

Graduate Theses, Dissertations, and Problem Reports

2021

Localization Algorithms for GNSS-denied and Challenging Environments

Chizhao Yang West Virginia University, cy0003@mix.wvu.edu

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Part of the Multi-Vehicle Systems and Air Traffic Control Commons, Navigation, Guidance, Control and Dynamics Commons, and the Robotics Commons

Recommended Citation

Yang, Chizhao, "Localization Algorithms for GNSS-denied and Challenging Environments" (2021). *Graduate Theses, Dissertations, and Problem Reports.* 10278. https://researchrepository.wvu.edu/etd/10278

This Dissertation is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Dissertation has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Localization Algorithms for GNSS-denied and Challenging Environments

Chizhao Yang

Dissertation submitted to the Benjamin M. Statler College of Engineering and Mineral Resources at West Virginia University

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy in Aerospace Engineering

Yu Gu, Ph.D., Chair Jason Gross, Ph.D. Guilherme Augusto Silva Pereira, Ph.D. Powsiri Klinkhachorn, Ph.D. Xin Li, Ph.D. Mario Perhinschi. Ph.D

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

Morgantown, West Virginia 2021

Keywords: Localization, Multi-Agent System, Challenging Environments, Cooperative Navigation, SLAM Copyright © 2021 - Chizhao Yang CC BY-NC 4.0

Abstract

Localization Algorithms for GNSS-denied and Challenging Environments

Chizhao Yang

In this dissertation, the problem about localization in Global Navigation Satellite System (GNSS)denied and challenging environments is addressed. Specifically, the challenging environments discussed in this dissertation include two different types, environments including only low-resolution features and environments containing moving objects. To achieve accurate pose estimates, the errors are always bounded through matching observations from sensors with surrounding environments. These challenging environments, unfortunately, would bring troubles into matching related methods, such as "fingerprint" matching, and Iterative Closest Point (ICP). For instance, in environments with low-resolution features, the on-board sensor measurements could match to multiple positions on a map, which creates ambiguity; in environments with moving objects included, the accuracy of the estimated localization is affected by the moving objects when performing matching. In this dissertation, two sensor fusion based strategies are proposed to solve localization problems with respect to these two types of challenging environments, respectively.

For environments with only low-resolution features, such as flying over sea or desert, a multiagent localization algorithm using pairwise communication with ranging and magnetic anomaly measurements is proposed in this dissertation. A scalable framework is then presented to extend the multi-agent localization algorithm to be suitable for a large group of agents (e.g., 128 agents) through applying Covariance Intersection (CI) algorithm. The simulation results show that the proposed algorithm is able to deal with large group sizes, achieve 10 meters level localization performance with 180 km traveling distance, while under restrictive communication constraints.

For environments including moving objects, lidar-inertial-based solutions are proposed and tested in this dissertation. Inspired by the CI algorithm presented above, a potential solution using multiple features motions estimate and tracking is analyzed. In order to improve the performance and effectiveness of the potential solution, a lidar-inertial based Simultaneous Localization and Mapping (SLAM) algorithm is then proposed. In this method, an efficient tightly-coupled iterated Kalman filter with a build-in dynamic object filter is designed as the front-end of the SLAM algorithm, and the factor graph strategy using a scan context technology as the loop closure detection is utilized as the back-end. The performance of the proposed lidar-inertial based SLAM algorithm is evaluated with several data sets collected in environments including moving objects, and compared with the state-of-the-art lidar-inertial based SLAM algorithms.

Acknowledgments

This work would not have been possible without the support from friends, family, and colleagues. To show my appreciation, I would like to take a brief moment to acknowledge a few of them.

I would first like to thank my advisor **Dr. Yu Gu** for leading me to analyze this problem and instructing me to explore ideas. I really appreciate your supports and mentorships in my academic researches. Your suggestions always lead my way in both my study and my life. Next, I would like to thank **Dr. Jason Gross** for your helpful suggestions and ideas in my researches, especially in the cooperative navigation and the localization system design in the pollination robot project. Your useful suggestions give me a lot of inspirations. Additionally, I would like to thank the remaining members of my committee, **Dr. Guilherme Pereira**, **Dr. Powsiri Klinkhachorn**, **Dr. Xin Li**, and **Dr. Mario Perhinschi**. Your applied and creative comments will make the proposed idea come true.

Moreover, I would like to recognize every member of the **WVU Interactive Robotics Laboratory(IRL)**. Everyone is so selfless and helpful at any time. I would like especially to thank **Dr. Jared Strader** for your assistants on most of my research projects, **Nick Ohi** for your supports in experiments whenever I need your help, **Jenn Nguyen** for your suggestions in my coding skills, **Dr. Kyle Lassak** for your inspirations in my algorithm development, **Scott Harper**, **Corner Castle**, **Mo Buzzo**, **Jonas Bredu**, **Dr. Cagri Kilic**, **Chris Tatsch**, **Dylan Covell**, **Maria Gonzalez** for your supports in hardware design and data collection.

Also, I would like to thank the supports from the US Air Force Research Laboratory (Award No. FA8651-16-1-0002), the USDA NIFA project (No. 2017-67022-25926), the NASA EPSCoR Research Cooperative Agreement (WV-80NSSC17M0053), the NASA Innovative Advanced Concepts (NIAC 80NSSC19K0963), and the West Virginia University Outstanding Merit Fellowship for Continuing Doctoral Students.

Finally, I would like to thank my dear parents and my dear wife **Minghao Liu**, without your support, I cannot finish anything.

Contents

1	INT	CODUCTION	1											
	1.1	Motivation and Objectives	2											
	1.2	Contributions	4											
	1.3	Organization	6											
2	Background													
	2.1	Cooperative Localization Based On Scalar Field Measurements	12											
		2.1.1 Background	12											
		2.1.2 Scalar Field-based Localization with Multiple Agents	14											
	2.2	Lidar-Based Localization	16											
		2.2.1 Background	16											
		2.2.2 Lidar-based Localization	20											
3	Сос	perative UAV Localization Using Inter-vehicle Ranging and Magnetic												
	ANO	MALY MEASUREMENTS	23											
	3.1	Introduction	24											
	3.2	Problem Statement												
	3.3	Technical Approach												
		3.3.1 Communication	27											
		3.3.2 Cooperative Ranging Localization	31											
		3.3.3 Cooperative Magnetic Localization	35											
	3.4	Simulation	39											
	3.5	Results	41											
		3.5.1 Case Studies	42											
		3.5.2 Sensitivity Analysis	45											
	3.6	Discussion	50											
4	Coc	perative Navigation Using Pairwise Communication with Ranging and												
1	MAG	NETIC ANOMALY MEASUREMENTS	52											

4.1	Introduction
4.2	Problem Statement
4.3	Technical Approach
	4.3.1 Communication
4.4	Simulator
4.5	Results
	4.5.1 Case Studies
	4.5.2 Sensitivity Analysis
4.6	Discussion
A so	CALABLE FRAMEWORK FOR MAP MATCHING BASED COOPERATIVE LOCALIZA-
TION	67
5.1	Introduction
5.2	Problem Statement and Notations
5.3	System Design 73
	5.3.1 System Overview
	5.3.2 Subgroup Cooperative Localization
	5.3.3 Data Fusion
5.4	Simulation
	5.4.1 Magnetic Anomaly Map
	5.4.2 Bathymetric Map
5.5	Results
5.6	Discussion
Lida	AR-INERTIAL BASED LOCALIZATION 87
6.1	Introduction
6.2	Problem Statement
6.3	Methods
	6.3.1 System Overview
	6.3.2 Lidar-inertial odometry 94
	6.3.3 Loop Closure Detection and Pose-Graph Optimization
6.4	Experiments
	6.4.1 Data Sets Collected Using A Robot
	6.4.2 Data sets collected using a handheld lidar
6.5	Results
6.6	Discussion
Con	ICLUSIONS AND FUTURE WORK 118
7.1	Conclusions
	 4.1 4.2 4.3 4.4 4.5 4.6 A so TION 5.1 5.2 5.3 5.4 5.5 5.6 LIDA 6.1 6.2 6.3 6.4 6.5 6.6 CON 7.1

7.2	Future Work	•••	•••	•••	• •	•	•	•••	•	• •	•	• •	•••	•	•••	•	•	•	•••	•	•	•	• •	•	•	•	•	121
Referen	CES																											136

List of Acronyms

AERB Advance Engineering Research Building
CDF Cumulative Distribution Function
CI Covariance Intersection
CU Covariance Union
DBSCAN Density-Based Spatial Clustering of Applications with Noise
DR dead-reckoning
EKF extended Kalman Filter
Fast-LIO Fast Lidar-Inertial Odometry 88
FC full communication
GNSS Global Navigation Satellite System
GPS Global Positioning System
GTSAM Georgia Tech Smoothing and Mapping
ICP Iterative Closest Point i
IEKF iterated extended Kalman Filter
IMU inertial measurement unit
INS Inertial Navigation System
Lego-LOAM Lightweight and ground optimized lidar odometry and mapping 88
LOAM Lidar Odometry and Mapping 88
LIO-SAM Lidar Inertial Odometry via Smoothing and Mapping
NDT Normal Distributions Transform
PDF Probability Density Function 17

RANSAC random sample consensus	10
RMSE Root Mean Square Error	42
SLAM Simultaneous Localization and Mapping	ii
SVD Singular Value Decomposition	16
UAV unmanned aerial vehicle	2
UKF unscented Kalman Filter	3
WVU West Virginia University	103

Introduction

1.1 Motivation and Objectives

The ability of localization (e.g., to estimate one's position and orientation with respect to surrounding environments) is necessary for autonomous navigation. Without accurate pose and velocity estimates, some key functions of autonomous navigation, such as collision avoidance, path planning, would become difficult. The GNSS assisted by the Inertial Navigation System (INS) has been adopted as a solution for solving outdoor localization problems for several decades [1, 2]. However, due to reasons such as signal blockages, multipath reflection, and jamming, GNSS is not always available or reliable [3]. In GNSS-denied environments, in order to achieve accurate pose estimates using other information sources (e.g., vision, magnetic information, altitude), the localization errors may be bounded through matching observations from sensors with surrounding environments. However, localization algorithms dependent on this information may fail in some challenging environments, such as environments including only low-resolution features, or environments containing moving objects. These challenging environments would bring troubles into matching related methods, such as "fingerprint" matching [4], and ICP [5]. For instance, in environments with low-resolution features, the onboard sensor measurements could match to multiple positions on a map, which creates ambiguity; in environments with moving objects included, the accuracy of the estimated localization solution is affected by the moving objects when performing matching. Meanwhile, some applications related to autonomous navigation have to be applied in these challenging environments, such as unmanned aerial vehicles (UAVs) flying over deserts or seas, robots navigating in a room with crowds of people. Therefore, to develop accurate and robust localization algorithms for GNSS-denied and challenging environments is a meaningful research topic and is the focus of this dissertation.

In order to reduce the influences of noisy information from the challenging environments, mea-

surements or observations from multiple sensors (e.g., internal sensors, such as wheel encoders, inertial measurement units (IMUs), or external sensors, such as cameras, lidars, magnetometers, altimeters) are often fused to achieve robust pose estimation. The sensor fusion algorithms, such as Kalman Filter[6] and its variances (e.g., extended Kalman Filter (EKF)[7], unscented Kalman Filter (UKF)[8]), CI[9], particle filter[10], factor-graph based optimizations[11], are often chosen to solve these problems depending on applications. Moreover, due to different types of challenges existing in different environments, localization solutions are not the same based on applications and sensors applied. In this dissertation, two sensor fusion-based localization algorithms are designed with respect to two specific applications in different types of challenging environments (i.e., environments including only low-resolution features and environments containing moving objects). Specifically, for environments with only low-resolution features, such as flying overseas or deserts, a multi-agent localization algorithm using inter-agent communication with ranging and scalar field measurements is proposed in this dissertation; for environments including a large number of moving objects, a lidar-inertial-based solution is proposed in this dissertation. To sum up, in this dissertation, the goal is to develop reliable localization systems for challenging environments, which are environments including only low-resolution features and environments containing moving objects). Depending on different types of challenging environments, the details of applications and solutions are different. Therefore, with respect to those two different types of challenging environments, the objectives of this dissertation are:

- To develop a cooperative localization algorithm mainly using inter-agent ranging and scalar field measurements with communication limitations;
- To design a reliable lidar-based localization system for environments including a large number of dynamic objects.

1.2 Contributions

This dissertation addresses the reliable localization problem with respect to two applications in two different types of challenging environments (i.e., environments including only low-resolution features and environments containing moving objects). Specifically, two sensor fusion based localization algorithms are presented.

For environments including only low-resolution features, a cooperative localization system using inter-agent ranging and scalar field measurements with limited communication is designed. The contributions of the presented cooperative localization system are summarized as follows.

- The presented cooperative localization system is able to estimate the global pose with lower computation requirements due to the significantly less number of states used in the particle filter compared with the previous approach presented in [12]. This would in turn allow the application of this cooperative navigation algorithm on a larger group.
- The geometric structure of the group (i.e., the relative position inside the group) are able to estimate using pairwise ranging information, which means that each agent at one-time step is able to communicate and perform ranging measurements with only one other agent in the group. The simulation study also suggests that the pairwise communication approach is capable of delivering a majority of the cooperative navigation benefit as compared to the complete communication scenario (i.e., at each time step, each agent can communicate and perform ranging measurements with all other agents in the group), which is more feasible with existing ranging devices.
- Through evaluating with different resolution map, the proposed method show more robust compared with a single agent case. Meanwhile, the results from the sensitivity analysis show

that the presented algorithm can tolerate large variations of velocity, yaw rate, and scalar field measurement noises.

• A scalable framework is presented to perform cooperative localization based on scalar field information, which is performed through fusing the solutions estimated by smaller local subgroups in a large group. The proposed framework can be scaled to large group sizes under communication constraints (e.g., a group of 128 agents was simulated) with a limitation that the cooperative localization performance is a function of the subgroup size instead of the full group size.

For environments including a large number of moving objects, a lidar-inertial based SLAM algorithm with a dynamic object removal filter is presented to estimate 3D pose for a long time drive. In the presented algorithm, an iterated extended Kalman Filter (IEKF) based lidar-inertial odometry solution using inertial measurements (i.e., 3D acceleration and angular rate) as prediction inputs and point cloud based matching results as observation updates is proposed. In order to reduce the influences about the noisy points from moving objects, a dynamic object removal filter through checking distance changes is designed to filter out the points from moving objects before expending the local map with the current scan. A factor graph-based framework is performed as the back-end of the proposed SLAM algorithm, and the scan context algorithm [13] is applied for loop closure detection. The contributions of the proposed lidar-inertial based SLAM algorithm are summarized as follows.

• Compared with existed point cloud matching based SLAM algorithms in a dynamic environment, the proposed algorithm is able to effectively detect and remove most of the points from moving objects. • A data set for evaluating the robustness of motion estimation algorithms in a dynamic environment is set up. The performance of the proposed algorithm is compared with the stateof-the-art lidar-inertial SLAM algorithms in the collected data set. The results show that the proposed algorithm is able to achieve more robust online localization compared with other algorithms.

1.3 Organization

The rest of this dissertation is organized in the following manner. The background and related works about localization system design for GNSS-denied and challenging environments, especially environments with only low-resolution features and environments including moving objects, are summarised in Chapter 2. With respect to two different applications, the background and related works about cooperative localization system design and lidar-inertial based localization algorithm design are discussed in detail in Chapter 2, respectively. In Chapter 3, a cooperative localization system is presented, and is evaluated with different communication strategies and simulated magnetic anomaly map. The cooperative localization system is then evaluated with magnetic anomaly map generated from published data set and a designed feedback control-based simulator in Chapter 4. A scalable framework is presented in Chapter 5 to extend the cooperative localization algorithm to suit for groups with large size. A lidar-inertial-based localization algorithm with build-in dynamic object removal filter is presented in Chapter 6. The dissertation is concluded in Chapter 7.

Background

In this dissertation, the problem about localization in GNSS-denied and challenging environments is investigated. As discussed in Chapter 1, the challenging environments analyzed in this dissertation contain two different types, which are environments with only low-resolution features and environments including a large number of moving objects. Due to different types of challenges for localization in different environments, the developed localization solutions are not the same based on applications and the sensors used.

For environments with only low-resolution features, the problem is defined as localization mainly relying on scalar field measurements in this dissertation. Scalar fields associate a scalar value with every point in space, and applications include gravity anomaly [14], magnetic anomaly [15, 4], topographic [16], and olfaction [17], to name a few. For some types of the scalar field, the data around the world is published and available, such as magnetic anomaly [4], which is a benefit for navigation. However, part of these available data is in low-resolution, such as magnetic anomaly map with resolution in about 900-meter [18], bathymetric map with resolution in 100-meter [19]. Methods utilizing scalar fields for localization regulate agents' dead-reckoning error growth through matching the information measured by onboard sensors with the prior given scalar field maps, such as terrain aid navigation 20 and magnetic anomaly-based navigation 4. However, these methods are sensitive to the characteristic information available in the local area near the agent, the sensor noises, and the resolution and accuracy of the given maps. For example, for a single agent localization, the onboard sensor measurements could match multiple positions on a scalar field map, creating ambiguity. This can be alleviated through matching the past sensor measurements along the agent's trajectory to the map [21], but is still often not robust in a real-world application. Fortunately, compared to a single agent, a group of collaborative agents may provide several navigational benefits, such as tolerance against individual sensor failures. This can be achieved through

sharing observations across a large spatial area on the scalar field.

Within cooperative localization problems, communication limitation and relative poses estimated inside the group are inevitable issues needed to be solved. Due to communication limitations (e.g., range or number of channels), each agent in the group can only communicate with a limited number of agents inside a certain range. Therefore, a reliable and scalable framework to satisfy the communication limitations is required to be designed. Moreover, to improve the reliability of relative observations, many approaches rely on both inter-agent ranging and bearing measurements [22, 23]. Bearing-only measurements have been tested and proven capable of estimating the pose of two nearby agents (about 10 meters apart) [24]. However, bearing measurements often have a limited range, such as measurements from cameras. Rapid and precise distance measurements at a long-range are available using a coherent laser ranging system presented in [25] or radio-based systems $\begin{bmatrix} 26 \end{bmatrix}$. With the low-resolution scalar field map, in order to cover a large spatial area to improve the localization performance, the distances among agents inside the group are usually over the limitation of the bearing sensors. In this case, estimating relative poses inside the group using ranging information is more applicable. Therefore, developing a cooperative localization algorithm mainly using inter-agent ranging and scalar field measurements with communication limitations is one part of the problem in this dissertation.

For environments including a large number of moving objects, the accuracy of the estimated localization is affected by the moving objects when performing visible features based matching algorithms, such as direct methods applied with image information [27], Normal Distributions Transform (NDT) applied with point cloud information [28], and ICP [5] applied with both image and point cloud information. Basically, these matching algorithms are trying to find a transformation between two input observations (e.g., images, point clouds) to minimize the differences (e.g., distance, intensity, distribution) between each pair of correspondences. The estimated transformation between two observations could be considered as the motion of the sensor when the surrounding environment is mostly static. In practice, however, the localization system has often been applied in the environment including moving objects [29, 30], such as self-driving vehicles passing through the street with crowded people, autonomous driving robotic guides working in museums at a busy time. For image-based observations, several algorithms were proposed to solve visual odometry or visual SLAM problems in dynamic environments [31, 32, 33, 34, 35, 36, 37, 38, 39]. These methods remove dynamic objects through detecting the difference of the intensity [31, 33, 34] or using semantic information [32]. However, these methods cannot be applied directly with point clouds information. Compared with the images, the point clouds from lidars only present sparse information (i.e., the information from one point cloud is not dense as from one image). Moreover, with point cloud data, object detection based on semantic is more difficult than using images due to less information from one point cloud and less available labeled data set used for training. In practice, however, the lidars cannot always be replaced with cameras since the cameras cannot work in dark environments and are not able to measure accurate depth information with long distances, which are necessary and useful for localization.

In order to reduce the noise from the dynamic objects, a straightforward solution is to preprocess point clouds before performing registration. In [40, 41], features, such as edges and planar points, extracted from raw point cloud are used to do an alignment. Even though these methods are able to reduce the ratio of points in dynamic objects in the processed point cloud, the rest of the noisy points (i.e., points from dynamic objects) in the processed point cloud still affect the performance of registration. To detect points from dynamic environment precisely, a random sample consensus (RANSAC) based outlier detection method is applied in [42] to perform pose estimation in urban dynamic environments. However, the method presented in [42] only estimates 2D poses and works in large outdoor environments. Dynamic points removing algorithms presented in [43, 44, 45, 46] are related on multiple scans (one scan is defined as the point cloud collected from one full sweep for mechanical lidars or one-shot for solid-state lidars) in sequential time slots and accurate transformations between each pair of continuing scans. In this case, these algorithms cannot be applied online to estimate poses in dynamic environments. In [47], semantic information learned using the RangeNet++ [48] method is applied to remove dynamic points when generating point cloud map. Due to the lack of the labeled data set, the types of the dynamic objects are limited in [47]. Therefore, to design a robust lidar-based localization system for environments including a large number of dynamic objects is another part of the problem in this dissertation.

To sum up, for environments with only low-resolution features, the problem is defined as cooperative localization mainly relying on scalar field measurements (e.g., magnetic anomaly, altitude); for environments including a large number of moving objects, lidar-based localization problem is focused. Although sensor fusion methods are applied in solutions for both environments, in order to state more clear, in this chapter, the background and related works related to the cooperative localization mainly based on scalar field measurements and lidar-based localization are discussed, respectively.

