, and B. Marthi, “Navigation in hybrid metric-topological maps,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 3041–3047.
- [58] F.-A. Moreno, J.-L. Blanco, and J. Gonzalez-Jimenez, “A constant-time SLAM back-end in the continuum between global mapping and submapping: application to visual stereo SLAM,” *The International Journal of Robotics Research*, p. 0278364915619238, 2015.
- [59] A. Angeli, D. Filliat, S. Doncieux, and J. A. Meyer, “Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1027–1037, Oct 2008.
- [60] H. Surmann, N. Berninger, and R. Worst, “3d mapping for multi hybrid robot cooperation,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 626–633.
- [61] A. Gawel, T. Cieslewski, R. Dub, M. Bosse, and R. Siegwart, “Structure-based vision-laser matching,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 182–188.
- [62] J. Jessup, S. N. Givigi, and A. Beaulieu, “Robust and efficient multirobot 3-d mapping merging with octree-based occupancy grids,” *IEEE Systems Journal*, vol. 11, no. 3, pp. 1723–1732, 2017.

- [63] S. Carpin, “Fast and accurate map merging for multi-robot systems,” *Autonomous Robots*, vol. 25, no. 3, pp. 305–316, 2008.
- [64] J. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, “Normal distributions transform monte-carlo localization (ndt-mcl),” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 382–389.
- [65] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees,” *Autonomous Robots*, 2013.
- [66] J. Oberlinder, A. Roennau, and R. Dillmann, “Hierarchical SLAM using spectral submap matching with opportunities for long-term operation,” in *Advanced Robotics (ICAR), 2013 16th International Conference on*, Nov 2013, pp. 1–7.
- [67] S. Schwertfeger and A. Birk, “Map evaluation using matched topology graphs,” *Autonomous Robots*, vol. 40, no. 5, pp. 761–787, Jun 2016.
- [68] M. Bosse and R. Zlot, “Map matching and data association for large-scale two-dimensional laser scan-based slam,” *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 667–691, 2008.
- [69] R. Aragues, C. Sagues, and Y. Mezouar, “Feature-based map merging with dynamic consensus on information increments,” *Autonomous Robots*, vol. 38, no. 3, pp. 243–259, Mar 2015.
- [70] J. Jessup, S. N. Givigi, and A. Beaulieu, “Robust and Efficient Multirobot 3-D Mapping Merging With Octree-Based Occupancy Grids,” *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–10, 2015.
- [71] M. J. Schuster, C. Brand, H. Hirschmüller, M. Suppa, and M. Beetz, “Multi-robot 6D graph SLAM connecting decoupled local reference filters,” in *Intel-*

- ligent Robots and Systems (IROS)*, 2015 IEEE/RSJ International Conference on. IEEE, 2015, pp. 5093–5100.
- [72] P. J. Besl, N. D. McKay *et al.*, “A method for registration of 3-d shapes,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [73] Y. Taguchi, Y. D. Jian, S. Ramalingam, and C. Feng, “Point-plane slam for hand-held 3d sensors,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 5182–5189.
- [74] A. Das, J. Servos, and S. L. Waslander, “3d scan registration using the normal distributions transform with ground segmentation and point cloud clustering,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 2207–2212.
- [75] A. Censi, “An accurate closed-form estimate of ICP’s covariance,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 3167–3172.
- [76] G. Agamennoni, S. Fontana, R. Y. Siegwart, and D. G. Sorrenti, “Point clouds registration with probabilistic data association,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 4092–4098.
- [77] A. Segal, D. Haehnel, and S. Thrun, “Generalized-icp.” in *Robotics: science and systems*, vol. 2, no. 4, 2009, p. 435.
- [78] J. Servos and S. L. Waslander, “Multi-channel generalized-icp: A robust framework for multi-channel scan registration,” *Robotics and Autonomous Systems*, vol. 87, no. Supplement C, pp. 247 – 257, 2017.
- [79] J. Liu, S. Du, C. Zhang, and J. Zhu, “Probability iterative closest point algorithm for position estimation,” in *International IEEE Conference on Intelligent Transportation Systems*, Oct 2014, pp. 458–463.