2.1 Cooperative Localization Based On Scalar Field Measurements

2.1.1 Background

Cooperative multi-agent systems have become increasingly popular due to the wide range of applications that they support, such as surveillance [49], search and rescue [50], and exploration [51]. In these applications, high-quality localization, the ability for agents to reliably and accurately estimate their poses (i.e., positions and orientations) with respect to the surrounding environment or to a geographic coordinate system, is crucial. One potential solution for localization in GNSS-denied environments is to utilize map matching techniques, given a prior map represented as a scalar field. Scalar fields associate a scalar value with every point in space, and applications include gravity anomaly [14, 52], magnetic anomaly [15, 4, 53], topographic [16, 54], and olfaction [17, 55], to name a few. Some of the scalar field information has been applied in localization because of their characterises, such as magnetic anomaly information [56, 4]. The magnetic anomalies present the high spatial frequency content of the Earth's magnetic field. Additionally, the magnetic anomaly information has been measured for most regions in the world [4]. Even in an indoor environment, the magnetic field can be mapped for supporting vehicles' navigation systems [57, 58]. Another reason for using magnetic anomaly information in localization is due to the technological maturity of magnetometers. For instance, optically pumped cesium magnetometers, which can achieve an accuracy of 0.1 nanoTesla(nT), have been used to create magnetic anomaly maps in geological surveys [59, 60].

Methods utilizing scalar fields for localization regulate agents' dead-reckoning error growth through matching the information measured by on-board sensors with the prior given scalar field maps,

such as terrain-aid navigation [20] and magnetic anomaly-based navigation [4]. However, these methods are sensitive to the characteristic information available in the local area near the agent, sensor noises, and the resolution and accuracy of the given maps. For example, for a single agent localization, the on-board sensor measurements could match to multiple positions on a scalar field map, creating ambiguity. This can be alleviated through matching the past sensor measurements along the agent's trajectory to the map [21] but is still often not robust in real-world applications. Fortunately, compared to a single agent, a group of collaborative agents may provide several navigational benefits, such as tolerance against individual sensor failures. This can be achieved through sharing observations across a large spatial area on the scalar field. Therefore, cooperative localization using scalar field is an active research topic that has been studied over the last decade [61, 62, 12, 63, 64, 65].

The scalar field-based cooperative localization algorithms can be classified into two main approaches. The first one is to treat the multi-agent group as a unity and to match observations from all agents with the given map to estimate their poses at each time step, which can be considered centralized methods. The centralized methods are able to achieve error growth bounded positioning and show robustness to issues such as low resolution of the map [15]. However, due to constraints on communication and the on-board computing resources for the agents, the group size of the centralized cooperative localization is limited in practice. The second approach is to perform decentralized cooperative localization, which means that each agent in the group estimates its own pose based on scalar field observations independently at each time step. Then, the estimates are updated, using the relative information (such as ranging, bearing) between this agent and its neighbors [61]. Usually, the communication constraints (e.g., range, connectivity, bandwidth) are considered when designing the decentralized localization approaches. In theory, decentralized

methods are scalable to the group size and robust to errors made by, or failures of, individual agents. However, in the existing approaches [66, 61], each individual agent needs to come up with a pose estimation, using its own scalar field measurement first, which has potential robustness issues in information poor regions.

2.1.2 Scalar Field-based Localization with Multiple Agents

Scalar field based localization system designs have been researched in applications such as gravityaid navigation [21, 67, 68], magnetic anomaly–based navigation [4, 69], and terrain-based navigation [70, 71]. In order to perform robust localization in featureless areas or with low-quality sensors, cooperative multi-agent localization systems are proposed to achieve accurate estimations [72, 73, 74]. Distributed multi-agent localization methods were first formulated based on Kalman filters [75, 76]. Even though these methods allow to perform an observation update and data exchange when agents are within the communication range, each agent in the group is required to estimate the poses of all agents, which does not scale well to large groups. Meanwhile, the Kalman filter-based estimation methods assume that the pose estimate can be presented by a unimodal Gaussian distribution. However, the scalar field-based estimation error distributions are usually multi-modal and difficult to be approximated by the Gaussian distribution [4, 61].

Canciani et al. formulated the magnetic anomaly-based cooperative navigation problem as a particle filter [12]. The method does not scale for large groups, due to the use of a centralized particle filter. The works in [15, 63] broke the localization process into two steps: the relative poses between agents are estimated using inter-agent ranging measurements through an EKF, and then each agent estimates its pose using all magnetic anomaly measurements and relative poses of the group through a particle filter. Although the particle filter in [15, 63] only contains four states,

the EKF formulation, which includes all agents' poses, does not scale to large group sizes.

A decentralized cooperative bathymetry based localization method was proposed in [66]. In [66], each agent is able to estimate its pose through matching altimeter measurements with a bathymetric map, using a marginalized particle filter. Then, the Gaussian belief, estimated based on the inter-agent ranging measurement and the other agent's position estimate, is applied to update the particles in the filter. Although this method is able to achieve scalable cooperative localization, it ignores the correlation of the information, which may lead to over convergence. Rui et al. then presented an extended information filter to address the issues about the correlation of the information [77]. However, the method described in [77] is reliable to GPS measurement or highly accurate bathymetric information-based estimations for the prediction update, which leads this method to be non-feasible in an underwater environment.

Wiktor et al. presented a decentralized CI based collaborative multi-agent localization algorithm applied in natural terrain-aid navigation [61]. Similar to [66], each agent is assumed to perform terrain-aid navigation to estimate its own pose and related covariances. The pose estimates are updated using inter-agent ranging measurements and agents' pose and covariances through a CI filter, which can fuse estimates with unknown correlation. One potential limitation of this method is the robustness within feature-poor regions due to the single measurement used in map matching for each agent at each time step. Compared with those methods presented in [66, 77, 61], which only use the information from the immediate neighbors, in the proposed algorithm, a subgroup strategy, instead of utilizing single measurements, is applied to improve the robustness of the localization system.

Active multi-agent navigation algorithms [78, 79, 80], which combine localization and active path planning algorithm, are interesting research directions to improve the robustness of the pose

estimation. However, to the best knowledge of the author, there is so far no active multi-agent navigation algorithms focused on localization based on map matching using scalar field information.

2.2 Lidar-Based Localization

2.2.1 Background

Lidar-based localization in environments including a large number of moving objects is a challenge problem [81]. In order to present a robust solution for this problem, knowledge from several research topics need to be fused, such as point cloud registration, outlier detection, and sensor fusion (data fusion). The background of these related knowledge are discussed in this section.

Registration Methods

Point cloud registration algorithms are designed to estimate the transformation (i.e., translation and rotation) which aligns two point clouds. The point cloud registration can support a wide range of applications, such as motion estimation and 3D reconstruction [40, 82], medical imaging [83], and object recognition [84], to name a few. When the correct correspondences between point clouds are known, the registration becomes a linear least-squares problem that aims to minimize the sum of the Euclidean distances between correspondence points. The linear least-squares problem can be solved robustly using the Singular Value Decomposition (SVD) method [85]. However, in practice, especially in motion estimation, the correspondences are usually unknown and difficult to estimate. Besl and McKay proposed a solution, named ICP, for the problem about point cloud registration with unknown correspondences [5]. The ICP iteratively processes finding correspondences, estimating transformations, and evaluating distance errors to disregard outliers and get the best transformation estimate. In details, two point clouds are assigned to source and reference separately, and the transformation from source to reference is the goal needed to be estimated. In every process circle, the source is transformed using the previous estimate of the rotation and translation parameters. Correspondence points between reference and transformed source are then selected based on nearest neighbor approaches [86]. Based on the correspondences, the temporary transformation is estimated through linear least-squares methods, such as SVD. The temporary transformation is chosen as the final result if the sum of the distance between correspondences is less than a threshold.

The original ICP algorithm is also called point-to-point ICP, which often falls into local minimum due to finding wrong correspondences. In order to improve the accuracy and robustness of the ICP, several variants were proposed. Chen and Medioni presented a point-to-plane variant of ICP which is taking advantage of surface normal information when calculating the distance error [87]. Generalized ICP, presented in [88], is extended from [87], which considered surface normal information from both source and reference. Therefore, the generalized ICP is also called the plane-to-plane variant of ICP. However, registration using points directly has a number of limitations. For example, noisy points from sensors have an influence on the accuracy of estimation. Although the surface normal information is utilized, point clouds contain no explicit information about surface characteristics such as orientation, smoothness, or holes. The NDT algorithm, first proposed by Biber and Straber, solves point cloud registration by converting the reference point cloud to normal-distribution map [89]. The NDT algorithm is first to subdivide the space occupied by the point cloud into a grid of cells. A Probability Density Function (PDF) is computed for each cell based on the point distribution within the cell. Each point in the source point cloud is assigned to a correspondent cell in a normal-distribution map related to a score defined by a likelihood function, and the gradient vector and the Hessian matrix are updated based on the score. As a result, the best transformation is computed using the gradient vector and the Hessian matrix. Magnusson improved the accuracy of the NDT by applying the Gaussian approximation of the log-likelihood of the mixture model to compute the score [90].

Compared with ICP, NDT is more robust since it applies normal-distribution of cells which is able to reduce the effectiveness of the noisy points in finding correspondences. However, both ICP and NDT are difficult to have a precise transformation estimate when a large number of dynamic objects is included in point clouds. Thus, in motion estimation applications, the existed point cloud registration algorithms cannot be applied in a environment including a large number moving objects directly. Meanwhile, the motion estimation algorithms designed based on these point cloud registration methods are not suitable in this type of environment.

Outlier Detection

Outlier detection is an important research topic in data analysis. Depended on different application domains, many outlier detection algorithms have been specifically developed. In order to figure out the moving object, outlier detection is a useful technology. The clustering algorithms have been developed for many decades and can be divided into different categories [91]. Clustering algorithms based on partitions, such as K-means [92] and K-medoids [93], are to determine the center of data points of the corresponding cluster. Although these algorithms perform with high computing efficiency, the results are relatively sensitive to the outliers and the number of clusters needs to be pre-defined. Hierarchy based clustering algorithms, such as CURE [94] and ROCK [95], are to cluster data through constructing the hierarchical relationship among data. Compared with partition-based algorithms, these algorithms are not sensitive to the outliers. The number

of clusters, however, still needs to be preset. The clustering algorithm based on density, such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [96], is to cluster the data which is in the region with a high density of the data space. The DBSCAN does not force every data instance to belong to a cluster, which means it is not sensitive to outliers. Moreover, the parameter about the number of clusters is not required. The DBSCAN is also suitable for data with arbitrary shapes.

The disadvantage of the original DBSCAN presented in [96] is that the clustering result is highly sensitive to the parameters (i.e., the minimum number of points in a neighborhood and the radius of the neighborhood). To overcome the drawback of the original DBSCAN, some methods extended from DBSCAN are introduced to select parameters heuristically, such as OPTICS [97], Density and distance-based clustering [98].

Sensor Fusion

Sensor fusion, also known as data fusion, is the process to produce more consistent and accurate information through integrating multiple data sources. Sensor fusion technologies have been applied in topics about motion estimation and object tracking for many decades.

The Kalman filter [99], a well-known sensor fusing algorithm, is presented to fuse data through estimating a joint probability distribution. The Kalman filter, however, is designed for linear systems. In order to solve problems in a nonlinear system, which exist in most engineering projects, some variants of the Kalman filter are presented. The EKF [100] linearizes the state transition and observation functions by computing their Jacobians. The IEKF is proposed to reduce the error caused by the linearization. The UKF that presented in [8] solves nonlinear problems by estimating mean and covariance using unscented transformation. For a complex nonlinear system, the particle filter algorithm [101] was developed based on Monte Carlo methods to predict mean and covariance using a large number of particles. Moreover, factor graph-based optimization algorithms, such as GTSAM [102], g20 [103], are also able to be utilized for fusing information.

These algorithms, however, assume that the given information is independent. In other words, there is no correlation between the given information. The assumption can not be guaranteed in many situations, and the correlation between information is often unknown. Julier et. al. presented a non-divergent estimation algorithm, named CI [8], to solve sensor fusion problem with unknown correlations. Uhlmann then extended the CI algorithm to the Covariance Union (CU) algorithm [104] for performing fault-tolerant distributed data fusion.

2.2.2 Lidar-based Localization

The lidar-based localization problem is related to research topics about lidar-based motion estimation, also known as lidar odometry estimation. The goal about these problems is to estimate the pose of the lidar using scans (i.e., point clouds from lidar) and velocity estimation from other sensors, such as IMUs, wheel encoders. The straightforward solutions are to compute the transformation between scans through point cloud registration methods, such as ICP, NDT [105, 106, 107]. These methods, which are always processed with raw scans or randomly sampled point clouds, however, are too expensive in computation to apply in online applications and not robust in complex environments. To overcome these disadvantages, features, which are extracted from raw point clouds, are applied in point cloud registration methods. Zhang et al. presented an edge-surface feature-based lidar odometry algorithm that is able to perform lidar motion estimation at 10 Hz[40]. Shan et al. treated ground points as another feature type when performing lidar motion estimation [41], which is extended from the algorithm presented in [40]. Because of noises from sensors, the errors from point cloud registrations that are accumulated at each time step, and then the performance of these lidar odometry algorithms become worse over time. Meanwhile, the estimations would diverge quickly in dynamic environments since point cloud registration methods, such as ICP and NDT, are designed for static environments (i.e., all objects around the lidar in the environment are static with respect to the environment). Although the errors generated from lidar odometry can be regulated through back-end optimization frameworks in SLAM (e.g., factor-graph based optimization algorithms Georgia Tech Smoothing and Mapping (GTSAM) [102] and g20 [103]), loop closure detection, which is a necessary step in back-end optimization frameworks, is sensitive with the moving objects in environments. Therefore, to develop a robust lidar-based motion estimation algorithm to be suitable for dynamic environments is important.

In a dynamic environment, solutions of current lidar odometry methods are to detect and discard dynamic objects or to track moving objects [81]. An Expectation-Maximization (EM) based algorithm is designed to detect static and dynamic objects by updating the probabilistic estimates related to each measured points using the currently estimated pose and the current map [108]. Wang et al. developed an algorithm, named SLAM with Moving Objects Tracking (SLAMMOT), to detect and track dynamic objects using a map built by SLAM [109]. A number of methods are developed utilizing multiple-layer maps to detect and remove outliers [46, 110, 111]. For these dynamic object detection algorithms, an assumption is required which is that the pose of the lidar is estimated and known before detecting dynamic objects. In other words, these algorithms utilize information from both dynamic and static objects to estimate pose. Although the errors from lidar odometry would be regulated in back-end optimization, the pose estimated partially based on dynamic objects would lead the dynamic objects detection algorithm to fail at some time. With the development in Deep Learning, features are able to be detected from point clouds through algorithms, such as VoxelNet [112], PointCNN [113], PV-RCNN [114], RangeNet++ [48]. Objects are able to be recognized in a point cloud through applying semantic segmentation algorithms (e.g., PointNet++ [115]). Chen et al. proposed an algorithm, called SUMA++, which attempted to remove the dynamic objects in the map using semantic information [47]. However, due to the lack of different types of labeled data (currently, the available labeled data is mainly focused on cars, pedestrians, and cyclist [116]), the objects in point clouds could not be classified in many different types.

Meanwhile, the semantic information could not represent the dynamic objects in some cases, such as a car which is parking off the street cannot be treated as a dynamic objects. Several moving object segmentation algorithms are presented, such as Lidar-MOS [45], removert [43]. However, these segmentation algorithms required pose estimates as input, and were only perform offline, which cannot be used for online odometry estimation. Meanwhile, compared with images, point clouds only present sparse information about the environment. In other words, it is more difficult to accurately predict the semantic information of objects from one point cloud.

3

Cooperative UAV Localization Using Inter-vehicle Ranging and Magnetic anomaly measurements

3.1 Introduction

The content of this chapter has been reproduced based on the author's previous works [15, 63]. The challenges about localization in GNSS-denied environments with low-resolution features could be the mismatching to multiple positions on a map, which creates ambiguity. One of the positioning applications in environments with only low-resolution features is to do localization mainly relying on scalar field measurements as discussed in Chapter 1. Scalar fields associate a scalar value with every point in space, and applications include gravity anomaly [14], magnetic anomaly [15, 4], topographic [16], and olfaction [17], to name a few. In this chapter, localization mainly using magnetic anomaly information is discussed. The Earth's magnetic anomalies present the high spatial frequency content of the Earth's magnetic field. Additionally, the magnetic anomaly information has been measured for most regions in the world [4]. Even in an indoor environment, the magnetic field can be mapped for supporting vehicles' navigation systems [57, 58]. Another reason for using magnetic anomaly information in localization is due to the technological maturity of magnetometers. For instance, optically pumped cesium magnetometers, which can achieve an accuracy of 0.1 nanoTesla(nT), have been used to create magnetic anomaly maps in geological surveys [59, 60].

Several research groups have performed single-vehicle navigation studies using magnetic anomalies as the primary source of information [4, 117, 118]. For example, Canciani et al. developed a navigation filter through a fingerprint matching method to successfully estimate a single UAV's global pose using the Earth's magnetic anomaly map and a navigation grade-INS (inertial navigation system) [4]. However, these magnetic anomaly-based navigation approaches were performed at a low altitude, which means the map is with high-resolution features. As altitude increases, the spatial frequency content of a magnetic anomaly field decreases [119], which directly affects navigation performance. Meanwhile, for applications like long-distance flying, the altitude of the vehicle cannot be kept at a low value all the time. Therefore, a localization algorithm needs to be developed to achieve localization mainly using scalar field information in low-resolution.

Compared to a single agent, a group of cooperative agents may provide several navigational benefits, such as reduced dead-reckoning error, tolerance against individual vehicle or sensor failures, distribution of sensors across a larger spatial area, and shared observations (e.g., landmark). This Chapter addresses the problem of cooperative localization for a small group of UAVs in a GNSSdenied environment. The developed method has two main components. First, a cooperative range localization algorithm is designed to reduce the dead-reckoning error growth for each UAV's global pose estimate and obtain a reliable estimate of relative positions inside the UAV group to be used as an input to a cooperative magnetic localization algorithm. The UAV poses are estimated by sharing inter-vehicle ranging and visual-inertial odometry information. Two communication scenarios are considered, the first one assumes that each UAV can communicate with all other UAVs at once, and the second scenario assumes that each UAV can only communicate with one other UAV at each point in time. Using the shared measurements, an EKF is running on-board each UAV to estimate the poses of all UAVs in the group. In the second step, the cooperative magnetic localization algorithm bounds the error growth of the UAVs' global poses using a particle filter to match magnetic anomaly measurements acquired by each UAV to a prior magnetic anomaly map based on the estimated UAV group shape. Simulation results show substantial improvements in navigation performance using the cooperative range localization algorithm as compared to the deadreckoning performance of each individual UAV. In addition, the developed cooperative magnetic localization approach shows performance and robustness benefits compared to cooperative magnetic localization using only a single UAV. Meanwhile, compared with the approach presented in [12], the proposed algorithms are able to estimate the global pose with lower computation requirements due to the significantly less number of states used in the particle filter. This would in turn allow the application of this cooperative navigation algorithm on a larger UAV group.

The remaining sections of the chapter proceed as follows. Section 3.2 provides a problem statement with associated assumptions. In Section 3.3, methods are presented for solving the relative localization problem in two communication scenarios as well as for performing cooperative magnetic navigation. Section 3.4 presents the simulation configurations and parameters used for evaluating these developed algorithms. In Section 3.5, simulation results are presented and analyzed. Section 3.6 concludes the contributions of this chapter and identifies the current limitations and future research directions.

3.2 **Problem Statement**

In this chapter, the case is considered where a group of N UAVs (N is assumed to be even number here due to communication requirements to be discussed later) are entering a GNSS-denied environment. The main objective is to achieve accurate relative localization among all UAVs in the group as well as to estimate their global poses.

During the GNSS-denied period, the UAVs are assumed to not have access to major visual landmarks or ranging beacons, which is the environment with only low-resolution features. Each UAV is assumed to be able to perform visual-inertial odometry, which provides estimates of vehicle velocity and yaw rate. Each UAV is assumed to be equipped with radios that enable it to exchange sensor measurements between UAVs, as well as performing inter-vehicle ranging measurements. Note that the range between UAVs is assumed to be measured through the communication signal, so each UAV is only able to obtain the range to another UAV at the moment they are communicating.
Each UAV also performs point measurements of the local magnetic field anomaly, which can be exchanged through the communication links. In addition, each UAV is assumed to be loaded with a known magnetic anomaly map. All UAVs are assumed to be flying at a same altitude, which means that this study only deals with the navigation problem in 2D. The initial position in the global frame of reference for each UAV is assumed to be known with a small uncertainty upon entering the GNSS denied environment. No additional information is used for UAV navigation in this study.

3.3 Technical Approach

3.3.1 Communication

Communication is an important component of any cooperative localization strategy. In order for the group to operate in a cooperative manner, each UAV must have knowledge of the other UAVs in the group. In this work, each UAV is assumed to have the ability to share information through ranging links where a ranging link is defined as a connection between a pair of UAVs such that each UAV exchanges sensor information and measures the inter-vehicle distance. Two cases are considered for the number of communication links that can be formed by a single UAV at any point in time. In the first case, each UAV can communicate with each other UAV at any points in time. This is the best possible scenario in regards to communication and will be referred to as complete communication. In the second case, each UAV is constrained to having only a single communication link, so each UAV is only able to communicate with one other UAV at any point in time. This scenario is more practically feasible and will be referred to as pairwise communication. This section provides an overview of the communication strategy and limitations for both complete and pairwise communication scenarios.

Complete Communication

In the complete communication scenario, a ranging link is formed between every pair of UAVs. Therefore, each UAV exchanges information with each other UAV at every time step. Consider the graph, $G_C = (V_C, E_C)$, representing the group of UAVs in the complete communication scenario where V_C and E_C are the sets of vertices and edges in G_C respectively. Let $V_C = \{v_1, v_2, \dots, v_N\}$ such that each vertex corresponds to a single UAV. Since a ranging link exists between each pair of UAVs, G_C is a complete graph on N vertices with the edge set given by

$$E_C = \{ v_i v_j : v_i, v_j \in V_C, v_i \neq v_j \}$$
(3.1)

where $v_i v_j$ denotes the edge incident to v_i and v_j , which corresponds to the ranging link between UAV *i* and UAV *j*. The graphs, G_C , for N = 4, 8 and 16 are presented in Figure 3.3.1.



Figure 3.3.1: Graph of the complete communication scenario for N = 4, 8 and 16 UAVs where each vertex represents a single UAV and each edge represents a ranging link between a pair of UAVs.

Pairwise Communication

In the pairwise communication scenario, each UAV forms a ranging link with one other UAV at each point in time. Therefore, multiple time steps are required in order for each UAV to exchange information with every other UAV. Consider the graph, $G_P = (V_P, E_P)$, representing the group of UAVs in the pairwise communication scenario where V_P and E_P are the sets of vertices and edges in G_P respectively. Let $V_P = \{v_1, v_2, \dots, v_N\}$ such that each vertex corresponds to a single UAV. In contrast to complete communication, multiple possibilities exist for E_P . Since each vertex in G_P is incident to exactly one edge, the number of possibilities for E_P correspond directly to the number of perfect matchings on a complete graph with N vertices. Note that a perfect matching in a graph, G, is an edge set, E, such that every vertex in G is incident to exactly one edge in E. Therefore, at least $\frac{(2L)!}{2^L L!}$ possibilities for E_P , only the perfect matchings given by $E_0 = \{v_1v_2, v_3v_4, \dots, v_{N-1}v_N\}$, $E_1 = \{v_2v_3, v_4v_5, \dots, v_Nv_1\}$, and $E_2 =$ $\{v_1v_{\frac{N}{2}+1}, v_2v_{\frac{N}{2}+2}, \dots, v_{\frac{N}{2}}v_N\}$ are considered. Note that these edge sets were chosen to produce the perfect matchings for N = 4. Therefore, additional possibilities exist for the chosen edge sets. Now, let E_P be defined by

$$E_P = \{E_0, E_1, E_2\} \tag{3.2}$$

where only one element of E_p is used at each iteration of the cooperative localization algorithm. Each of the edge sets produces isomorphic graphs; however, the isomorphisms are not label-preserving, so each edge set provides a different set of measurements. The graphs for N = 4, N = 8, and N = 16 are presented in Figure 3.3.2.



Figure 3.3.2: Pairwise communication graphs where the white nodes represent v_1 and the black nodes represent $\{v_2, v_3, \dots, v_N\}$ in each graph.

Exchange of Information

An important aspect of the communication strategy is the number of steps required to propagate a piece of information throughout the entire group of UAVs. In the case of complete communication, only a single time step is needed to propagate a piece of information throughout the group. In contrast, multiple steps are required to propagate information throughout the group for pairwise communication. In order to calculate the required number of steps, a graph can be constructed representing the propagation of information with respect to the first UAV. This is shown in Figure 3.3.3 where the edges represent the connections made at each step and the vertices represent each source of information. Therefore, after k discrete time steps, the length of the path from v_i to v_1 is the number of steps required to propagate a piece of information from v_i to v_1 .

Now, consider the number of vertices added to the graph at each step prior to k - 2. For E_0 and E_1 , exactly 4 vertices are added to the graph. For E_2 , each edge only provides a connection to an existing vertex, so the number of vertices remain the same. Thus, vertices are only added to



Figure 3.3.3: Propagation of information for pairwise communication by cycling between E_0 at k, E_1 at k-1, and E_3 at k-2 repeatedly.

the graph for E_0 and E_1 prior to k - 2. For steps k and k - 1 exactly two vertices are added to the graph separately, and for steps k - 2 exactly four vertices are added to the graph, as shown in Figure 3.3.3. Therefore, the number of vertices reached after k discrete time steps is given by

$$m = \begin{cases} 2(k+1) & \text{if } k \le 1 \\ 4k - 4 \left\lfloor \frac{k-2}{3} \right\rfloor \text{ otherwise} \end{cases}$$
(3.3)

where $k \in \{0, 1, 2, \dots\}$ and $\lfloor \rfloor$ is the floor function. Now, the number of steps required for one UAV to obtain the information of the rest of the group, denoted s, is bounded by $s \geq \lfloor \frac{3}{8}N \rfloor$ where N is the number of UAVs in the group and $\lfloor \rceil$ represents the ceiling function.

3.3.2 Cooperative Ranging Localization

The goal of the cooperative ranging localization algorithm is to obtain a reliable estimate of the relative position of each UAV in the group for input to the cooperative magnetic localization algorithm introduced in Section 3.3.3. The problem is formulated as a state estimation problem with an EKF. In general, assuming additive noise, a discrete nonlinear dynamic system can be described by the state transition and observation models provided by

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \tag{3.4}$$

$$\mathbf{z}_k = \boldsymbol{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{3.5}$$

where **f** is the vector-valued discrete state prediction function, h is the vector-valued observation function, **x** is the state vector, **z** is the output vector, **u** is the measured control vector, **w** is the process noise vector, **v** is the measurement noise vector, and k is the discrete time index. Both the process and measurement noises are assumed to be multivariate Gaussian white noises with covariances Q and R, where $\mathbf{w}_k \sim N(0, Q_k)$ and $\mathbf{v}_k \sim N(0, R_k)$, respectively. The state vector in the EKF is given by

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{\pi}^{(1)}, \ \boldsymbol{\pi}^{(2)}, \ \dots, \ \boldsymbol{\pi}^{(N)} \end{bmatrix}^T$$
(3.6)

where $\boldsymbol{\pi}^{(i)} = \begin{bmatrix} x^{(i)}, y^{(i)}, \theta^{(i)} \end{bmatrix}^T$ is the pose of UAV i in the global frame such that $i \in 1, 2, \cdots, N$ where x and y are the Cartesian coordinates for the position and θ is the heading. The discrete time state transition function is given by $\mathbf{f} = \begin{bmatrix} \mathbf{f}^{(1)}, \mathbf{f}^{(2)}, \dots, \mathbf{f}^{(N)} \end{bmatrix}^T$ where $\boldsymbol{\pi}_k^{(i)} = \mathbf{f}^{(i)} \left(\boldsymbol{\pi}_{k-1}^{(i)}, \mathbf{u}_k^{(i)} \right)$ is the discrete state transition function for UAV i given by

$$\mathbf{f}^{(i)}(\boldsymbol{\pi}_{k-1}^{(i)}, \mathbf{u}_{k}^{(i)}) = \begin{bmatrix} x_{k-1}^{(i)} \\ y_{k-1}^{(i)} \\ \theta_{k-1}^{(i)} \end{bmatrix} + T_{s} \begin{bmatrix} v_{k}^{(i)} \cos\left(\theta_{k-1}^{(i)} + T_{s}\omega_{k}^{(i)}\right) \\ v_{k}^{(i)} \sin\left(\theta_{k-1}^{(i)} + T_{s}\omega_{k}^{(i)}\right) \\ \omega_{k}^{(i)} \end{bmatrix}$$
(3.7)

where $v_k^{(i)}$ is the velocity for UAV i, $\omega_k^{(i)}$ is the yaw rate for UAV i at time step k, and T_s is the change in time between discrete time steps.

The observation function varies depending on the communication graph at each step. The observation function for complete communication is given by

$$\mathbf{h}_{C}(\hat{\mathbf{x}}_{k|k-1}) = \{\sqrt{(x^{(i)} - x^{(j)})^{2} + (y^{(i)} - y^{(j)})^{2}} : (v_{i}, v_{j}) \in E_{C}\}$$
(3.8)

where \mathbf{h}_C is dependent on the the edge set E_C . Notice the observation function is identical at each step, and since each UAV exchanges information with each other UAV, each UAV can perform an update at each step using all inter-vehicle ranging measurements. In contrast, the observation function for pairwise communication varies between steps, which is given by

$$\mathbf{h}_{P}(\hat{\mathbf{x}}_{k|k-1}) = \{\sqrt{(x^{(i)} - x^{(j)})^{2} + (y^{(i)} - y^{(j)})^{2}} : (v_{i}, v_{j}) \in E_{k \pmod{3}}\}$$
(3.9)

where \mathbf{h}_P is dependent on the the edge set for E_P . Note that each UAV only communicates with a single UAV at each step. Therefore, multiple steps are required to propagate the ranging and odometry information throughout the group of UAVs before each UAV can perform an update. The number of required steps for N UAVs is given by Eq. 3.3. Therefore, the state estimate at any point in time is obtained by adding steps of dead-reckoning onto the state estimate obtained from most recent EKF update. This is given by

$$\hat{\mathbf{x}}_{k} = \mathbf{x}_{k-s} + \Delta \mathbf{x} \Big|_{k-s}^{k}$$
(3.10)

where s is the number of steps since the most recent EKF update and $\Delta \mathbf{x} \Big|_i^j$ is the change in pose

between time steps i and j from dead-reckoning. The measurement vector is given by $\mathbf{z}_k = \{d_{ij} : (v_i, v_j) \in E_C\}$ for complete communication and $\mathbf{z}_k = \{d_{ij} : (v_i, v_j) \in E_k \pmod{3}\}$ for pairwise communication, where d_{ij} is the distance between UAV i and UAV j as measured by the ranging sensors.

The standard first-order EKF equations [121] are used for the state prediction and update where the observation function is dependent on the communication scenario.

The relative position of each UAV with respect to other UAVs in the group can be derived through

$$\begin{bmatrix} \hat{x}^{(i/j)} \\ \hat{y}^{(i/j)} \end{bmatrix} = \begin{bmatrix} \hat{x}^{(i)} \\ \hat{y}^{(i)} \end{bmatrix} - \begin{bmatrix} \hat{x}^{(j)} \\ \hat{y}^{(j)} \end{bmatrix}$$
(3.11)

where $[\hat{x}^{(i/j)}, \hat{y}^{(i/j)}]^T$ are the relative position coordinates for UAV i with respect to UAV j, $[\hat{x}^{(i)}, \hat{y}^{(i)}]^T$ and $[\hat{x}^{(j)}, \hat{y}^{(j)}]^T$ are the position estimates of UAV i and UAV j from Eq. 3.10, respectively.

Some limitations of the cooperative ranging localization should be mentioned in regard to observability. First, if the UAVs are traveling at same speed and direction with parallel trajectories, an infinite number of solutions exist for the relative position of each UAV. This phenomenon is discussed in detail in [122] and could potentially result in divergence of the EKF. This can be avoided by varying the velocity of each UAV and is discussed further in Section 3.4. Second, since ranging-only measurements are used for updating the poses, the EKF is only capable of preserving the pairwise distances. As a result, only the geometric structure of the group is maintained through the EKF. Thus, the geometric structure will rotate in the global frame with an angle γ , which is exactly the rotation error of the group if the geometric structure is known without error. This is a well-studied property described by Euclidean distance matrices [123, 124]. The method for approximating γ to reduce the rotation error and recover the global pose of the group is discussed in the following section.

3.3.3 Cooperative Magnetic Localization

The goal of the cooperative magnetic localization algorithm is to bound the error growth of the global pose by matching the magnetic anomaly measurements on-board each UAV in the group to a prior magnetic anomaly map. This is achieved by leveraging the cooperative ranging localization solution to fix the relative position of each UAV (i.e., the group shape). The global position is maintained with a Bayes filter using a prediction-update framework. Since the magnetic anomaly map is highly non-linear, a particle filter is selected for solving this problem.

For reducing the rotation error of the group during flight, γ introduced in previous section is defined as a state. An example of the rotation error of the group is presented in Figure 3.3.4 for the four-UAV case. In Figure 3.3.4, the gray node is the position of UAV j, the black nodes connected



Figure 3.3.4: Example of rotation error of the group in relative position extracted from EKF for N = 4.

with dashed lines are estimated relative positions of remaining UAVs with respect to UAV j from the EKF, and the white nodes connected with solid lines are the true relative positions of remaining UAVs with respect to UAV j. The relative position estimates of each UAV with respect to the other UAVs in the group can be corrected based on γ as,

$$\begin{bmatrix} \dot{x}^{(i/j)} \\ \dot{y}^{(i/j)} \end{bmatrix} = \begin{bmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} \hat{x}^{(i/j)} \\ \hat{y}^{(i/j)} \end{bmatrix}$$
(3.12)

where $[\hat{x}^{(i/j)}, \hat{y}^{(i/j)}]^T$ are the relative position coordinates for UAV *i* with respect to UAV *j* after applying the update utilizing γ , and $[\hat{x}^{(i/j)}, \hat{y}^{(i/j)}]^T$ are the relative position coordinates for UAV *i* with respect to UAV *j* from the cooperative ranging localization.

An intuitive way to design the filter is to include all UAVs' pose in the state vector. However, more states lead to more complexity and greater chance of overfitting. Therefore, the particle filter introduced in this chapter only involves four states and is independent of group size,

$$\mathbf{p} = [x, y, \theta, \gamma]^T \tag{3.13}$$

where x and y are the Cartesian coordinates for the global position of a UAV, θ is the heading of a UAV, and γ is the rotation error of the group introduced previously. Note that each UAV runs a particle filter individually to estimate its own pose. Thus, the state transition model for each particle is given by

$$\mathbf{p}_{k} = \mathbf{g} \left(\mathbf{p}_{k-1}, \, \mathbf{u}_{k}^{P} \right) + \, \nu_{k} \tag{3.14}$$

where g is the vector-valued discrete state prediction function, \mathbf{u}_k^P is the measured control vector of a UAV, ν_k is the process noise vector, and k is the discrete time index. Similar to Eq. 3.7, the vector-valued discrete state prediction function is given by

$$\mathbf{g}\left(\mathbf{p}_{k-1}, \mathbf{u}_{k}^{P}\right) = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \\ \gamma_{k-1} \end{bmatrix} + T_{s} \begin{bmatrix} v_{k}\cos\left(\theta_{k-1} + T_{s}\omega_{k}\right) \\ v_{k}\sin\left(\theta_{k-1} + T_{s}\omega_{k}\right) \\ \omega_{k} \\ 0 \end{bmatrix}$$
(3.15)

where v_k is the velocity for the UAV, ω_k is the yaw rate for the UAV, and T_s is the change in time between discrete time steps. The state γ is propagated by random walk. The observation model for each particle is given by

$$\mathbf{y}_{k} = \boldsymbol{h}_{M}\left(\mathbf{p}_{k}, \, \mathbf{r}_{k}\right) + \eta_{k} \tag{3.16}$$

where \mathbf{y}_k is the observation vector and η_k is the measurement noise vector. The relative position \mathbf{r}_k is calculated from Eq. 3.11. Therefore, for each UAV, the other UAVs' predicted global positions can be extracted by adding the updated relative positions from Eq. 3.12 to the estimate from the particle filter, shown as

$$\begin{bmatrix} \dot{x}^{(i)} \\ \dot{y}^{(i)} \end{bmatrix} = \begin{bmatrix} x^{(j)} \\ y^{(j)} \end{bmatrix} + \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} \hat{x}^{(i/j)} \\ \hat{y}^{(i/j)} \end{bmatrix}$$
(3.17)

where $[\hat{x}^{(i)}\hat{y}^{(i)}]^T$ is another UAV's predicted global position, $x^{(j)}, y^{(j)}$, and γ are from the UAV's state vector **p**, and $[\hat{x}^{(i/j)}, \hat{y}^{(i/j)}]^T$ is the relative position from **r**. The vector-valued observation function h_M is used to extract the predicted magnetic anomaly measurements from the given map based on each vehicles' predicted positions.

The goal at each time step is to approximate the posterior distribution $p(\mathbf{p}_k|\mathbf{y}_{1:k})$ using a set of

weighted particles. Consider the set of M particles given by

$$\mathbf{P}_{k} = \left[\mathbf{p}_{k}^{(1)}, \mathbf{p}_{k}^{(2)}, \cdots, \mathbf{p}_{k}^{(M)}\right]$$
(3.18)

where $\mathbf{p}_{k}^{(i)} = \left[x_{k}^{(i)}, y_{k}^{(i)}, \theta_{k}^{(i)}, \gamma_{k}^{(i)}\right]^{T}$ is the state vector for the i^{th} particle. The corresponding weights are given by

$$\mathbf{W}_{k} = \left[w_{k}^{(1)}, w_{k}^{(2)}, \cdots, w_{k}^{(M)} \right]$$
(3.19)

where $w_k^{(i)}$ is the weight for the *i*th particle. According to the observation function described in Eq. 3.16, for each particle, there are N different predicted observations corresponding to the N UAVs' predicted positions. In this study, the magnetic anomaly measurement noise is assumed as a Gaussian white noise [4], and the likelihood function can be calculated as a Gaussian distribution. Meanwhile, the magnetic anomaly measurements from different UAVs are assumed to be conditionally independent. Therefore, for each particle, the likelihood function based on all observations at time k is given by

$$p\left(\mathbf{y}_{k}|\mathbf{p}_{k}\right) = \prod_{j=1}^{N} p\left(y_{k}^{(j)}|\mathbf{p}_{k}\right)$$
(3.20)

where $p\left(y_{k}^{(j)} \middle| \mathbf{p}_{k}\right) = \frac{1}{\sqrt{2\pi\sigma_{m}^{2}}} \exp\left(-\frac{\left[y_{k}^{(j)}-t_{k}^{(j)}\right]^{2}}{2\sigma_{m}^{2}}\right)$ such that $y_{k}^{(j)}$ is the predicted observation on UAV j based on the predicted state vector \mathbf{p}_{k} , $t_{k}^{(j)}$ is the magnetic anomaly measurement from UAV j's on-board magnetometer, and σ_{m} is the standard deviation of the magnetic anomaly measurement.

The i^{th} particle's weight is given by the likelihood function of the observations at time k and its

normalized weight in the previous time step, $w_k^{(i)} = p\left(\mathbf{y}_k \middle| \mathbf{p}_k^{(i)}\right) \widetilde{w}_{k-1}^{(i)}$ where $\widetilde{w}_{k-1}^{(i)}$ is the normalized weight. The weights \mathbf{W}_k are then normalized such that the sum of all weights equal one. The expectation of the state at time k is given by

$$E[\mathbf{P}_k] \approx \sum_{i=1}^M \tilde{w}_k^{(i)} \mathbf{P}_k^{(i)}.$$
(3.21)

Also, similar to the cooperative ranging localization method discussed earlier, the cooperative magnetic localization approach requires additional steps to gather all information in the group for the pairwise communication.

3.4 Simulation

The cooperative navigation methods are evaluated using simulated data sets for groups of UAVs flying in a two-dimensional environment. The simulations are performed only for the GNSS-denied portion of the flight where the initial pose of each UAV is assumed to be known with a small uncertainty. In the simulation, the odometry measurements are obtained at a frequency of 10 Hz and the ranging measurements and magnetic anomaly measurements are exchanged at a frequency of 5 Hz. The coverage of the ranging links are assumed to be sufficiently large; therefore, the UAVs are able to form ranging links regardless of their relative positions. The magnetic anomaly map, shown in Figure 3.4.1, is generated using an upward continuation function [125] with a Gaussian white noise $G \sim \mathcal{N}(0, 1000)$ nanotesla.

The simulation study is broken into two parts. The first part consists of multiple case studies, which are performed for each of the presented communication scenarios for different group sizes



Figure 3.4.1: The low fidelity magnetic anomaly map used in the simulation study.

(N = 4, 8, and 16). Each of these case studies are performed where the UAVs are flying along parallel trajectories as shown in Figure 3.4.2a. In this case, each UAV is traveling along a path that is parallel to the paths of other UAVs while the speed is slowly varing following a sine function throughout the duration of the flight. The speed offsets for each UAV are generated randomly so each UAV is traveling at different speed as shown in Figure 3.4.2b.

For the case studies, the velocity errors are drawn from a zero mean Gaussian distribution with standard deviation $\sigma_v = 0.1$ m/s and a turn on bias, $b_v \sim \mathcal{N}(0, 0.1\sigma_v)$. The yaw rate errors are drawn from a zero mean Gaussian distribution with standard deviation $\sigma_h = 0.1$ deg/s with a turn on bias, $b_h \sim \mathcal{N}(0, 0.1\sigma_h)$. The ranging errors are drawn from a zero mean Gaussian distribution with standard deviation $\sigma_r = 1$ m. Finally, the magnetic anomaly measurement errors are drawn from a zero mean Gaussian distribution with standard deviations $\sigma_m = 100$ nanotesla.

Sensitivity analysis is then performed to evaluate the effect of UAV group geometry and sensor quality on the cooperative range localization algorithm performance, as well as the robustness



Figure 3.4.2: UAV trajectories used in the simulation study. Subplot 3.4.2a is an example set of parallel trajectories. Each UAV is presented with a different color, dots are the positions of all UAVs and arrows show the headings of UAVs. k present time steps, and $n = t/T_s$, where t is the simulation time and T_s is the sampling time; Subplot 3.4.2b shows one example of the time-varying UAV speeds associated with the parallel trajectories ($0 \sim 1,000$ seconds).

of the cooperative magnetic localization algorithm with respect to the initial position error uncertainty. For both the case studies and the sensitivity analysis, multiple Monte Carlo simulations are performed consisting of 100 trials each for a flight duration of 1 hour. The number of particles for the cooperative magnetic localization algorithm is set to 10,000 in each simulation.