- [80] S. Nobili, R. Scona, and M. Caravagna, “Overlap-based icp tuning for robust localization of a humanoid robot,” in *IEEE International Conference on Robotics and Automation*, May 2017, pp. 4721–4728.
- [81] S. Ying, J. Peng, S. Du, and H. Qiao, “A scale stretch method based on icp for 3d data registration,” *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 3, pp. 559–565, July 2009.
- [82] Q. Li, R. Xiong, and T. Vidal-Calleja, “A gmm based uncertainty model for point clouds registration,” *Robotics and Autonomous Systems*, vol. 91, no. Supplement C, pp. 349 – 362, 2017.
- [83] D. Viejo and M. Cazorla, “A robust and fast method for 6dof motion estimation from generalized 3d data,” *Autonomous Robots*, vol. 36, no. 4, pp. 295–308, Apr 2014.
- [84] J. Serafin and G. Grisetti, “Using extended measurements and scene merging for efficient and robust point cloud registration,” *Robotics and Autonomous Systems*, vol. 92, no. Supplement C, pp. 91 – 106, 2017.
- [85] T. D. Stoyanov, M. Magnusson, H. Andreasson, and A. Lilienthal, “Fast and accurate scan registration through minimization of the distance between compact 3d ndt representations,” *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1377–1393, 2012.
- [86] M. Velas, M. Spanel, and A. Herout, “Collar line segments for fast odometry estimation from velodyne point clouds,” in *IEEE International Conference on Robotics and Automation*, May 2016, pp. 4486–4495.
- [87] Y. He, B. Liang, J. Yang, S. Li, and J. He, “An iterative closest points algorithm for registration of 3d laser scanner point clouds with geometric features,” *Sensors*, vol. 17, no. 8, 2017.

- [88] J. Jessup, S. N. Givigi, and A. Beaulieu, “Merging of octree based 3d occupancy grid maps,” in *Systems Conference (SysCon), 2014 8th Annual IEEE*. IEEE, 2014, pp. 371–377.
- [89] S. Carpin, “Fast and accurate map merging for multi-robot systems,” *Autonomous Robots*, vol. 25, no. 3, pp. 305–316, Oct 2008.
- [90] F. Tombari, S. Salti, and L. D. Stefano, “A combined texture-shape descriptor for enhanced 3d feature matching,” in *2011 18th IEEE International Conference on Image Processing*, Sept 2011, pp. 809–812.
- [91] G. C. Sharp, S. W. Lee, and D. K. Wehe, “Icp registration using invariant features,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 90–102, Jan 2002.
- [92] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk, “Beyond points: Evaluating recent 3d scan-matching algorithms,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 3631–3637.
- [93] A. Gawel, R. Dub, H. Surmann, J. Nieto, R. Siegwart, and C. Cadena, “3d registration of aerial and ground robots for disaster response: An evaluation of features, descriptors, and transformation estimation,” in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, Oct 2017, pp. 27–34.
- [94] B. Lee and D. D. Lee, “Learning anisotropic icp (la-icp) for robust and efficient 3d registration,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 5040–5045.
- [95] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, “Probabilistic data association for semantic slam,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2017, pp. 1092–1098 vol.2.

- [96] Q. Zhang and R. Pless, “Extrinsic calibration of a camera and laser range finder (improves camera calibration),” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sept 2004, pp. 2301–2306 vol.3.
- [97] A. Geiger, F. Moosmann, . Car, and B. Schuster, “Automatic camera and range sensor calibration using a single shot,” in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 3936–3943.
- [98] ——. (2012) Camera and range sensor calibration toolbox. [Online]. Available: <http://www.cvlibs.net/software/calibration/index.php>
- [99] M. Velas, M. Spanel, Z. Materna, and A. Herout, “Calibration of rgb camera with velodyne lidar,” in *WSCG 2014*, 2014.
- [100] C. Guindel, J. Beltrn, D. Martn, and F. Garca, “Automatic extrinsic calibration for lidar-stereo vehicle sensor setups,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct 2017, pp. 1–6.
- [101] S. Vidas, P. Moghadam, and M. Bosse, “3d thermal mapping of building interiors using an rgb-d and thermal camera,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 2311–2318.
- [102] S. Vidas, R. Lakemond, S. Denman, C. Fookes, S. Sridharan, and T. Wark, “A mask-based approach for the geometric calibration of thermal-infrared cameras,” *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 6, pp. 1625–1635, 2012.
- [103] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, “A tutorial on graph-based SLAM,” *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.
- [104] P. Agarwal, G. D. Tipaldi, and L. Spinello, “Robust map optimization using dynamic covariance scaling,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 62–69.

- [105] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, “3-d mapping with an rgb-d camera,” *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, Feb 2014.
- [106] Y. Yue, D. Wang, P. G. C. N. Sena, and C. Yang, “Robust submap-based probabilistic inconsistency detection for multi-robot mapping,” in *2017 European Conference on Mobile Robots (ECMR)*, Sept 2017, pp. 1–6.
- [107] D. Hähnel, S. Thrun, B. Wegbreit, and W. Burgard, “Towards lazy data association in SLAM,” in *Robotics Research. The Eleventh International Symposium*. Springer, 2005, pp. 421–431.
- [108] M. Mazuran and G. Diego Tipaldi and L. Spinello and W. Burgard and C. Stachniss, “A statistical measure for map consistency in slam,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 3650–3655.
- [109] I. Bogoslavskyi, M. Mazuran, and C. Stachniss, “Robust homing for autonomous robots,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2550–2556.
- [110] E. Rehder and A. Albrecht, “Submap-based SLAM for road markings,” in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 1393–1398.
- [111] J. L. Blanco, J. A. Fernandez-Madrigal, and J. Gonzez, “Toward a Unified Bayesian Approach to Hybrid Metric–Topological SLAM,” *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 259–270, April 2008.
- [112] K. Ni and F. Dellaert, “Multi-level submap based slam using nested dissection,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 2558–2565.
- [113] M. Guo and M. M. Zavlanos, “Multi-robot data gathering under buffer constraints and intermittent communication,” *arXiv preprint arXiv:1706.02092*, 2017.

- [114] J. Levinson and S. Thrun, “Automatic online calibration of cameras and lasers.” in *Robotics: Science and Systems*, 2013, pp. 24–28.
- [115] W. Maddern and P. Newman, “Real-time probabilistic fusion of sparse 3d lidar and dense stereo,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 2181–2188.
- [116] H. Carrillo, P. Dames, V. Kumar, and J. A. Castellanos, “Autonomous robotic exploration using occupancy grid maps and graph slam based on shannon and renyi entropy,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 487–494.
- [117] P. Payeur, P. Hebert, D. Laurendeau, and C. M. Gosselin, “Probabilistic octree modeling of a 3d dynamic environment,” in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 2, Apr 1997, pp. 1289–1296 vol.2.
- [118] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *Intelligent Robots and Systems, 2004 IEEE/RSJ International Conference on*, vol. 3, Sep 2004, pp. 2149–2154 vol.3.
- [119] M. Labb and F. Michaud, “Online global loop closure detection for large-scale multi-session graph-based slam,” in *Intelligent Robots and Systems , 2014 IEEE/RSJ International Conference on*, Sep 2014, pp. 2661–2666.
- [120] G. Grisetti, C. Stachniss, and W. Burgard, “Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb 2007.
- [121] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 1–4.
- [122] K. Klasing, D. Althoff, D. Wollherr, and M. Buss, “Comparison of surface normal estimation methods for range sensing applications,” in *2009 IEEE*

- International Conference on Robotics and Automation*, May 2009, pp. 3206–3211.
- [123] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” *Image and vision computing*, vol. 10, 1992.
- [124] M. Magnusson, “The three-dimensional normal-distributions transform — an efficient representation for registration, surface analysis, and loop detection,” 12 2009.
- [125] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
- [126] Z. Yan, N. Jouandeau, and A. A. Cherif, “A survey and analysis of multi-robot coordination,” *International Journal of Advanced Robotic Systems*, vol. 10, 2013.