3.5 Results

An example bird's-eye view of a single representative simulation where N = 4 is shown in Figure 3.5.1. From the simulations, both the cooperative range localization and cooperative magnetic localization perform similarly for each UAV. Therefore, since the algorithms running on each UAV are identical, the estimation error is only evaluated for a single UAV in the group in this section.



Figure 3.5.1: Bird's-eye view of the estimated and truth trajectories for a simulation with N = 4 using pairwise communication. Note that the estimated cooperative magnetic localization trajectories and the truth trajectories are nearly overlapping. (DR: dead-reckoning; RL: cooperative range localization; ML: cooperative magnetic localization.)

3.5.1 Case Studies

An important component of the cooperative magnetic localization algorithm is the relative position extracted from the EKF in the cooperative range localization algorithm. However, the relative position estimates suffer from heading drift resulting in large error over time due to the rotation error, γ . Since the shape of the group with some unknown rotation, γ , is completely defined given the inter-vehicle distances, the inter-vehicle distance errors are used to analyze the quality of the group shape estimated by the EKF. For a set of 100 simulations, the average Root Mean Square Errors (RMSEs) of the distance between each pair of UAVs are presented in Table 3.5.1.

In Table 3.5.1, the RMSE of the Euclidean distance error decreases as the number of UAVs increase in the complete communication scenario. This is due to the fact that the number of observations at each time increase exponentially with the number of UAVs, which is used to update

	Pairwise	Complete
N = 4	0.624	0.450
N = 8	0.738	0.347
N = 16	1.02	0.292

 Table 3.5.1: RMSE of Euclidean distance error from EKF output. (Unit: Meters)

the relative positions. In contrast, the Euclidean distance error increases as the number of UAVs increase in the pairwise communication scenario. This is most likely due to the fact that the number of observations increase only linearly with the number of UAVs in the group. This reduces the overall connectivity of the graph compared to the complete communication scenario resulting in larger relative position errors.

Figures 3.5.2 shows the Cumulative Distribution Function (CDF) of the heading and position RMSE of one UAV in the group acquired through the Monte Carlo simulations. It shows that cooperative range localization provides significant benefit compared to the dead-reckoning of an individual UAV. Most of the performance gain is shown to be provided by the reduced heading error, which is regulated in the EKF with the velocity measurements provided by cooperating UAVs from large distances. The cooperative localization performance also improves when the number of UAVs in the group increase. It is interesting to note that the performances between the complete communication scenario and the pairwise communication scenario are similar. The difference in distance errors for both communication scenarios are insignificant compared to the actual distance between UAVs in the simulations. This indicates that most of the navigational benefits may be achieved with only a single communication link on each UAV, which is practically more feasible.

A cooperative magnetic localization simulation result with four UAVs in the group using pairwise communication scenario is shown in Figure 3.5.3 as an example. Unlike the solutions of dead-



Figure 3.5.2: The CDF of heading (left) and position (right) RMSE for one UAV in the group with dead-reckoning (black) and cooperative range localization (blue) in both complete communication (solid lines) and pairwise communication (dashed lines) scenarios. The sizes of the group are N = 4 (top), N = 8 (middle) and N = 16 (bottom). (DR: dead-reckoning; RL (cc): cooperative range localization in complete communication; RL (pc): cooperative range localization.)

reckoning and cooperative range localization, the cooperative magnetic localization result does not grow over time.

The cooperative magnetic localization performance is evaluated by comparing position RMSE in different group sizes and communication scenarios. The average means and standard deviations of position RMSE for one UAV in the group in Monte Carlo simulations are shown in Table 3.5.2. It shows that cooperative magnetic localization (i.e. N > 1) provides significant performance gain compared to a single UAV cooperative magnetic localization. However, the number of UAVs in the group do not seem to have a major effect to the navigation performance. Note that the single UAV cooperative magnetic localization in this study applies the same particle filter model presented in



Figure 3.5.3: Position error for one UAV in the group in dead-reckoning (black), cooperative range localization (blue) and cooperative magnetic localization (red) in one representative simulation.

Section 3.3.3 with the rotation error, γ , set as zero.

Table 3.5.2: The means and standard deviations of position RMSE for one UAV in the group in cooperative magnetic localization with 100 times simulations. (Unit: Meters)

	Con	nplete	Pairwise		
	Mean	Std dev	Mean	Std dev	
N = 1	35.0	37.1	35.0	37.1	
N = 4	8.61	8.01	9.26	8.18	
N = 8	6.96	7.97	10.2	8.88	
N = 16	7.46	8.81	9.12	8.48	

3.5.2 Sensitivity Analysis

The sensitivity analysis simulations are first performed with 4 UAVs and the pairwise communication scenario to evaluate the effect of UAV group geometry on the cooperative range localization performance. In this case, the velocity, yaw rate, and ranging noise are held constant at $\sigma_v = 0.1$ m/s, $\sigma_h = 0.1$ deg/s, and $\sigma_r = 1$ m respectively, while varing each UAV's speed variation range. Meanwhile, the distances between each pair of neighboring UAVs' initial positions are set to 2, 500 m. The CDF for the position RMSE for one UAV in the group using cooperative range localization algorithm with four speed variation ranges are presented in Figure 3.5.4a. Similarly, using the same simulation parameters and a fixed speed variance range of 10 m/s, the distances between each pair of neighboring UAVs' initial positions are varied. The CDF of position RMSE for one UAV in the group in cooperative range localization with four different initial distances values are presented in Figure 3.5.4b.



Figure 3.5.4: (a): The CDF of position RMSE for one UAV in the group in cooperative range localization with varied amplitudes of each UAV's speed variation range. (Unit of A: Meters per second) (b): The CDF of position RMSE for one UAV in the group in cooperative range localization with varied distances between each pair of neighboring UAVs' initial positions. (Unit of D: Meters)

Figure 3.5.4 shows that the cooperative range localization performance improves with increased speed variations and increased UAV separations. Both error factors may contribute to the creating of more favorable geometries that allow better constraints on the UAV heading growth. The effects of these factors, however, are nonlinear, which will be investigated in the follow-on studies.

The impact of measurement noises on the cooperative range localization performance is then evaluated. In this set of simulations, the amplitudes of each UAV's speed variation are set to 10 m/s and the distances between each pair of neighboring UAVs' initial positions are set to 2, 500 m. Four UAVs are simulated with the pairwise communication scenario. First, the ranging and yaw rate noises are held constant at $\sigma_r = 0$ m and $\sigma_h = 0$ deg/s respectively, while varying the velocity noise. The CDF for the position RMSE for one UAV in the group in cooperative range localization with three different velocity noise assumptions are presented in Figure 3.5.5a. Second, the velocity and ranging noises are held constant at $\sigma_v = 0$ m/s and $\sigma_r = 0$ m respectively, while varying the yaw rate noise. The CDF for the position RMSE for one UAV in the group in cooperative range localization with three different yaw rate noise assumptions are presented in Figure 3.5.5b. Third, the velocity and yaw rate noises are held constant at $\sigma_v = 0$ m/s and $\sigma_h = 0$ deg/s respectively, while varying the ranging noise. The CDFs for the position RMSE for one UAV in the group in cooperative range localization with four different ranging noise assumptions are presented in Figure 3.5.5c.



Figure 3.5.5: (a): The CDFs of position RMSE for one UAV in the group in cooperative range localization with varied velocity noises. (Unit of σ_v : Meters per second) (b): The CDF of position RMSE for one UAV in the group in cooperative range localization with varied yaw rate noises. (Unit of σ_h : Degrees per second) (c): The CDFs of position RMSE for one UAV in the group in cooperative range localization with varied ranging noises. (Unit of σ_r : Meters)

Figures 3.5.5 shows that the EKF used in the cooperative range localization algorithm can toler-

ate large variations of sensor noises, with the exception of cases with poor ranging measurements (in the case when $\sigma_r = 10$ m). The UAV position estimation performance is roughly proportional to the quality of sensor performance provided.

The impact of magnetic anomaly measurement noise on the cooperative magnetic localization performance is also evaluated. In this set of simulations, the velocity, yaw rate and ranging noise are held constant at $\sigma_v = 0.1 \text{ m/s}$, $\sigma_h = 0.1 \text{ deg/s}$ and $\sigma_r = 1 \text{ m}$ respectively, while varing the magnetic anomaly measurement noise. Meanwhile, the amplitude of the speed variation and the distances between each pair of neighboring UAVs' initial positions are set to 10 m/s and 2,500 mrespectively. The CDFs for the position RMSE for one UAV in the group in cooperative magnetic localization with four different magnetic anomaly measurement noise assumptions are presented in Figure 3.5.6. It shows that the cooperative magnetic localization algorithm can tolerate large variations of magnetic sensor quality.



Figure 3.5.6: The CDFs of position RMSE for one UAV in the group in cooperative magnetic localization with varied magnetic anomaly measurement noises. (Unit of σ_m : Nanotesla)

To evaluate the robustness of the cooperative magnetic localization algorithm presented in this chapter, the UAVs' assumed initial positions are offset from the truth. In this set of simulations, the same simulation parameters as the previous simulations and a fixed magnetic anomaly measurement noise $\sigma_m = 100$ nanotesla are used. An offset is added to the assumed initial position for all the UAVs with a constant distance R and a random direction. The CDF for the position RMSE in the steady state (i.e. the second half of each simulation run) for one UAV in the group using the cooperative magnetic localization algorithm with two different initial position offset assumptions in four different group sizes (N = 1, 4, 8 and 16) are presented in Figure 3.5.7. The means and standard deviations of position RMSE in the steady state for one UAV in the group, as well as the percentage of cases that the filter converges are listed in Table 3.5.3. Note that in each simulation, the particle filter is not aware of this initial position offset and assumes 1 m standard deviation uncertainty in each UAV's initial position.



Figure 3.5.7: The CDF of position RMSE in the steady state for one UAV in the group using the cooperative magnetic localization algorithm with two initial position offset assumptions in different group sizes. (Unit of *R*: Meters)

Table 3.5.3: The means and standard deviations of position RMSE in the steady state, as well as the percentage of cases that the filter converges using the cooperative magnetic localization algorithm with two initial position offset assumptions in different group sizes. (Unit: Meters)

	N = 1		N = 4		N = 8		N = 16	
	R = 10m	R = 100m						
Mean	55.3	52.2	13.5	59.9	12.3	47.6	9.06	28.5
Std dev	51.5	44.3	13.8	53.4	11.4	39.0	10.8	30.6
Convergence%	98%	36%	100%	94%	100%	75%	99%	46%

Figure 3.5.7 and Table 3.5.3 show that the cooperative magnetic localization algorithm can mostly tolerate small initial group position offset, such as R = 10 m. In this case, the means and standard deviations of position RMSE in the steady state are similar with results in Table 3.5.2. When the

initial group position offset increased to R = 100 m, the cooperative magnetic localization is shown to be more robust than the single UAV magnetic localization. However, simulation study shows that the robustness decreases with the increased number of UAVs in the group. This is due to a reduced probability of matching multiple UAV's magnetic anomaly measurements to the prior map. Methods for solving this issue will be developed in future efforts.

3.6 Discussion

In this chapter, a set of algorithms was presented for cooperatively localizing a group of UAVs in a GNSS-denied environment using shared visual-inertial odometry measurements, magnetic anomaly measurements, inter-vehicle ranging measurements, and a prior magnetic anomaly map. The developed approach has two sequential parts: cooperative range localization and cooperative magnetic localization. Two communication scenarios are considered: complete communication and pairwise communication.

The results from multiple Monte Carlo simulations demonstrate the performance and robustness benefits of the cooperative magnetic localization algorithm assisted by cooperative range localization algorithm with respect to non-cooperative localization approaches. The simulation study also suggests that the pairwise communication approach is capable of delivering a majority of the cooperative navigation benefit as compared to the complete communication scenario, which is more feasible with the existing ranging devices.

There are several limitations to the presented algorithms such as UAV group size and trajectory constraints. Specifically, the algorithms are not directly suited for odd number of UAVs in the pairwise communication scenario; the EKF will not work if all UAVs fly at the same speed along the parallel trajectories. These constraints will be partially removed in following work through ex-

ploring different communication topology and through a tighter coupling between the relative and magnetic navigation steps.

4

Cooperative Navigation Using Pairwise Communication with Ranging and Magnetic Anomaly Measurements

4.1 Introduction

The content of this chapter has been reproduced based on the author's previous works [15, 63]. In the previous Chapter 3, the previous analysis, which was based on simulated map, was not adequate for understanding the performance in more realistic scenarios. Additionally, the group size was restricted to an even number of UAVs in Chapter 3. In this chapter, the cooperative localization method presented in Chapter 3 is expended to support odd numbers of UAVs in the group with pairwise communication. The magnetic anomaly map used in this chapter is obtained from the United States Geological Survey to cover a larger area [18], which is more realistic than the simulated map in Chapter 3. In this chapter, the simulator is expanded to support the UAV control, which relies on the estimated poses for feedback. Meanwhile, in this chapter, the performance and robustness of the presented methods under different sensor noise assumptions and map resolutions are reported.

The remaining sections of the chapter proceed as follows. Section 4.2 provides a problem statement with the associated assumptions. In Section 4.3, methods are presented for solving the relative localization problem as well as for performing cooperative magnetic localization. Section 4.4 presents the simulator and the simulation configurations. Then, in Section 4.5, simulation results are presented and analyzed. Section 4.6 concludes the contributions of this chapter and identifies the current limitations and future research directions.

4.2 Problem Statement

In this study, the case is considered where a group of N UAVs is entering a GNSS-denied environment. The size of N is appropriate for a typical UAV mission (e.g., between 2 and 20). The main objective is to achieve accurate and robust global localization for all UAVs in the group.

All UAVs are assumed to be flying at the same altitude, which means that this study only deals with the navigation problem in 2D. The initial pose in the global frame for each UAV is assumed to be known with a small uncertainty (e.g., provided by GPS) before entering the GNSS-denied environment.

Each UAV is assumed to be equipped with radios that enable communication with all other UAVs to exchange sensor measurements, as well as to perform inter-vehicle ranging. In this chapter, each UAV is restricted to communicate and perform ranging measurements with only one other UAV at any point in time, which is referred as to pairwise communication. Also, each UAV is assumed to be able to store and exchange information of the other UAVs.

Each UAV also performs point measurements of the magnetic anomaly field at each time step, which can be exchanged through the communication link. In addition, each UAV is assumed to be loaded with the same magnetic anomaly map. Meanwhile, each UAV is assumed to be able to measure or estimate its velocity and yaw rate through some methods, such as using Vehicle Dynamic Model (VDM) presented in [126] or Visual Odometry presented in [127, 128]. No additional information is used for UAV navigation in this study.

4.3 Technical Approach

The framework of the cooperative navigation algorithm presented in this chapter is shown in Figure 4.3.1. The communication block is responsible for storing and parsing the historic data shared through the communication links with other UAVs and the data measured by sensors onboard the current UAV. The goal of the cooperative ranging localization step is to obtain a reliable es-



Figure 4.3.1: The framework of the cooperative navigation algorithm.

timate of the relative position of each UAV in the group. The cooperative magnetic localization algorithm is able to bound the error growth of the global pose by matching the magnetic anomaly measurements on-board each UAV in the group to a prior magnetic anomaly map. The details of the cooperative ranging localization and cooperative magnetic localization algorithms design were explained in Sections 3.3.2 and 3.3.3 under Chapter 3, respectively.

4.3.1 Communication

This subsection provides the implementation for odd number of the group size case. Under pairwise communication as introduced in Section 4.2, multiple time steps are required in order for each UAV to exchange information with every other UAV. Initially, in Chapter 3, the communication strategy required an even number of UAVs. In this chapter, scenarios with an odd number of UAVs are expended. The details about the pairwise communication and the exchange of information for even number of the group size were explained in Sections 3.3.1 and 3.3.1 under Chapter 3.

If the group size is an odd number (e.g., N - 1), the bound for the N - 1 UAV case is at most the same as for the N UAV case. This is straightforward from the previous discussion in Sections 3.3.1 and 3.3.1 under Chapter 3. Consider the communication graph for N UAVs (as depicted in Figure 3.3.3). Now, delete a vertex and the incident edges to form a group of N - 1 UAVs, the resulting graph is still connected, and the number of steps required to propagate information remains the same. The resulting edge sets for an odd number of UAVs are obtained by deleting v_N and all edges incident to v_N . The odd edge sets are then given by $E_0^{\text{odd}} = E_0^{\text{even}} \setminus v_{N-1} v_N$, $E_1^{\text{odd}} = E_1^{\text{even}} \setminus v_N v_1$, and $E_3^{\text{odd}} = E_3^{\text{even}} \setminus v_N v_N$.

4.4 Simulator

The cooperative ranging and magnetic localization algorithms are evaluated through a simulator, illustrated in Figs. 4.4.1 and 4.4.2. Since all UAVs use the same simulation model, the framework of one UAV's simulator is presented in Figure 4.4.1 where the purple blocks are user defined inputs and yellow blocks are measurements simulated from sensors and information shared from other UAVs through the communication links. The framework of the proposed algorithms are explained in Figure 4.4.2 where $\mathbf{RG}_{k-s,k-s+1,k-s+2}$ are pairwise ranging measurements from the group at time steps k-s,k-s+1, and k-s+2, $\mathbf{V}_{k-s},\mathbf{\Omega}_{k-s}$, and \mathbf{M}_{k-s} present all UAVs' velocity, yaw rate, and magnetic anomaly measurements at time step k-s, respectively, $\mathbf{v}_{k-s}^{(i)}$ and $\omega_{k-s}^{(i)}$ are the i^{th} UAV's velocity and yaw rate measurements at time step k-s, respectively, \mathbf{r}_{k-s} is the relative position estimates inside the group at time step k - s, $\hat{\mathbf{x}}_{k-s}$ is the global pose estimate at time k - s, and $\Delta \mathbf{x} \Big|_{k-s}^{k}$ is the change in global pose between time steps k - s and k, as described in Eq. 3.10. Note that, in Figure 4.4.2, the communication block is responsible for storing and parsing the historic data shared through the communication links with other UAVs and the data measured by sensors onboard the current UAV. In detail, the data is stored in the shared information block as a set of packets, and the parse data block interprets the packets for input in the other parts of the system. Each packet contains the data collected by all UAVs at time step k, denoted as $D_k = \{d_k^{(1)}, d_k^{(2)}, ..., d_k^{(N)}\}$ where $d_k^{(i)}$ is the data transferred from the i^{th} UAV at time step k. A packet is incomplete (denoted by D_k) if the packet does not contain the data for all vehicles at time step k. The parse data block is responsible for adding missing information to incomplete packets if data is available from packets received from other UAVs at a different time step. Applying the communication scheme presented in Sections 3.3.1 and 4.3.1, the data transferred by each vehicle at time step k is the time history given by $\overrightarrow{P}_k = \{D_k, D_{k-1}, ..., D_{k-s}\}$ where all the packets are incomplete except for the packet containing data for time step k - s. The packets for data recorded prior to time step k - s are dropped as they are not used for localization.



Figure 4.4.1: The framework of the simulator for each UAV.

In the simulator, a bicycle model [129] is used as a simplified UAV model in the 2D plane. The trajectory of each UAV is controlled based on a reference trajectory (including position and velocity) and the estimated vehicle states. In this chapter, all reference trajectories are set to be parallel to latitude lines in the geographic coordinate system, and new parallel reference trajectories, at a 1,000 meters distance, are added to the simulation when the UAV group size increases. All UAVs are assumed to fly from west to east with the same initial longitude and small uncertainties on the initial positions, which are drawn from Gaussian distributions with zero mean and 1-meter standard deviation. The reference trajectories are close to the ground truths shown in Figure 3.5.1. Due



Figure 4.4.2: The overview of the proposed algorithms applied in the i^{th} UAV.

to the observability requirement of the EKF, the reference velocity for each UAV is slowly varying following a sine function throughout the duration of the flight as shown in Figure 3.4.2b.

A controller steers the UAV based on the global pose estimates from the presented algorithms as feedback. Note that, in this chapter, the velocity and yaw rate are measured at a frequency of 10 Hz, and the ranging measurements and magnetic anomaly measurements are obtained at a frequency of 5 Hz. The noise of the velocity measurement and yaw rate measurement are drawn from a Gaussian distribution with zero mean and standard deviation σ_v for velocity and σ_g for yaw rate. A turn-on bias b_v for velocity and b_g for yaw rate is added separately. Also, the noise of the ranging measurements and the magnetic anomaly measurements are drawn from a Gaussian distribution with zero mean and standard deviation σ_r and σ_m , respectively. The magnetic anomaly map utilized in this study, shown in Figure 4.4.3, is obtained from the United States Geological Survey [18] and contains the magnetic anomaly information at 305 meters altitude from the area around Columbus, Ohio, United States. The left part of the Figure 4.4.3 shows the zoom-in map of the area, which is marked as a red square on the geographic coordinate map. Note that, the unit in the



Figure 4.4.3: The magnetic anomaly map of the area around Columbus, Ohio, United States in geographic coordinate system (right). Left: The zoom-in part of the map.

zoom-in map (left) is converted from degree to meters along the east and north directions, and the left bottom corner of the geographic coordinate map is set as the original point in the zoom-in map.

4.5 Results

4.5.1 Case Studies

The simulation case studies are performed for different group sizes (N = 1, 3, 4, 7, 8, 15, and 16). In each case, multiple Monte Carlo simulations are performed consisting of 200 trials each. The baseline simulation parameters are listed in Table 4.5.1 and Table 4.5.2. Based on the parameter setting of the reference velocity and flight duration, the trajectory length of each UAV is about 180 kilometers in each case.

Since the shape of the group is defined given the inter-vehicle distances, the inter-vehicle dis-

 Table 4.5.1:
 The baseline noise set in case studies.

σ_r	σ_m	σ_v	b_v	σ_g	b_g
1 meter	10 nT	0.3 m/s	$\sim N(0, 0.1\sigma_v)$	0.005 deg/s	$\sim N(0, 0.1\sigma_g)$

 Table 4.5.2:
 The baseline parameter set in case studies.

Each UAV's reference velocity (sine function of time)			Number of particles	Eilight dynation		
Amplitude	Baseline	Frequency	Phase	inumber of particles	ringin duration	
10 m/s	50 m/s	0.05 S	rand*2*pi	10,000	1 hour	

tance errors are used to analyze the quality of the group shape estimated by the cooperative ranging localization algorithm. Meanwhile, due to the limitation of the pairwise communication, the distances between some pairs of UAVs in the group are not measured by the ranging sensor at any point in time. The estimated distances are calculated based on the pose estimates from the cooperative ranging localization algorithm. The average distance error for each simulation is computed for measured pairs and not measured pairs separately. Table 4.5.3 shows the means and standard deviations of these distance error for 200 Monte Carlo simulations.

Table 4.5.3: The means and standard deviations of the average distance error for 200 Monte Carlo simulations.

Group size -	Estimation	with not measured edges	Estimation with measured edges		
	Mean (m)	Standard deviation (m)	Mean (m)	Standard deviation (m)	
3	N/A	N/A	0.6109	0.0061	
4	N/A	N/A	0.5715	0.0050	
7	0.8585	0.1112	0.6074	0.0034	
8	0.7296	0.0299	0.5898	0.0030	
15	1.0828	0.2373	0.6035	0.0022	
16	0.8902	0.0692	0.5953	0.0021	

In Table 4.5.3, most of the means of average distance errors are less than 1 m. The standard derivation of the ranging measurement noise is also set as 1 m in this case study. In other words,

the cooperative ranging localization algorithm is able to maintain an accurate estimate of the intervehicle distances for both the measured and not measured pairs using pairwise communication. This implies that the geometric structure (i.e., shape) of the group is maintained even in the case of missing range measurements. Interestingly, the distance errors of odd numbered groups are always larger than the nearest even numbered groups errors. This is due to the fact that the total number of measured pairs in odd numbered groups is less than in the nearest even numbered groups as discussed in Section 4.3.1, even though the nearest even number is larger than the odd number. Note that, the distance errors are growing as the group size increases, which implies that the presented method is not scalable to a very large UAV group.

For evaluating the performance on global localization, the global position estimate errors of one UAV are reported since the algorithms running on each UAV are identical. In other words, the average position error of one UAV in the group is computed for each simulation. The CDF and statistic data of average position error of one UAV in the group for 200 Monte Carlo simulations for different group sizes are shown in Figure 4.5.1 and Table 4.5.4 separately. In Figure 4.5.1, the dots present the empirical CDF, the continues curves are approximated based on the dots. The following CDF figures follow the same format as Figure 4.5.1. The average position error is computed for each Monte Carlo simulation. The sizes of the group are N = 1, 3, 4, 7, 8, 15 and 16. Note that, for the N = 1 case, since there is no relative information inside the UAV group, Eq. 3.16 in the particle filter is replaced by $\mathbf{y}_k = \mathbf{h}_M(\mathbf{p}_k) + \eta_k$.

The statistical data in Table 4.5.4 shows substantial improvements in navigation performance using the presented navigation algorithm as compared to the dead-reckoning performance of each individual UAV. The reason for only comparing the performance of the presented algorithm with



Figure 4.5.1: The CDF of average position error of one UAV in the group for 200 Monte Carlo simulations for different group sizes.

Table 4.5.4: The statistical data of each simulation's average position error of one UAV in the group for 200 Monte Carlo simulations.

Group size -	Mag	netic localization	Dead-reckoning		
	Mean (m)	Standard deviation (m)	Mean (m)	Standard deviation (m)	
1	59.4	35.3			
3	52.3	28.7			
4	47.8	23.6			
7	33.9	17.6	741.3	552.9	
8	30.3	15.2			
15	22.6	11.5			
16	20.2	10.3			

the dead-reckoning performance is that other magnetic anomaly-based cooperative localization algorithms, such as methods presented in [12], cannot work with the same number of UAVs under the same requirements. The results shown in Figure 4.5.1 and Table 4.5.4 present that the means and standard deviations of position error reduce when the group size increases for the simulated cases. This may due to increased spatial coverage of the UAV group and the increased number of
magnetic anomaly measurements used for map matching. Note that, in the simulated cases, the decrements of the position error are not proportional to the group size. The estimation errors may also be affected by the resolution of the magnetic anomaly map, the quality of sensors, and the error of the relative pose estimates.

4.5.2 Sensitivity Analysis

Sensitivity analysis is performed to evaluate the effect of sensor quality as well as the resolution of the magnetic anomaly map on the presented algorithms. Simulations are first performed with a UAV group with N = 8 and the baseline simulation parameters shown in Table 4.5.1 and Table 4.5.2. In order to evaluate the impact of sensor noise on the algorithms presented in this chapter, the simulations are performed while varying the velocity noise, yaw rate noise, and magnetic anomaly measurement noise separately. Similar as the case studies, multiple Monte Carlo simulations consisting of 200 trials each are performed in each case. Figures 4.5.2, 4.5.3, and 4.5.4 present the CDF of each simulation's average position error for one UAV in the group with varied velocity noise, yaw rate noise, and magnetic anomaly measurement noise, respectively. The blue lines are the baseline case in each figure. The sensitivity analysis results show that the developed cooperative navigation method can tolerate large variations of these sensor noises, with the exception of the cases with $\sigma_v = 3 m/s$ and $\sigma_q = 0.5 deg/s$.



Figure 4.5.2: The CDF of each simulation's average position error for one UAV in the group for 200 Monte Carlo simulations with varied velocity noise.



Figure 4.5.3: The CDF of each simulation's average position error for one UAV in the group for 200 Monte Carlo simulations with varied yaw rate noise.



Figure 4.5.4: The CDF of each simulation's average position error for one UAV in the group for 200 Monte Carlo simulations with varied magnetic anomaly measurement noise.

In order to evaluate the algorithms in different resolution maps, another magnetic anomaly map is used in the simulations, which covers the same area as the map presented in Figure 4.4.3 but from a higher altitude, 3050 meters. As discussed in Section 3.1, when the altitude increase, the resolution of magnetic anomaly map decreases. In this case, the lower altitude map is called the high resolution map, and the higher altitude map is called the low resolution map. The algorithms are simulated for different group sizes (N = 1, 4, 8, and 16) with different resolution maps. In each case, multiple Monte Carlo simulations are performed consisting of 200 trials each. Other parameters are set to be the same as ones shown in Table 4.5.1 and Table 4.5.2. Figure 4.5.5 shows the boxplots of each simulation's average position error for one UAV in the group for 200 Monte Carlo simulations with different group sizes using different resolution maps. In each box, the five lines, from bottom to top, show the minimum, first quartile, median, third quartile, and maximum of the data separately.



Figure 4.5.5: The boxplots of each simulation's average position error for one UAV in the group with different group sizes using different resolution maps.

4.6 Discussion

In this chapter, a cooperative navigation algorithm was presented to estimate a group of UAVs' global poses using inter-vehicle ranging and the Earth's magnetic anomaly measurements. The approach contains two sequential steps: cooperative ranging localization for relative navigation, formulated as an EKF, and cooperative magnetic localization utilizing a particle filter to estimate each UAV's global pose. Furthermore, the presented algorithm is designed to perform using the pairwise

communication, which is compatible with the current communication and ranging technology.

The simulation results show that the developed cooperative ranging localization algorithm is able to provide reasonable relative pose estimates. Utilizing the cooperative magnetic anomaly algorithm, each UAV can estimate its global pose with a high accuracy. Compared with a single UAV case, the cooperative UAV group can provide more accurate global pose estimates, as well as more robustness when using a lower resolution map. The results from the sensitive analysis show that the presented algorithm can tolerate large variations of velocity, yaw rate, and magnetic anomaly measurement noises.

There are several limitations to the presented algorithms such as trajectory constraints. Specifically, the presented algorithm is not scalable to a very large UAV group, which will be the subject of the following chapter. Also, in order to analyze the robustness of the presented algorithms for different cooperative localization applications, various types of application settings will be applied for evaluation in the following chapter.

5

A scalable Framework for Map Matching Based Cooperative Localization

5.1 Introduction

The content of this chapter has been reproduced based on the author's previous work [130]. The scalar field-based cooperative localization algorithms can be classified into two main approaches. The first one is to treat the multi-agent group as a unity and to match observations from all agents with the given map to estimate their poses at each time step, which can be considered centralized methods. The centralized methods are able to achieve error growth bounded positioning and show robustness to issues such as low resolution of the map [15]. However, due to constraints on communication and the on-board computing resources for the agents, the group size of the centralized cooperative localization is limited in practice. The second approach is to perform decentralized cooperative localization, which means that each agent in the group estimates its own pose based on scalar field observations independently at each time step. Then, the estimates are updated, using the relative information (such as ranging, bearing) between this agent and its neighbors $\begin{bmatrix} 6_1 \end{bmatrix}$. Usually, the communication constraints (e.g., range, connectivity, bandwidth) are considered when designing the decentralized localization approaches. In theory, decentralized methods are scalable to the group size and robust to errors made by, or failures of, individual agents. However, in the existing approaches [66, 61], each individual agent needs to come up with a pose estimation, using its own scalar field measurement first, which has potential robustness issues in information poor regions. Thus, to develop a scalable cooperative localization algorithm that is robust to map matching errors while respecting practical communication constraints is the focus of this work.

In this chapter, a scalable framework is presented to perform cooperative localization based on scalar field information, which is performed through fusing the solutions estimated by smaller local subgroups in a large group. The general concept of the proposed approach is illustrated in Figure 5.1.1. The overall group is first organized into several subgroups. A subgroup is defined with one

agent as the fusion center and a limited number of agents that the fusion center can communicate with as members, constrained by either the communication range or the number of available communication channels. In this case, the number of subgroups is the same as the number of agents in the group, and each agent belongs to several different subgroups at the same time. The inter-agent ranging measurements are assumed to be available within the subgroups. A locally centralized cooperative localization method is performed for each subgroup at the fusion center agent to estimate all members' global pose and error covariance. Since each agent in the group belongs to multiple subgroups (as the fusion center or a member) at the same time, it can receive multiple global pose estimates and the corresponding covariances from fusion centers of these subgroups through the communication links. Eventually, each agent would gain an improved global pose estimate through applying a CI method to fuse these redundant estimates, using information provided by other subgroups.

The contributions of this chapter are summarized as follows. First, the proposed algorithm can be scaled to large group sizes under communication constraints (e.g., a group of 128 agents was simulated) with a limitation that the cooperative localization performance is a function of the subgroup size instead of the full group size. Second, the simulation results demonstrate that the proposed algorithm can provide good localization performance for two different types of scalar fields based applications (i.e., magnetic anomaly matching for aerial vehicles and terrain matching for underwater vehicles). Third, compared with the works in Chapters 3 and 4 [15, 63], the proposed algorithm is shown to have improved performance with a similar computation cost for each agent in the group. Finally, the source code of the proposed algorithm and simulator is shared online to allow the readers to more easily verify and build on this work. The code is available online: https://github.com/wvu-irl/Scalable-Framework-Cooperative-Localization.git. The rest of this chapter is organized in the following manner. The problem statement and notations are introduced in Section 5.2. Section 5.3 explains the proposed algorithm design in detail. The simulation utilized to evaluate the proposed algorithm is introduced in Section 5.4, with results discussed in Section 5.5. The chapter concludes in Section 5.6.



Figure 5.1.1: Illustration of the proposed framework for map matching based cooperative localization. (**Left**) A large group of agents is divided into subgroups based on communication constraints where one subgroup is created for each agent. (**Upper right**) The geometry of the subgroups (i.e., the relative positions inside the subgroups) are estimated, using range-only measurements, then the geometry is used to extract measurements of the scalar field to estimate the pose and associated uncertainty of each member in the subgroup. (**Lower right**) An agent, as an example, receives multiple copies of its pose estimate through its membership in several subgroups and fuses them to reduce pose error.

5.2 **Problem Statement and Notations**

In this study, the case is considered where a large group of N agents (e.g., space, aerial, ground, surface, or underwater vehicles) is entering an environment without GNSS. At the moment entering the GNSS-denied environment, the initial pose in the global frame for each agent is assumed to be known with a small uncertainty (e.g., provided by GNSS). The main objective is to achieve accurate and robust global localization for all agents in a large group mainly based on local communication and inter-agent ranging measurements along with scalar field information. In this chapter, a one-based numbering system is used for indexing. Let the group of N agents be denoted by $\Phi = {\varphi_1, \varphi_2, \dots, \varphi_N}$ where φ_i denotes the *i*th agent in the group. The pose of each agent is partially observable; thus, at each time step, the pose is represented as a Gaussian belief $\varphi_i[\mathbf{b}_k] = {\varphi_i[\hat{\mathbf{x}}_k], \varphi_i[\mathbf{P}_k]}$ where $\varphi_i[\hat{\mathbf{x}}_k]$ and $\varphi_i[\mathbf{P}_k]$ are the mean vector and covariance matrix of the state of the *i*th agent at the *k*th time step.

Each agent is assumed to be equipped with radios that enable communication with nearby agents to exchange information, as well as to perform undirected inter-agent ranging. Due to communication limitations (e.g., range or number of channels), each agent can only communicate and perform ranging with a limited number of agents inside a certain range. Therefore, the agents are divided into subgroups based on the communication constraints as follows:

$$\Phi_i = \{\varphi_j \in \Phi \mid \zeta(\varphi_i, \varphi_j) < \epsilon\}$$
(5.1)

where $\zeta(\varphi_i, \varphi_j)$ is a user defined function indicating if agent *i* is capable of communicating with agent *j* and ϵ is a user defined threshold. For example, $\zeta(\varphi_i, \varphi_j)$ may be based on signal strength, and ϵ may define a threshold determining whether the signal strength is adequate for agent *i* to communicate with agent j.

Based on this definition, the group has at least N subgroups, each with M members, where subgroup Φ_i corresponds to the agents capable of communicating with agent φ_i . The agent i is named the fusion center agent of the subgroup i. Furthermore, the subgroups are not disjointed (i.e., $\Phi \neq \bigcup_{i=1}^{N} \Phi_i$), as each agent is included in multiple subgroups. In other words, the situations of isolated agents and isolated subgroups are not considered in this study. The beliefs of subgroup i at time step k are denoted by the following:

$$\Phi_i[\mathbf{b}_k] = \{\varphi_j[\mathbf{b}_k] \mid \varphi_j \in \Phi_i\}$$
(5.2)

where $\Phi_i[\hat{\mathbf{x}}_k]$ and $\Phi_i[\mathbf{P}_k]$ are similarly the set of means and covariance matrices of subgroup i at time step k. Each agent in a subgroup measures the distance between other agents in the subgroup. Let the distance between agent i and agent j be denoted by $d[\varphi_i, \varphi_j]$, and let the set of distance measurements between agents in a subgroup be denoted by $\Phi_i[d]$.

Each agent is assumed to be loaded with a prior scalar field map covering the operation area. Since the types of scalar fields evaluated in this study change very little in a short time frame [4, 16], the prior loaded scalar field map is assumed to be deterministic. Each agent also performs realtime measurements of the scalar field at the current location with sensor noises, such as gravity anomaly, magnetic anomaly, or altimeter measurements, at each time step, which can be exchanged through the communication links. In general, scalar fields may vary with altitude depending on the applications (e.g., magnetic anomaly) and the available data at different altitudes, which may not be available or dense enough; however, this study only deals with the navigation problem in 2D (i.e., assuming the agents are moving at a constant altitude). The scalar field measurements of the *i*th agent are denoted by $\varphi_i[I]$, and the sets of these measurements in a subgroup are denoted by $\Phi_i[I].$

Meanwhile, each agent is assumed to be able to measure or estimate its velocity at each time step, which could be achieved through utilizing the Doppler velocity log [131], wheel odometry [107], or vehicle dynamic model [126], along with yaw rate measurements, using gyroscopes. Similar to before, the velocity and yaw rate measurements of the *i*th agent are denoted by $\varphi_i[v]$ and $\varphi_i[\omega]$, and the sets of measurements in a subgroup are denoted by $\Phi_i[v]$ and $\Phi_i[\omega]$.

5.3 System Design

5.3.1 System Overview

The proposed algorithm is developed for estimating each agent's global pose in a scalable group with communication constraints. The pipeline of the proposed multi-subgroup based algorithm is shown in Figure 5.3.1.



Figure 5.3.1: The pipeline of the proposed algorithm. The details of the subgroup localization process are shown in Figure 5.3.2.

Each agent in the group is assigned to different subgroups based on the communication range or the number of channels, and these subgroups overlap, as discussed earlier in Section 5.2. Note that in this study, in order to conveniently evaluate the performance of the proposed algorithm for each agent in the group, the subgroups are pre-set at the beginning, which means that the memberships of each subgroup are assumed to be constant during an operation. Without pre-setting the subgroups, each agent in the group cannot be guaranteed to belong to the same subgroups during the operation, which could lead the performance of each agent to be unequal and difficult to evaluate. Each subgroup takes each agent's velocity, yaw rate, scalar field measurements, as well as the ranging measurements between the agents in the subgroup as inputs. The outputs of each subgroup are pose estimates of the agents belonging to this subgroup. As mentioned earlier, each agent could belong to multiple subgroups. Thus, multiple pose and covariance estimates from different subgroups exist for each agent in the group at each time step. For example, each subgroup contains Magents and each agent is a member of L subgroups. Therefore, L estimates of the pose and covariance can be communicated to that agent. For each agent, the data fusion algorithm, explained in Section 5.3.3, is developed for estimating the agent's pose based on this redundant information.



Figure 5.3.2: The pipeline of the cooperative localization process in a subgroup *i*. $\Phi_i[r_k]$ means the relative poses between each pair of agents in the subgroup *i* at time step *k*.

5.3.2 Subgroup Cooperative Localization

The cooperative localization algorithm that was developed in Chapter 3 is applied to obtain the estimated poses and covariances of all agents in a subgroup at each time step. For completeness, the main approach are explained in this subsection. The pipeline of the subgroup process is described in Figure 5.3.2. Specifically, two sequential algorithms, named cooperative ranging localization and cooperative scalar field localization, are included in this process and performed in the fusion center agent in each subgroup. The cooperative ranging localization step estimates the relative poses between each pair of agents in the subgroup. The cooperative scalar field localization step then estimates the pose and covariances of the fusion center agent in the subgroup through matching the multiple scalar field measurements with relative pose constraint to the prior scalar field map. Since the processes in the rest of this subsection are identical for each subgroup, in order to present equations clearly, all notations listed in the rest of this subsection do not include the index of the subgroup.

The cooperative ranging localization algorithm is formulated as an EKF as presented in Section 3.3.2 under Chapter 3.

With the estimated relative poses in the subgroup, the scalar field measurements from each agent can be treated as points on a fix geometric shape to match to the map. The cooperative scalar field localization method is formulated as a particle filter as introduced in Section 3.3.3 under Chapter 3. Since the particle filter is performed in the fusion center agent, for each particle, the states, shown in Eq. 3.13, at the next time step are predicted, using the fusion center agent's velocity and yaw rate measurements and the state estimate at the current time step.

In the subgroup, the other agents' predicted global positions can be extracted by adding the updated relative positions from Eq. 3.17 to the predicted states from the particle filter. The vector-

valued observation function h_M is used to extract the predicted scalar field measurements from the given map based on each agent's predicted position.

5.3.3 Data Fusion

The goal of the data fusion step is to estimate an improved pose for each agent. Since the correlation among the pose and covariance estimates is unknown due to information reuse between subgroups, the CI algorithm [9] is applied to fuse the estimates of each agent from multiple subgroups. In general, the CI algorithm fuses information based on a convex combination of the information matrices (i.e., the information states and the inverse of corresponding covariance matrices). Therefore, to apply CI, each agent must have knowledge of the expectations and associated covariance matrices estimated in the subgroups it belongs to. The procedure for obtaining these values is described in detailed in this section.

Covariance Estimates From Particles

The expectation of the state at time k for the fusion center agent in each subgroup, denoted as $\hat{\mathbf{p}}_{k}$, can be obtained from the particle filter by computing the weighted average of the particles given the particle weights. Let ${}^{l}\mathbf{p}(i)$ denote *i*th state of the *l*th particle, and let $\hat{\mathbf{p}}(i)$ denote the expected value of the *i*th state. Now, the covariance matrix of the state denoted by *C* is calculated from the particles as follows:

$$C(l,p) = \sum_{i=1}^{n} \{ i \tilde{w} \left(i \mathbf{p}(l) - \hat{\mathbf{p}}(l) \right) \left(i \mathbf{p}(p) - \hat{\mathbf{p}}(p) \right) \}$$
(5.3)

where ${}^{i}\tilde{w}$ is the weight of the *i*th particle, *n* is the total number of particles, and *l* and *p* represent the indices of the elements in the matrix. Then, the covariance matrix of the pose estimate of each agent at time step *k* is given by the following:

$$\varphi_i[\mathbf{P}_k](l,p) = C_k(l,p) : (l,p=1,2,3)$$
(5.4)

where $\varphi_i[\mathbf{P}_k](l, p)$ denotes the element in the *l*th row and *p*th column of the covariance matrix of *i*th agent. The pose estimates of other agents in the subgroup are able to be derived by utilizing the fusion center agent's pose estimate, the relative pose from Eq. 3.17, and the subgroup rotation error γ . According to [132], the covariance matrix of the pose estimate for each agent in the subgroup at one time step is the same as the covariance of the fusion center agent.

Covariance Intersection

In order to present the details of CI applied in this problem clearly, $(\varphi_i[\hat{x}_k^j], \varphi_i[P_k^j])$ denotes the *j*th estimate of pose and covariance of *i*th agent at time step *k*. The implementation of CI in this situation is presented as follows:

$$\varphi_i[\hat{\boldsymbol{x}}_k] = \varphi_i[\boldsymbol{P}_k] \sum_{j=1}^{L} (\alpha_j \varphi_i[\boldsymbol{P}_k^j]^{-1} \varphi_i[\hat{\boldsymbol{x}}_k^j])$$
(5.5)

$$\varphi_i[\mathbf{P}_k] = [\sum_{j=1}^{L} (\alpha_j \varphi_i[\mathbf{P}_k^j]^{-1})]^{-1}, \sum_{j=1}^{L} \alpha_j = 1, \alpha_j \ge 0$$
(5.6)

where $\alpha = [\alpha_1, \alpha_2, \cdots]$ are the weighting coefficients, which need to be solved, and L is the total number of solutions of the *i*th agent (i.e., the total number of subgroups, including agent *i*). Based

on [9], α is assumed to be computed by solving a nonlinear optimization problem, which is to find α to minimize the determinant or trace of $\varphi_i[P_k]$. However, to solve the nonlinear optimization problem is computationally expensive, especially, when the *i*th agent has a large number of solutions. To reduce the computation cost, the fast CI method presented in [133], which is designed to solve the weighting coefficients quickly, is applied as follows:

$$\alpha_j = \frac{\det(\boldsymbol{S}) - \det(\boldsymbol{S} - \varphi_i[\boldsymbol{P}_k^j]) + \det(\varphi_i[\boldsymbol{P}_k^j])}{L\det(\boldsymbol{S}) + \sum_{q=1}^{L} [\det(\varphi_i[\boldsymbol{P}_k^q]) - \det(\boldsymbol{S} - \varphi_i[\boldsymbol{P}_k^q])]}$$
(5.7)

with

$$\boldsymbol{S} = \sum_{q=1}^{L} \varphi_i [\boldsymbol{P}_k^q].$$
(5.8)

By solving for α from Eq. (5.7), the improved pose of the agent at time step k (i.e., $\varphi_i[\hat{x}_k]$) can be computed through Eqs. (5.5) and (5.6).

5.4 Simulation

In order to verify that the proposed algorithm is suitable for different types of scalar fields, a magnetic anomaly map and a bathymetric map are utilized in the simulation study.

For the ease of simulating a large number of agents, the agents are simulated with a bicycle model [129] in the 2D plane. A feedback controller steers an agent to follow a reference trajectory and reference velocity based on the pose estimates from the proposed algorithm. All reference trajectories are set to be parallel to latitude lines in a geographic coordinate system. New parallel reference trajectories are added with a certain distance when the group size increases. Agents are assumed to be moving from west to east with the same initial longitude and small uncertainties (a Gaussian

white noise with zero mean and 1 m standard deviation) on the initial poses as discussed in Section 5.2. The reference trajectories are close to the ground truths shown in Figure 3.5.1. In order to simulate situations when all agents are moving at different speeds, the reference velocity for each agent varies slowly, following a sine function with a random offset throughout the duration of the mission.

In this simulation, the communication among agents is restricted by the limited number of communication channels available for each agent. The fusion center agent and agents it can reach are assigned as a subgroup at the beginning of the simulation. Meanwhile, different scalar fields are related to different experimental environments and platforms. For example, magnetic anomaly maps can be used by the localization system of aerial vehicles on Earth [4], or space probes exploring other planets, and bathymetric maps are used for underwater vehicles' localization [61]. Therefore, different parameter settings are introduced corresponding to two map types in the following.

5.4.1 Magnetic Anomaly Map

The Earth's magnetic anomaly information that presents the variations of the Earth's magnetic field is stable and distinctive at different locations in a certain range of altitude [4]. The magnetic anomaly map utilized in this simulator, shown in Figure 5.4.1, is obtained from the United States Geological Survey [18] and contains the magnetic anomaly information at 305 m altitude from the area around Columbus, Ohio, United States.

The noise of the velocity measurement and yaw rate measurement are drawn from a Gaussian distribution with zero mean and standard deviation $\sigma_v = 0.3$ m/s for velocity and $\sigma_g = 0.005$ deg/s for the yaw rate. A turn-on bias $b_v \sim N(0, 0.1\sigma_v)$ for velocity and $b_g \sim N(0, 0.1\sigma_g)$ for



Figure 5.4.1: The magnetic anomaly map of the area around Columbus, Ohio, United States in geographic coordinate system at 305 m altitude.

yaw rate is added separately. According to the references [25, 4], the noise of the ranging measurements and the magnetic anomaly measurements are drawn from a Gaussian distribution with zero mean and standard deviation $\sigma_r = 1 \text{ m}$ and $\sigma_m = 10 \text{ nT}$, respectively. The initial distance between each pair of neighbor agents in latitude is set to 1000 m. The reference velocity for each agent varies from 40 to 60 m/s. The mission duration is 1 h, and the trajectory length of each agent is about 180 km.

5.4.2 Bathymetric Map

The bathymetry, also known as submarine topography, presents the depths of the underwater terrain. The bathymetric map applied in this simulator, shown in Figure 5.4.2, is acquired from the



Figure 5.4.2: The bathymetric map of the area around the Santa Barbara Channel, United States, in geographic coordinate system.

United States Geological Survey [19].

According to the descriptions of parameter setting in underwater experiments in [61, 62], the standard deviation of the velocity measurement noise and yaw rate measurement noise are selected as $\sigma_v = 0.1 \text{ m/s}$ and $\sigma_g = 0.1 \text{ deg/s}$. A turn-on bias $b_v \sim N(0, 0.1\sigma_v)$ for velocity and $b_g \sim N(0, 0.1\sigma_g)$ for the yaw rate is also added separately. The noise of the altimeter measurements is drawn from a Gaussian distribution with zero mean and standard deviation $\sigma_a = 1 \text{ m}$. The noise of the inter-agent ranging measurements are drawn from a Gaussian distribution with zero mean and standard deviation with zero mean and standard deviation $\sigma_r = 1 \text{ m}$. The initial distance between each pair of neighboring agents in latitude is set to be 200 m. The reference velocity for each agent varies from 0.5 to 1.5 m/s. The mission duration is 1 h and the trajectory length of each agent is about 3600 m.

5.5 Results

The proposed algorithm is designed to remove the constraints of the group size in previous works in performing cooperative localization [15, 63]. Therefore, the algorithms presented in Chapter 3 are utilized as the comparison to evaluate the proposed algorithm. The performance of the proposed algorithm is compared with simulations under the full communication (FC) assumption, where all agents can communicate with each other agent at each time step. The performances and sensitivity analysis of FC could be found in Chapter 3. The average position RMSE in each simulation from both FC and the proposed algorithm are evaluated. In each case, multiple Monte Carlo simulations are performed, consisting of 160 trials each.

The proposed algorithm and FC are first evaluated, using a magnetic anomaly map-based simulation environment. Figure 5.5.1 shows the CDF of each simulation's average position error for all agents in the group at each time step from the Monte Carlo simulations. The FC is performed with different group sizes (i.e., N = 4, 8, 16). Meanwhile, the proposed algorithm is also evaluated with various group sizes (i.e., N = 8, 16, 32) and subgroup sizes (i.e., the number of agents in each subgroup), such as M = 4, 8, 16, as shown in Figure 5.5.1. For example, N = 32, M = 16 means that the full group has 32 agents, and each subgroup has 16 agents. Table 5.5.1 shows the statistical data of the Figure 5.5.1 along with performance of the simulation with dead-reckoning (DR) without using the ranging and magnetic anomaly information. From Figure 5.5.1 and Table 5.5.1,

Table 5.5.1: Mean of Monte Carlo simulations' position RMSE in each situation, using magnetic anomaly map (unit: meters).

N = 4	N = 8	N = 8	N = 16	N = 16	N = 32	ער
FC	M = 4	FC	M = 8	FC	M = 16	DK
58.4	35.7	25.3	12.2	12.9	5.2	751.3



Figure 5.5.1: The CDF of each simulation's average position error for all agents in the group for 160 Monte Carlo simulations with different algorithms and group sizes. These simulations are run with the magnetic anomaly map shown in Figure 5.4.1.

it is clear that the performance of both the proposed algorithm and FC improves as the group size increases, and both are much better than doing DR alone. Note that since the number of particles used in different scenarios is the same at the cooperative scalar field localization step, the performance comparison is suitable to be made between FC and the proposed algorithm, when the group size of FC is equal to the subgroup size of the proposed algorithm. In this case, with the same number of particles, the particle filter in both situations deals with the same number of scalar field measurements at each time step. Meanwhile, the computation of each agent applying the proposed algorithm is similar to the computation of each agent working in FC because of the fast CI method [133] applied. The performance of the proposal algorithm shows improvement over the FC in Figure 5.5.1 and Table 5.5.1.

Similar results are acquired using a bathymetric map-based simulation environment. Figure 5.5.2 shows the CDF of each simulation's average position error for all agents in the group at each

time step. Due to the smaller bathymetric map, a maximum of up to 16 in group sizes were simulated. Table 5.5.2 shows the statistical data of the Figure 5.5.2 along with performance of the DR.



Figure 5.5.2: The CDF of each simulation's average position error for all agents in the group for 160 Monte Carlo simulations with different algorithms and group sizes. These simulations are run with the bathymetric map shown in Figure 5.4.2.

Table 5.5.2: Mean of the position RMSE in each situation using bathymetric map (unit: meters).

N = 4	N = 8	N = 8	N = 16	תת
FC	M = 4	FC	M = 8	DK
29.6	14.7	19.6	8.2	293.1

The performance of the proposed algorithm applied with different group sizes and the same subgroup size using the magnetic anomaly map is shown in Table 5.5.3. It shows that the proposed algorithm works with large group sizes (e.g., N = 128). The computation of each agent in the

Table 5.5.3: Mean of Monte Carlo simulations' position RMSE, using magnetic anomaly map with different group size and same subgroup size (unit: meters).

N = 16 (M = 8)	N = 32 (M = 8)	N = 128 (M = 8)
12.2	13.5	15.3

group, which is only spent on locally centralized cooperative localization and data fusion, does not increase when the group size grows. However, since each agent uses the same amount of information in different sizes of groups (i.e., the sizes of the subgroups are the same), the performance of these simulations are similar.

5.6 Discussion

In this chapter, a scalable cooperative localization framework based on scalar field prior maps and real-time measurements is presented. In order to satisfy the communication constraints, a large agent group is separated into several subgroups, where each agent is treated as the fusion center in each subgroup. A locally centralized cooperative localization is performed to estimate the agents' poses in each subgroup through matching multiple scalar field measurements constrained by relative positions to the given map. In order to avoid over-convergence due to using correlated information, a fast CI algorithm is applied to estimate an improved pose for each agent based on its multiple pose and covariance estimates from its membership in multiple subgroups.

The simulation results show that the proposed algorithm is able to deal with large groups (e.g., N = 128), and to achieve higher performance, even under more restrictive communication constraints, compared to the previous works [15, 63]. Additionally, this approach is suitable for missions with different types of scalar fields.

There are several limitations in this work that need to be addressed in the future, such as extend-

ing the algorithm to agents distributed in 3D spaces and better integration of the uncertainty in the group geometry estimates into the map-matching process. A great challenge is to find decentralized localization solutions that can utilize all agents' measurements in the group in an efficient and robust manner, under communication constraints. Currently, the performance of the proposed algorithm is limited by the subgroup size, instead of the full group size. The future work will focus on allowing information to flow beyond the immediate neighbors while maintaining the stability of the pose estimation algorithm in the agent network. Meanwhile, the idea that using geometry-based multiple independent measurements to solve matching problem leads a solution for lidar-based SLAM, which will be introduced in the following chapter.

Lidar-inertial Based Localization

6.1 Introduction

Lidar-based localization in environments including moving objects, such as self-driving vehicles passing through the street with crowded people, autonomous driving robotic guides working in museums at a busy time, is a challenging problem [134]. The points detected from moving objects could affect the accuracy of transformation estimation between two point clouds using matching related methods, such as ICP [5], NDT [28]. Meanwhile, because of the benefits of the lidar sensors, such as accurate ranging detection, working in poorly illuminated environments, the lidar sensors cannot always be replaced. Therefore, to design a robust lidar-based localization system for environments including moving objects is a necessary problem.

The current solutions for estimating lidar-based odometry, such as Lidar Odometry and Mapping (LOAM) [40], Lightweight and ground optimized lidar odometry and mapping (Lego-LOAM) [41], are not designed to work in dynamic environments. With the assistant from inertial sensors (e.g., IMUs), the lidar-inertial-based odometry solutions, such as Fast Lidar-Inertial Odometry (Fast-LIO) [135], Lidar Inertial Odometry via Smoothing and Mapping (LIO-SAM) [136], show more robustness compared with the lidar-based odometry in high-speed moving scenarios. These methods, however, are assumed to be designed for environments with only static objects. Semantic information extracted based on deep learning technologies, such as PV-RCNN [114], RangeNet++ [48], is applied to filter out the points from dynamic objects [47]. However, due to the lack of different types of labeled data (currently, the available labeled data is mainly focused on cars, pedestrians, and cyclist [116]), the objects in point clouds could not be classified in many different types. Meanwhile, the semantic information could not represent the dynamic objects in some cases, such as a car which is parking off the street cannot be treated as a dynamic objects. Several moving object segmentation algorithms are presented, such as Lidar-MOS [45], removert [43]. However, these segmentation algorithms required pose estimates as input, and were only perform offline, which cannot be used for online odometry estimation.

A potential solution for lidar-based localization algorithm for environments including dynamic objects was proposed based on independent feature tracking, which is inspired from the scalable framework presented in the previous chapter. The idea of the potential solution is demonstrated through using an example shown in Figure 6.1.1.

In Figure 6.1.1, a point cloud and measurements from other sensors are inputted into the designed system. In this example, five features extracted from the point cloud are presented as five squares with different colors. These features are assumed to be tracked in the previous point cloud in this example. Therefore, five estimates about the pose of the system are able to be predicted with respect to each feature. The dots and eclipses in the figure present the estimated pose and covariances separately, and different colors are correspondent with different features. Note that, the dot and eclipse in black color present the estimated pose and covariances based on measurements from other sensors. In the figure, it is easy to notice that the estimated pose related to feature 5 is far away from other estimates. In this case, feature 5 may be extracted from a moving object or the pose is estimated with wrong information (such as, the wrong pair of correspondent features between current time step and previous time step). As a result, the estimated pose related to feature 5 is then removed as an outlier. The cluster of pose estimates and covariances which is close to the pose and covariance estimated based on prediction using measurements from other sensors is kept. Eventually, the estimated pose and covariance are computed through fusing all remained pose estimates and covariances. However, the independent feature tracking based lidar odometry algorithm is assumed working with accurate feature detection and tracking, which are not easy problems to solve without using learning-based algorithms.



Figure 6.1.1: One example of the independent feature tracking based lidar odometry algorithm. Colored squares are the features extracted from one point cloud. The colored dots and ellipses represent the estimated position and covariance using the feature with the same color. The dot and eclipse in black color present the estimated pose and covariances based on measurements from additional sensors. The red dot and ellipse show the fused position and covariance.

In this chapter, a lidar-inertial based SLAM framework with a dynamic object removal filter is presented to solve 3D pose (i.e., orientation and position) estimation problem in environments including a large number of moving objects. The SLAM framework is designed to have a front-end, which is to perform odometry estimate using a lidar and an inertial sensor, and a back-end, which is to do optimization for the global map. The front-end of the SLAM framework is built as an IEKF [135], which preforms tightly-coupled lidar-inertial odometry estimation. In the IEKF, the states are updated through matching the current scan to a local map, which is generated using estimated odometry and input point clouds. In order to reduce the influences of the noisy points from dynamic objects, a dynamic object removal filter is designed to filter out most of points from dynamic objects inside the input point cloud before adding to the local map. Meanwhile, in order to provide the capability of mapping large areas and to regulate the error growth (i.e., accumulated error from the front-end odometry), a pose-graph optimization strategy is applied using iSAM2 [137]. Moreover, the Scan Context method presented in [13] is utilized for loop closure detection.

In this chapter, several collected data sets (using wheeled robot as the platform and using handheld devices) have been applied to evaluate the performance of the proposed algorithm. The results show the robustness of the proposed algorithm with dynamic object removal filter compared with the case without dynamic object removal filter. The performance of the proposed algorithm is evaluated with indoor and outdoor data sets, and is compared with other state-of-the-art lidar-inertial based SLAM algorithms. The benefit of the pose-graph optimization strategy is verified through comparing the localization error using odometry only with using both odometry and pose-graph optimization. In order to evaluate that the proposed algorithm is versatile (i.e., the proposed algorithm is suitable for different applications, such as wheeled ground robot or handheld devices, with different types of lidars), different types of data sets collected in different environments are applied to solve 3D pose estimate in this chapter.

The rest of this chapter is organized in the following manner. The details of the presented algorithm are described in Section 6.3. The experiments utilized to evaluate the presented algorithm are introduced in Section 6.4. In Section 6.5, the performance of the proposed algorithm is analyzed based on results from the experiments. The chapter is concluded in Section 6.6.

6.2 **Problem Statement**

The objective of this chapter is to solve the 3D lidar-inertial-based localization problem in environments including a large number of moving objects. The required sensors applied in this chapter contain a mechanical lidar, such as Velodyne[®] HDL-32 channel lidar or Ouster[®] 64 channel lidar, and a six-axis IMU (i.e., the measurements from the IMU only include three-axis acceleration and three-axis angular velocity). During the operation, the transformation between the lidar and the IMU is rigid, which means there is no relative moving between the lidar and the IMU. However, the calibration matrix between the lidar and the IMU could be unknown, which is able to be estimated in the proposed algorithm.

In order to state the problem more clearly, three coordinate systems are used in this chapter. The IMU frame is denoted as I; the lidar frame is denoted as L; and the world frame (i.e., the global frame) is denoted as W. Note that, in this chapter, the world frame is defined as the IMU frame at the first time step (i.e., the start time). The extrinsic from lidar to IMU is defined as T_{IL} , and the pose of the IMU in world frame is defined as T_{WI} , where T is the transformation matrix.

Note that, since the update rate for the IMU usually is faster than the update rate of the lidar, and the extrinsic T_{IL} would be estimated at the same time, the problem in this chapter is defined to achieve online estimation for T_{WI} only using sequential lidar point clouds and measurements from a six-axis IMU. Meanwhile, the lidar-inertial-based SLAM framework is designed to work in environments including a large number of moving objects.

6.3 Methods

6.3.1 System Overview

In this chapter, observations and measurements from lidar and an IMU are fused to solve the problem of localization in environments including a large number of dynamic objects. A SLAM framework containing lidar-inertial odometry as the front-end and a pose-graph optimization as the back-end is designed in this chapter. The framework of the designed system is shown in Figure 6.3.1. The inputs of this system only require the point clouds from a lidar, which could be in



Figure 6.3.1: The overview framework of the proposed system.

different models (e.g., Velodyne 32 channel lidar, Ouster 64 channel lidar), and the acceleration and angular rate measured from an inertial sensor (e.g., IMU). As introduced before, the framework contains two main components, which are the lidar-inertial odometry and the pose-graph optimization as shown in Figure 6.3.1.

The lidar-inertial odometry is formulated as an IEKF [135], which uses measurements from the IMU to do forward propagation, and applies the scan-to-map matching method using point clouds

from the lidar to do observation update. A local map, which presents the point clouds in a userdefined area around the lidar, is generated and maintained in the lidar-inertial odometry algorithm. In order to reduce the influence of the noisy points from the dynamic objects, a dynamic object removal filter is designed to filter out these noisy points before adding the current scan (i.e., the point cloud collected by the lidar after one full shot or swipe) to the local map. An ikd-Tree data structure presented in [138] is adopted to store points in the local map to improve the computational efficiency of the presented algorithm. In order to save memory and computation resources for the pose-graph optimization, a keyframe strategy is applied in the framework. A keyframe scan is defined as the current scan being beyond the user-defined distance from the previous keyframe scan. Note that, the first scan is set as a keyframe scan. Therefore, the outputs of the lidar-inertial odometry module are the odometry pose and keyframe scan (if available at the current time step). The details of the lidar-inertial odometry module are explained in Section 6.3.2.

The iSAM₂ algorithm presented in [1₃₇] is applied in the pose-graph optimization to bound the error growth from the odometry poses. The pose-graph optimization algorithm takes each keyframe scan as the vertex and the odometry pose estimates as the edges among these vertices. A loop-closure algorithm, named scan context [1₃], is applied to detect the loop-closure (i.e., the visited places). The details of the pose-graph optimization are introduced in Section 6.3.3.

6.3.2 Lidar-inertial odometry

The lidar-inertial odometry is designed to provide odometry estimates and keyframes for posegraph optimization. The framework of the lidar-inertial odometry algorithm is presented in Figure 6.3.2.

In order to achieve online and accurate pose estimation, the problem is formulated as an IEKF,



Figure 6.3.2: Lidar-inertial odometry algorithm design.

which uses inertial measurements (i.e., three-axis acceleration and three-axis angular velocity) for forwarding propagation and applying a scan-to-map matching algorithm to compute the residual for state update. The IEKF is able to iteratively update the state using observations, which is beneficial for reducing the errors caused by linearization. Meanwhile, a light computation method presented in [135] is applied to speed up the computing of the Kalman gain K that includes an observation Jacobian Matrix in large size, which would be explained in detail in Section 6.3.2.

The residual inside the IEKF is computed through matching the undistorted point cloud at the current time step with the local map, called scan-to-map matching. Since each point in one scan is not collected at the same time, when the lidar is moving, each raw point cloud collected by the lidar has the in-scan motion error. During the undistortion step, all points in one scan have been projected to the scan end-time based on the estimated relative pose from the forward propagation and the sampling time of each point in the scan. In order to improve the accuracy of the point cloud matching, a point-to-plane matching algorithm presented in [88] is applied to compute the residual. The details of the residual computation will be introduced in Section 6.3.2.

The local map used for scan-to-map matching contains points around the lidar in the world frame. Since each point in the downsampled income point cloud has to find five nearest neighbor points in the local map for computing the residual, which will be explained in Section 6.3.2, in order to improve the efficiency of point searching, the size of the local map is limited in user-defined area around the lidar. In this case, the local map can be treated as a patch of the global map with downsampled points. Meanwhile, an effective data structure, named ikd-Tree presented in [138], is adopted to store points for the local map, which was proved more effective in k-nearest searching and map managing (i.e., insertion and delete) [138].

As the system is designed for working in environments including a large number of moving objects, before adding the undistorted scan to the local map, a distance-based dynamic object removal filter is proposed to filter out most of the points from dynamic object removal filter, which would be explained in detail in Section 6.3.2.

The odometry poses estimated using the IEKF are published at the IMU update rate. Meanwhile, the undistorted scan which is beyond the user-defined distance from the previous keyframe scan is also published as a new keyframe. Note that, the distance between the current undistorted scan and the previous keyframe is estimated using the estimations from the lidar-inertial odometry.

State Estimation

The goal of the lidar-inertial-based odometry algorithm is to estimate the pose of the IMU in the world frame and to estimate the extrinsic between lidar and IMU. The state vector, defined in Eq. 6.1, contains fifteen states of the IMU, the gravity vector, and the rotation matrix and translation

from lidar to IMU.

$$\mathbf{x} = [\mathbf{R}_{WI}, \mathbf{p}_{WI}, \mathbf{v}_{WI}, \mathbf{b}_{\omega}, \mathbf{b}_{a}, \mathbf{g}_{W}, \mathbf{R}_{IL}, \mathbf{p}_{IL}]^{T}$$
(6.1)

where, \mathbf{R}_{WI} and \mathbf{p}_{WI} are the attitude and position of the IMU in the world frame, \mathbf{v}_{WI} is the velocity vector of the IMU in the world frame, \mathbf{b}_{ω} and \mathbf{b}_{a} are the biases of the gyroscope and the accelerator, \mathbf{g}_{W} is the gravity vector in the world frame, \mathbf{R}_{IL} and \mathbf{p}_{IL} are the calibration transformations from lidar to IMU. Note that rotation R is defined as SO(3). The state transition model using the inertial measurements as inputs could be given by

$$\mathbf{x}_{i+1} = \mathbf{x}_i \boxplus \left(\Delta t f(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i) \right)$$
(6.2)

where the encapsulation operator \boxplus establish a bijective mapping from a local neighborhood to its tangent space, defined in [135], \mathbf{x}_i is the state vector at time step i, and Δt is the time between continuing time steps. The \mathbf{u}_i is the inputs at time step i, defined as:

$$\mathbf{u} = [\omega_m, a_m]^T \tag{6.3}$$

where, ω_m and a_m are the angular velocity and acceleration measurements from IMU. The \mathbf{w}_i is the noise at time step i, defined as:

$$\mathbf{w} = [\mathbf{n}_{\omega}, \mathbf{n}_{a}, \mathbf{n}_{b\omega}, \mathbf{n}_{ba}]^{T}$$
(6.4)

where \mathbf{n}_{ω} and \mathbf{n}_{a} are the white noise of IMU measurements, $\mathbf{n}_{b\omega}$ and \mathbf{n}_{ba} are the Gaussian noises of the IMU biases, which are modeled as the random walk. The transition function f is modelled

$$f(\mathbf{x}, \mathbf{u}, \mathbf{w}) = \begin{pmatrix} \omega_m - b_\omega - \mathbf{n}_\omega \\ \mathbf{v}_{WI} \\ \mathbf{R}_{WI}(a_m - b_a - \mathbf{n}_a) + \mathbf{g}_W \\ \mathbf{n}_{b\omega} \\ \mathbf{n}_{ba} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \end{pmatrix}$$
(6.5)

Residual Computation

Each measured point, denoted as p_{Lj} (*j*th point in one scan in lidar frame), from lidar is considered to include noise, denoted as n_{Lj} . Since the point cloud data from lidar is sparse, instead of finding a correspondence point in the local map using a point in the current scan, finding a small patch of correspondence plane is more reliable. A correspondence plane, which is modeled using the five nearest points in the local map, means the point in the current scan should be in the same plane after applying the transformation. This could be explained using Eq. 6.6

$$0 = u_{Wj}(T_{WIk}T_{IL}(p_{Lj} + n_{Lj}) - q_{Wj})$$
(6.6)

where, u_{Wj} is the normal vector of the corresponding plane for point j in world frame, T_{WIk} is the estimated pose of the IMU in world frame at time step k, T_{IL} is the estimated calibration matrix from the lidar to the IMU, q_{Wj} is one corresponding point in world frame for point j, which is used to model the corresponding plane, and the equation $(p_{Lj} + n_{Lj})$ means the jth point in lidar

as:
frame without noise. As shown in Eq. 6.6, without noise, the product between a vector inside a plane and the normal of the plane should be zero. However, the noise caused by estimation and measurement always exists. Therefore, during the point cloud matching, for j th point in the current scan, the residual is defined as,

$$\mathbf{z}_j = u_{Wj} (T_{WIk} T_{IL} p_{Lj} - q_{Wj}) \tag{6.7}$$

In this case, the size of the residual matrix would be $m \times 1$, where m is the number of the points used for point cloud matching in the current scan. Meanwhile, the Jacobian of the observation matrix **H**, derived based on the residual matrix, should be in size $m \times 24$ (24 is the number of states introduced in Eq. 6.1). Since the number of points used for point cloud matching is often huge, the size of **H** is large, which is not effective for calculate Kalman gain using Eq. 6.8 [7].

$$\mathbf{K} = \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1}$$
(6.8)

where, **P** is the estimated covariance with respect to the state vector, **R** presents measurement noises. To invert the matrix \mathbf{HPH}^T is computation expensive when the size of the matrix is huge. A new form of Kalman gain proved in [135] is adopted in this method, as shown below:

$$\mathbf{K} = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \mathbf{P}^{-1})^{-1} \mathbf{H}^T \mathbf{R}^{-1}.$$
(6.9)

Since **R** is a diagonal matrix, the Eq. 6.9 only requires to invert two matrices both with the dimension of the state. In this case, the computation cost in IEKF could be reduced when a large number of points is used for matching. The rest of the IEKF process is the same as the equations presented in [139].

Dynamic Object Removal Filter

The proposed algorithm is designed to work in environments including a large number of moving objects. With the influences from the moving objects, the transformation estimated based on point cloud matching may not present the accurate movement of the sensors. A dynamic object removal filter is designed to remove most of the points from moving objects in a scan before adding them to the local map. Since the observation update is performed by matching the current scan to the local map, the effect of the points from moving objects is reduced after removing these points from the local map.

The dynamic object removal filter is designed to check the moving objects by comparing the distance between the transformed point in the current scan in the world frame and its nearest point in the local map, shown as Eq. 6.10.

$$\mathbf{D}_{min} \le ||T_{WIk}T_{IL}p_L - p_W|| \le \mathbf{D}_{max} \tag{6.10}$$

where, $T_{WIk}T_{IL}p_L$ presents the transformed point in the current scan using estimated transformation from IEKF, p_W is the nearest point in the local map after transformed, \mathbf{D}_{min} and \mathbf{D}_{max} are the user-defined parameters. Equation 6.10 shows that a point would be considered as the point from moving objects when the distance between the transformed point and its nearest point in the local map is in the user-defined distance.

With the assistant of the transformation estimated from the IEKF, which is reliable in a short time frame, the point in the current scan should be very close to its nearest point after performing transformation if the point reflects a static object. In this case, the lower boundary of the distance \mathbf{D}_{min} should be set larger than the estimated error from the IEKF in a short time frame. Meanwhile, the upper boundary of the distance \mathbf{D}_{max} is set based on the average velocity of the moving objects in the environment.

6.3.3 Loop Closure Detection and Pose-Graph Optimization

For robust localization in long-term drives, the error growth is always bounded through applying pose-graph optimization with assistant from a loop closure detection algorithm.

The loop closure detection problem, also known as place recognition, is important in long-term navigation. In this chapter, the scan context algorithm presented in [13] is adopted to detect loop closure. Other than learning-based detection algorithms, the scan context algorithm only uses information from a sensor. Meanwhile, the height value of points stored in a row-column designed grid map guarantee the accuracy of the place recognition, which is important for pose-graph optimization.

The pose-graph optimization framework is refereed from iSAM₂ presented in [137]. The iSAM₂ formulates the factor graph problem as a Bayes tree, which allows easily updating the variance with respect to each vertex. Meanwhile, when the graph grows large, only part of the Bayes tree, which is close to the additional edge, needs to be updated using iSAM₂. In this case, iSAM₂ is more computation effectively when performing pose-graph optimization. In this chapter, the Levenberg-Marquardt algorithm [140] is applied to perform non-linear optimization.

6.4 Experiments

Several data sets, collected using different sets of sensors in different real environments, are applied to evaluate the performance of the presented algorithm. Specifically, one part of the data sets was collected using a wheeled ground robot mounted with a Velodyne® HDL-32E 32-channel 3D lidar and a Novatel® INS system. The other part of the data sets were collected using a handheld device which included an Ouster® 64-channel 3D lidar and a six-axis IMU. These data were collected in indoor and outdoor environments. In outdoor environments, the GNSS data collected by the Novatel® SPAN GNSS/INS system is processed as the ground truth of the positioning. The details of the experiments are introduced as follows.

6.4.1 Data Sets Collected Using A Robot

Platform

A wheeled ground robot, named "BrambleBee", is utilized to collect data in both indoor and outdoor environments. The BrambleBee, introduced in [141, 142, 143], is designed to perform autonomous pollination for bramble planes (e.g., raspberry). As shown in Figure 6.4.1, the Bramble-Bee is built primarily on a ClearPath Robotics [®] Husky platform, which enable the robot to drive in both indoor and outdoor environments. The BrambleBee platform is equipped with a Velodyne[®] HDL-32E 32-channel 3D lidar, which is able to collect point clouds data at about 10 Hz. A Novatel[®] SPAN GNSS/INS system is also installed onboard. The INS system is able to provide six-axis inertial measurements (i.e., three-axis accelerations and angular velocities) at 50 Hz.

The positions solved using carrier-phase DGPS [144, 145] based on measurements from a base station, shown in Figure 6.4.2, and the onborad GNSS system are considered as the ground truth



Figure 6.4.1: The BrambleBee robotic platform in the West Virginia University (WVU) greenhouse.

in this experiments.

During data collections, the BrambleBee was remotely controlled by an operator to perform driving forward, backward, and rotation with speed at about an average of 1 meter per second and the angular rate at about an average of 20 degrees per second. Note that, the arm of the BrambleBee was removed during the data collections for easy operation.

Environments

The data set collected using BrambleBee covers some places on WVU Evansdale campus in both indoor and outdoor environments. The data set "esb" is collected by driving BrambleBee from



Figure 6.4.2: The GNSS base station receiver.

the Mineral Resources Building to the ground level of the Engineering Research Building through passing the ground level of the Engineering Sciences Building (i.e., the indoor environments). The data set "campus" is collected by driving the BrambleBee from the Advance Engineering Research Building to the Engineering station through the outdoor environments. The data set "campus2" is collected by driving the BrambleBee from the Mineral Resources Building to the entrance of the Crossing building. The data set "rec-center" is collected by driving the BrambleBee from the ground level of the Crossing building to the Rec Center, and then driving back to the ground level of the Crossing building. Note that, both "campus2" and "rec-center" contain indoor and outdoor environments with one drive. Due to the COVID-19, only part of the data set "esb" is collected in the environment that includes a large number of the moving objects (e.g., humans in this data set). Therefore, the data set "esb" would be used for evaluating the robustness of the proposed algorithm. Other data sets are utilized to evaluate the localization and mapping performance of the proposed algorithm in different environments.

6.4.2 Data sets collected using a handheld lidar

In order to evaluate the ability of the proposed algorithm for different types of lidars in different applications, a data set was collected with a handheld Ouster[®] 64 channel lidar in a stone mine is applied in the experiments. Compared with the data collected using a ground robot, the handheld data contains more rotation motion in roll and pitch, which is close to drone driving. The inertial measurements are collected with the IMU mounted in the Ouster[®] lidar.

6.5 Results

In this chapter, a lidar-inertial-based SLAM algorithm with a dynamic object removal filter is designed for solving localization problems in environments including a large number of moving objects. The robustness of the proposed algorithm is evaluated with the data set "esb" in this chapter. Meanwhile, the performance of the proposed algorithm is compared with the state-of-the-art lidarinertial SLAM algorithm LIO-SAM using data set "esb". To choose the LIO-SAM algorithm as the comparison for three reasons. The first one is that current dynamic object aware based 3D SLAM algorithms, such as SUMA++ [47, 146], are performed dynamic detection using Deep learning methods, which requires enough data for training. In this case, the learning based algorithms cannot be used for the comparison due to the limited number of available labeled data in the experimental environments. The second one is that available non-learning related dynamic detection based SLAM algorithms, such as the method presented in [42], are performed localization for 2D solutions, which cannot be applied to solve 3D pose estimation problem directly. The third reason is that even though the LIO-SAM algorithm is not developed for environments including a large number of moving objects, the LIO-SAM algorithm is able to work in some moving objects included environments due to using measurements from both lidar and IMU.

In order to reduce the influences of the points from moving objects, a dynamic object removal filter is designed to remove most of the points from moving objects before adding them to the local map. Figure 6.5.1 shows one example of the local map with and without the dynamic object removal filter. These local maps shown in Figure 6.5.1 are generated using data set "esb" at the time when two people are walking on the left of the BramebleBee. The Figure 6.5.1 shows that the designed dynamic object removal filter is able to remove most of points from moving objects before adding the current scan to the local map. In this case, the thresholds in the dynamic object removal filter \mathbf{D}_{min} and \mathbf{D}_{max} are set to 0.08 meter and 0.2 meters, respectively. A sensitivity analysis about these thresholds, \mathbf{D}_{min} and \mathbf{D}_{max} , are performed. The result from the sensitivity analysis shows that the dynamic object removal filter has a similar performance when setting $\mathbf{D}_{min} \ge 0.05$ meter and $\mathbf{D}_{max} \le 0.28$ meter in this experiment. Note that, since the update frequency of the lidar is set to 10 Hz in this experiment, the range of the thresholds is close to the speed of pedestrians.

The global maps generated using the data set "esb" with the proposed algorithm with and without dynamic object removal filter are presented in Figure 6.5.2. Since the data set is collected in indoor environment, there is no ground truth for evaluate the accurate performances of the proposed algorithm. The most different place (marked as red circles) in these two generated maps



(a) With dynamic object removal filter.



(b) Without dynamic object removal filter.

Figure 6.5.1: One example of the local map with and without the dynamic object removal filter. The axis shows the pose of the IMU (positive x(red) is forward, positive y(green) is left, and positive z(blue) is up).

shows that the proposed algorithm with dynamic object removal filter is more accurate than the algorithm without dynamic object removal filter in this case through comparing these generated map with the build map. Meanwhile, one scan of the raw point clouds in data set "esb" around the time step when the proposed algorithm without dynamic object removal filter has troubles is shown in Figure 6.5.3. Figure 6.5.3 shows that there are about nine moving objects (i.e., humans) walking in the hall way. In this case, the environment could be considered as including a large number of moving objects. Figure 6.5.4 shows that the LIO-SAM [136] also fails with data set "esb". The Figs. 6.5.2, 6.5.4, and 6.5.3 demonstrate that the proposed algorithm with dynamic object removal filter is more robust than other algorithms in environments including a large number of moving objects.

In order to present the performance of the proposed algorithm, the trajectories estimated using the proposed algorithm with back-end optimization and without back-end optimization are compared with the ground truth using three data sets collected around Advance Engineering Research



(a) With dynamic object removal filter.



(b) Without dynamic object removal filter.

Figure 6.5.2: One example of global map with and without dynamic object removal filter in indoor dynamic environment. The most different place between these two figures are pointed out with red circles.



Figure 6.5.3: One scan of the raw point clouds in data set "esb" around the time step when the algorithm without dynamic object removal filter has troubles. There are about nine moving objects (i.e., humans) in this scan.



Figure 6.5.4: The LIO-SAM algorithm [136] failed with data set "esb" at the place with a large number of moving objects existed.

Building (AERB). The experimental environments with ground truth estimated using the DGPS algorithm are shown in Figures 6.5.5, 6.5.6, and 6.5.7.

The comparisons of the trajectories estimated using the proposed algorithm with back-end optimization and without back-end optimization with the ground truths in three experimental environments are shown in Figures 6.5.8, 6.5.9, and 6.5.10. Meanwhile, the statistic data (RMSE in East-axis and North-axis) corresponded with these trajectories are listed in Tables 6.5.1, 6.5.2, and 6.5.3.

Note that, the trajectories estimated using the proposed algorithm, shown in Figures 6.5.8, 6.5.9, and 6.5.10, are rotated using the guessed initial heading compared to the North direction in GPS solutions. Meanwhile, the wrong position estimation in GPS solution shown in Figure 6.5.8 was removed before calculating the RMSE. The information from the Figures 6.5.8, 6.5.9, and 6.5.10, and the Tables 6.5.1, 6.5.2, and 6.5.3 shows that the proposed algorithm is able to achieve same level positioning accuracy compared with the DGPS algorithm. Besides that, the proposed algorithm is able to correct the error from the DGPS algorithm caused by signal issues. Due to the scan-to-map strategy (i.e., each income scan is matched to the local map to estimate the pose) applied in the front-end of the proposed algorithm, the performances between with and without the back-end optimization did not show large differences with data sets collected in environments around the AERB and the parking lot since the size of the local map is able to cover most of the area of the experimental environments. However, with the data set collected in environments around the grass-



Figure 6.5.5: Environments around AERB and the trajectory estimated using the DGPS algorithm. (Note that, one position is estimated in wrong due to signal issue of the Global Positioning System (GPS).)



Figure 6.5.6: Environments around the parking lot and the trajectory estimated using the DGPS algorithm.

Table 6.5.1: RMSE in East-axis and North-axis (Data set: Environments around AERB).

	East (meters)	North (meters)
Odom	0.0724	0.0670
Optimized	0.0694	0.0683



Figure 6.5.7: Environments around the grassland and the trajectory estimated using the DGPS algorithm.



Figure 6.5.8: Trajectories estimated using the proposed algorithm with back-end (Optimized) and without back-end optimization (Odom) and ground truth (GPS) (Data set: Environments around AERB).



Figure 6.5.9: Trajectories estimated using the proposed algorithm with back-end (Optimized) and without back-end optimization (Odom) and ground truth (GPS) (Data set: Environments around the parking lot).

Table 6.5.2: RMSE in East-axis and North-axis (Data set: Environments around the parking lot).

	East (meters)	North (meters)
Odom	0.0082	0.0023
Optimized	0.0084	0.0030

Table 6.5.3: RMSE in East-axis and North-axis (Data set: Environments around the grass-land).

	East (meters)	North (meters)
Odom	0.2888	0.1941
Optimized	0.2597	0.1789

land, which can be considered as featureless environments, the performance with the back-end optimization shows better than the performance without the back-end optimization. The reason is that a featureless environment (e.g., the grassland area for the lidar) brings troubles for point cloud matching, and due to the inaccurate estimation, the error in the local map would be accumulated. Since the end position of this data set is collected in a feature-rich place (i.e., several buildings close to the position), the loop-closure detection and factor-graph-based optimization are able to reduce the accumulated errors that happened in the grassland area.

The point cloud map generated with the handheld data set is shown in Figure 6.5.11. With more motion in roll and pitch when collecting data, Figure 6.5.11 shows that the proposed algorithm is suitable for different types of sensors in more aggressive movement.

Figures 6.5.12, 6.5.13, and 6.5.14 present the global maps generated using data set "campus", "campus2", "rec-center", respectively. These global maps demonstrate that the proposed algorithm is able to perform robust and accurate 3D localization in both indoor and outdoor environments.

6.6 Discussion

In this chapter, a lidar-inertial based SLAM framework with a dynamic object removal filter is presented to solve the 3D localization problem in environments including a large number of moving objects.

The presented framework has been evaluated with several data sets collected in real environments with different types of platforms. The results show that the proposed dynamic object removal filter is able to effectively remove most of the points from moving objects. The proposed



Figure 6.5.10: Trajectories estimated using the proposed algorithm with back-end (Optimized) and without back-end optimization (Odom) and ground truth (GPS) (Data set: Environments around the grassland).



Figure 6.5.11: Global map generated with data set collected using handheld devices in a stone mine.



Figure 6.5.12: Global map generated using data set "campus".



Figure 6.5.13: Global map generated using data set "campus2".



Figure 6.5.14: Global map generated using data set "rec-center".

SLAM framework shows more robustness in environments including a large number of moving objects compared with the state-of-the-art lidar-inertial SLAM algorithm. Moreover, the proposed algorithm shows the ability about working with different types of applications in both indoor and outdoor environments. Meanwhile, the results demonstrate that the pose-graph-based optimization assisted with the scan context loop closure detection algorithm is able to regular the error growth from the lidar-inertial odometry.

There are several limitations to the proposed algorithm that need to be addressed in the future. A more effective method to set up the threshold in the proposed dynamic object removal filter needs to be designed. So far, the data sets which include a large number of moving objects are collected in limited situations and environments. More data sets utilized for evaluating the robustness of the lidar-based SLAM algorithms need to be collected in the future.

Conclusions and Future Work

7.1 Conclusions

In this dissertation, the problems about localization in GNSS-denied and challenging environments (i.e., environments with only low-resolution features and environments including a large number of moving objects) are addressed. To achieve robust and accurate localization, several solutions based on sensor fusion methods are proposed with respect to different applications in these two different types of challenging environments.

For environments with only low-resolution features, a scalable cooperative localization framework mainly using inter-agent ranging and scalar field measurements with communication limitations is presented. Compared with a single-agent case, the proposed cooperative localization framework shows more robustness in localization with a low-resolution scalar field map. In order to satisfy the communication constraints (i.e., each agent can only communicate a limited number of other agents in the group) and to allow scalability, a large agent group is separated into several subgroups, where each agent is treated as the fusion center in each subgroup. A locally centralized cooperative localization algorithm is developed to estimate the agents' poses in each subgroup more effectively compared with other scalar field-based centralized cooperative localization algorithms [12]. The locally centralized cooperative localization algorithm contains two sequential steps: cooperative ranging localization for relative navigation, formulated as an EKF, and cooperative scalar filed localization utilizing a particle filter to estimate each agent's global pose inside the subgroup. In order to avoid over-convergence due to using correlated information, a fast CI algorithm is applied to estimate an improved pose for each agent based on its multiple pose and covariance estimates from its membership in multiple subgroups.

The proposed framework has been evaluated with a feedback control-based simulator using two different types of scalar field information (i.e., magnetic anomaly measurements and bathymet-

ric measurements) in two different applications (i.e., UAV navigation and underwater navigation). The results from multiple Monte Carlo simulations demonstrate that the proposed algorithm is able to deal with large groups (e.g., 128 agents in a group) under restrictive communication constraints. The proposed locally centralized cooperative localization algorithm has been shown to provide more accurate and robust global pose estimates compared with single-agent cases with communication constraints. Meanwhile, the results from the sensitivity analysis show that the presented algorithm can tolerate large variations of velocity, yaw rate, and scalar field measurement noises.

For environments including a large number of moving objects, a lidar-inertial based SLAM framework with a dynamic object removal filter is presented to solve the 3D pose (i.e., orientation and position) estimation problem. In order to reduce the influences about the points from moving objects, a dynamic object removal filter is designed through checking distance changes to filter out the points from moving objects. A lidar-inertial odometry algorithm is built on an IEKF, which uses inertial measurements (i.e., three-axis acceleration and angular rate) as prediction inputs and point cloud-based scan-to-map matching results as observation updates. To achieve the online process, the data structure, ikd-Tree [138], is utilized to manage points in the map. In order to satisfy the requirement for a long-time navigation, a pose-graph-based optimization algorithm is adopted to regular the error incremented from the lidar-inertial based odometry estimation. The scan context algorithm is applied to detect loop closure for pose-graph optimization.

The presented lidar-inertial-based SLAM framework has been evaluated with several data sets collected in real environments with different types of platforms. The results show that the proposed dynamic object removal filter is able to effectively detect and remove most of the points from moving objects before adding the current scan to the map. With this benefit, the proposed SLAM algorithm shows more robustness in environments including a large number of moving ob-

jects (e.g, about nine moving objects shown in the data set) compared with the state-of-the-art lidar-inertial SLAM algorithms. Moreover, the proposed algorithm shows the ability about working with different types of applications in both indoor and outdoor environments by evaluating with different types of data sets. Meanwhile, the results demonstrate that the pose-graph-based optimization assisted with the scan context loop closure detection algorithm is able to regular the error growth from the lidar-inertial odometry.

Another potential solution for solving localization in environments including a large number of moving objects is proposed and discussed in this dissertation. Different from the current methods, which perform matching algorithms with the full scan, the proposed algorithm treats each small selected cluster of points as an independent feature. The pose of the lidar is estimated through tracking each independent feature. Meanwhile, since the tracking process for each feature is independent, the features detected from the moving objects could be removed by applying an outlier detection algorithm. The pose of the lidar at each time step could be estimated by fusing all estimates based on each independent feature after outlier removal. The evaluation with a 2D lidar in a simulated environment shows the potential of the proposed algorithm to work robustly in dynamic environments if a reliable feature detection algorithm is available.

7.2 Future Work

There are several limitations about these proposed algorithms in this dissertation that need to be addressed in the future.

For the scalable cooperative localization framework working for the environments with only low-resolution features, the proposed algorithm needs to be extended to agents distributed in 3D spaces. Meanwhile, better integration of the uncertainty in the group geometry estimates (i.e, relative pose estimates inside the group) into the map-matching process needs to be developed. A great challenge is to find decentralized localization solutions that can utilize all agents' measurements in the group in an efficient and robust manner, under communication constraints. Currently, the performance of the proposed algorithm is limited by the subgroup size, instead of the full group size. For this issue, an algorithm that allows information to flow beyond the immediate neighbors while maintaining the stability of the pose estimation needs to be developed in the future. The model of the agents in the simulator, which is developed to evaluate the proposed algorithm, is considered as the bicycle model. A More complex model will be designed and applied in the simulator in the future.

For the lidar-inertial SLAM algorithm working in environments including a large number of moving objects, a more effective method to set up the threshold in the proposed dynamic object removal filter needs to be designed. So far, the data sets which include a large number of moving objects are collected in limited situations and environments (i.e., only collected from Campus environments and a stone mine). More data sets utilized for evaluating the robustness of the lidar-based SLAM algorithms need to be collected in the future. Meanwhile, a sensitivity analysis about the number of dynamic objects need to performed with more data sets in the future.

References

- [1] Scott Gleason, Demoz Gebre-Egziabher, and Demoz Gebre Egziabher. Gnss applications and methods. 2009.
- [2] James J Spilker Jr, Penina Axelrad, Bradford W Parkinson, and Per Enge. Global positioning system: theory and applications, volume I. American Institute of Aeronautics and Astronautics, 1996.
- [3] Franc Dimc, Matej Bažec, Daniele Borio, Ciro Gioia, Gianmarco Baldini, and Marco Basso. An experimental evaluation of low-cost gnss jamming sensors. *Navigation: Journal of The Institute of Navigation*, 64(1):93–109, 2017.
- [4] Aaron Canciani and John Raquet. Absolute positioning using the earth's magnetic anomaly field. Navigation: Journal of The Institute of Navigation, 63(2):111–126, 2016.
- [5] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In Sensor fusion IV: control paradigms and data structures, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.
- [6] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.
- [7] Mohinder S Grewal and Angus P Andrews. *Kalman filtering: Theory and Practice with MAT-LAB*. John Wiley & Sons, 2014.
- [8] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In Signal processing, sensor fusion, and target recognition VI, volume 3068, pages 182–193. International Society for Optics and Photonics, 1997.
- [9] Simon J Julier and Jeffrey K Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the 1997 American Control Conference (Cat. No.* 97CH36041), volume 4, pages 2369–2373. IEEE, 1997.
- [10] Fredrik Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7):53–82, 2010.

- [11] Giorgio Grisetti, Rainer Kümmerle, and Kai Ni. Robust optimization of factor graphs by using condensed measurements. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 581–588. IEEE, 2012.
- [12] A Canciani and K Brink. Improved magnetic anomaly navigation accuracy through cooperative navigation. In *Proceedings of the ION 2017 Pacific PNT Meeting*, pages 239–262, 2017.
- [13] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4802–4809. IEEE, 2018.
- [14] Bo Wang, Li Yu, Zhihong Deng, and Mengyin Fu. A particle filter-based matching algorithm with gravity sample vector for underwater gravity aided navigation. *IEEE/ASME Transactions on Mechatronics*, 21(3):1399–1408, 2016.
- [15] Chizhao Yang, Jared Strader, Yu Gu, Alexander Hypes, Aaron Canciani, and Kevin Brink. Cooperative uav navigation using inter-vehicle ranging and magnetic anomaly measurements. In 2018 AIAA Guidance, Navigation, and Control Conference, page 1595, 2018.
- [16] Francisco Curado Teixeira, João Quintas, Pramod Maurya, and António Pascoal. Robust particle filter formulations with application to terrain-aided navigation. *International Journal* of Adaptive Control and Signal Processing, 31(4):608–651, 2017.
- [17] Lino Marques, Urbano Nunes, and Andal T de Almeida. Olfaction-based mobile robot navigation. *Thin solid films*, 418(1):51–58, 2002.
- [18] United States Geological Survey. Magnetic anomaly maps and data for north america, 2014.
- [19] United States Geological Survey, 2007.
- [20] Stephen M Rock, Brett Hobson, and Sarah E Houts. Return-to-site of an auv using terrainrelative navigation: Field trials. In 2014 IEEE/OES Autonomous Underwater Vehicles (AUV), pages 1–8. IEEE, 2014.
- [21] Yurong Han, Bo Wang, Zhihong Deng, and Mengyin Fu. A combined matching algorithm for underwater gravity-aided navigation. *IEEE/ASME Transactions on Mechatronics*, 23(1):233-241, 2017.
- [22] Dieter Fox, Wolfram Burgard, Hannes Kruppa, and Sebastian Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous robots*, 8(3):325–344, 2000.

- [23] Ryo Kurazume, Shigemi Nagata, and Shigeo Hirose. Cooperative positioning with multiple robots. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, pages 1250–1257. IEEE, 1994.
- [24] Agostino Martinelli, Frederic Pont, and Roland Siegwart. Multi-robot localization using relative observations. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 2797–2802. IEEE, 2005.
- [25] I Coddington, William C Swann, Ljerka Nenadovic, and Nathan R Newbury. Rapid and precise absolute distance measurements at long range. *Nature photonics*, 3(6):351, 2009.
- [26] Ivan Alexander Getting. Perspective/navigation-the global positioning system. IEEE spectrum, 30(12):36-38, 1993.
- [27] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In 2014 IEEE international conference on robotics and automation (ICRA), pages 15–22. IEEE, 2014.
- [28] Martin Magnusson, Achim Lilienthal, and Tom Duckett. Scan registration for autonomous mining vehicles using 3d-ndt. *Journal of Field Robotics*, 24(10):803–827, 2007.
- [29] Yuhao Chen, Chizhao Yang, Bowen Song, Nicholas Gonzalez, Yu Gu, and Boyi Hu. Effects of autonomous mobile robots on human mental workload and system productivity in smart warehouses: A preliminary study. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 64, pages 1691–1695. SAGE Publications Sage CA: Los Angeles, CA, 2020.
- [30] Yuhao Chen, Chizhao Yang, Yu Gu, and Boyi Hu. Influence of mobile robots on human safety perception and system productivity in whole sale and retail trade environments: A pilot study. *IEEE Transactions on Human-Machine Systems*, 2021.
- [31] Wei Tan, Haomin Liu, Zilong Dong, Guofeng Zhang, and Hujun Bao. Robust monocular slam in dynamic environments. In 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pages 209–218. IEEE, 2013.
- [32] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. Ds-slam: A semantic visual slam towards dynamic environments. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1168–1174. IEEE, 2018.
- [33] Yuxiang Sun, Ming Liu, and Max Q-H Meng. Improving rgb-d slam in dynamic environments: A motion removal approach. *Robotics and Autonomous Systems*, 89:110–122, 2017.

- [34] GQ Huang, AB Rad, and YK Wong. Online slam in dynamic environments. In ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005., pages 262–267. IEEE, 2005.
- [35] Girish Chowdhary, Eric N Johnson, Daniel Magree, Allen Wu, and Andy Shein. Gps-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft. *Journal of Field Robotics*, 30(3):415–438, 2013.
- [36] Davide Scaramuzza, Michael C Achtelik, Lefteris Doitsidis, Fraundorfer Friedrich, Elias Kosmatopoulos, Agostino Martinelli, Markus W Achtelik, Margarita Chli, Savvas Chatzichristofis, Laurent Kneip, et al. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments. *IEEE Robotics* & Automation Magazine, 21(3):26–40, 2014.
- [37] Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011.
- [38] Young-Min Ahn, Daniel J Block, and Ramavarapu S Sreenivas. Autonomous navigation and localization of a quadrotor in an indoor environment. *Journal of Aerospace Information Systems*, 12(12):699–709, 2015.
- [39] Daniel Magree and Eric N Johnson. Factored extended kalman filter for monocular visionaided inertial navigation. *Journal of Aerospace Information Systems*, 13(12):475–490, 2016.
- [40] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, 2014.
- [41] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4758–4765. IEEE, 2018.
- [42] Rendong Wang, Youchun Xu, Miguel Angel Sotelo, Yulin Ma, Thompson Sarkodie-Gyan, Zhixiong Li, and Weihua Li. A robust registration method for autonomous driving pose estimation in urban dynamic environment using lidar. *Electronics*, 8(1):43, 2019.
- [43] Giseop Kim and Ayoung Kim. Remove, then revert: Static point cloud map construction using multiresolution range images. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 10758–10765. IEEE, 2020.

- [44] David Yoon, Tim Tang, and Timothy Barfoot. Mapless online detection of dynamic objects in 3d lidar. In 2019 16th Conference on Computer and Robot Vision (CRV), pages 113–120. IEEE, 2019.
- [45] X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, and C. Stachniss. Moving Object Segmentation in 3D LiDAR Data: A Learning-based Approach Exploiting Sequential Data. *IEEE Robotics and Automation Letters (RA-L)*, 2021.
- [46] Rainer Kümmerle, Michael Ruhnke, Bastian Steder, Cyrill Stachniss, and Wolfram Burgard. A navigation system for robots operating in crowded urban environments. In 2013 IEEE International Conference on Robotics and Automation, pages 3225–3232. IEEE, 2013.
- [47] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguère, Jens Behley, and Cyrill Stachniss. Suma++: Efficient lidar-based semantic slam. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4530–4537. IEEE, 2019.
- [48] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4213–4220. IEEE, 2019.
- [49] Gabriele Ferri, Andrea Munafò, Alessandra Tesei, Paolo Braca, Florian Meyer, Konstantinos Pelekanakis, Roberto Petroccia, João Alves, Christopher Strode, and Kevin LePage. Cooperative robotic networks for underwater surveillance: an overview. *IET Radar, Sonar & Navigation*, 11(12):1740–1761, 2017.
- [50] Garth Herman, Aleksander Milshteyn, Airs Lin, Manuel Garcia, Charles Liu, Darrell Guillaume, Khosrow Rad, and Helen Boussalis. Cooperative semi-autonomous robotic network for search and rescue operations. In *International Conference on Universal Access in Human-Computer Interaction*, pages 636–647. Springer, 2014.
- [51] Jing Yuan, Yalou Huang, Tong Tao, and Fengchi Sun. A cooperative approach for multirobot area exploration. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1390–1395. IEEE, 2010.
- [52] Andy Komendat and Agamemnon Crassidis. Center of gravity estimation of an aircraft solely using standard aircraft measurement sensors. In *AIAA Atmospheric Flight Mechanics Conference*, page 4411, 2012.
- [53] John Springmann, James Cutler, and Hasan Bahcivan. Magnetic sensor calibration and residual dipole characterization for application to nanosatellites. In AIAA/AAS Astrodynamics Specialist Conference, page 7518, 2010.

- [54] Reuben Rohrschneider. Terrain relative navigation using crater identification in surface topography data. In *AIAA Guidance, Navigation, and Control Conference,* page 6490, 2011.
- [55] Victoria A Natalie and Jamey D Jacob. Experimental observations of the boundary layer in varying topography with unmanned aircraft. In AIAA Aviation 2019 Forum, page 3404, 2019.
- [56] MW McElhinny and Phillip L McFadden. *The magnetic field of the earth: paleomagnetism, the core, and the deep mantle,* volume 63. Academic Press, 1998.
- [57] Bartosz Brzozowski, Krzysztof Kaźmierczak, Zdzisław Rochala, Marta Wojda, and Konrad Wojtowicz. A concept of uav indoor navigation system based on magnetic field measurements. In 2016 IEEE Metrology for Aerospace (MetroAeroSpace), pages 636–640. IEEE, 2016.
- [58] Bartosz Brzozowski and Krzysztof Kaźmierczak. Magnetic field mapping as a support for uav indoor navigation system. In 2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace), pages 583–588. IEEE, 2017.
- [59] Asaf Grosz, Michael J Haji-Sheikh, and Subhas C Mukhopadhyay. *High sensitivity magnetometers*. Springer, 2017.
- [60] Peter Hood. History of aeromagnetic surveying in canada. *The Leading Edge*, 26(11):1384–1392, 2007.
- [61] Adam Wiktor and Stephen Rock. Collaborative multi-robot localization in natural terrain. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 4529– 4535. IEEE, 2020.
- [62] Tan Yew Teck, Mandar Chitre, and Franz S Hover. Collaborative bathymetry-based localization of a team of autonomous underwater vehicles. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 2475–2481. IEEE, 2014.
- [63] Chizhao Yang, Jared Strader, Yu Gu, Aaron Canciani, and Kevin Brink. Cooperative navigation using pairwise communication with ranging and magnetic anomaly measurements. *Journal of Aerospace Information Systems*, pages 1–10, 2020.
- [64] Rongxin Cui, Yang Li, and Weisheng Yan. Mutual information-based multi-auv path planning for scalar field sampling using multidimensional rrt. *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 46(7):993–1004, 2015.
- [65] Anusna Chakraborty, Rajnikant Sharma, and Kevin Brink. Cooperative localization for multi-rotor uavs. In AIAA Scitech 2019 Forum, page 0684, 2019.

- [66] Yew Teck Tan, Mandar Chitre, and Franz S Hover. Cooperative bathymetry-based localization using low-cost autonomous underwater vehicles. *Autonomous Robots*, 40(7):1187– 1205, 2016.
- [67] Hubiao Wang, Lin Wu, Hua Chai, Yaofei Xiao, Houtse Hsu, and Yong Wang. Characteristics of marine gravity anomaly reference maps and accuracy analysis of gravity matching-aided navigation. Sensors, 17(8):1851, 2017.
- [68] Jisun Lee, Jay Hyoun Kwon, and Myeongjong Yu. Performance evaluation and requirements assessment for gravity gradient referenced navigation. Sensors, 15(7):16833-16847, 2015.
- [69] Aaron Canciani and John Raquet. Airborne magnetic anomaly navigation. *IEEE Transactions on aerospace and electronic systems*, 53(1):67–80, 2017.
- [70] José Melo and Aníbal Matos. Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean Engineering*, 139:250–264, 2017.
- [71] James S McCabe and Kyle J DeMars. Anonymous feature-based terrain relative navigation. Journal of Guidance, Control, and Dynamics, 43(3):410–421, 2020.
- [72] Jacob Anderson and Geoffrey A Hollinger. Communication planning for cooperative terrain-based underwater localization. Sensors, 21(5):1675, 2021.
- [73] Yuan Sun. Autonomous integrity monitoring for relative navigation of multiple unmanned aerial vehicles. *Remote Sensing*, 13(8):1483, 2021.
- [74] Maurice F Fallon, Georgios Papadopoulos, John J Leonard, and Nicholas M Patrikalakis. Cooperative auv navigation using a single maneuvering surface craft. *The International Journal of Robotics Research*, 29(12):1461–1474, 2010.
- [75] Stergios I Roumeliotis and George A Bekey. Distributed multirobot localization. *IEEE transactions on robotics and automation*, 18(5):781–795, 2002.
- [76] Esha D Nerurkar, Stergios I Roumeliotis, and Agostino Martinelli. Distributed maximum a posteriori estimation for multi-robot cooperative localization. In 2009 IEEE International Conference on Robotics and Automation, pages 1402–1409. IEEE, 2009.
- [77] Gao Rui and Mandar Chitre. Cooperative multi-auv localization using distributed extended information filter. In 2016 IEEE/OES autonomous underwater vehicles (AUV), pages 206– 212. IEEE, 2016.

- [78] Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J Pappas. Decentralized active information acquisition: Theory and application to multi-robot slam. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 4775-4782. IEEE, 2015.
- [79] Brent Schlotfeldt, Dinesh Thakur, Nikolay Atanasov, Vijay Kumar, and George J Pappas. Anytime planning for decentralized multirobot active information gathering. *IEEE Robotics and Automation Letters*, 3(2):1025–1032, 2018.
- [80] John F Quindlen and Jonathan P How. Machine learning for efficient sampling-based algorithms in robust multi-agent planning under uncertainty. In AIAA Guidance, Navigation, and Control Conference, page 1921, 2017.
- [81] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309– 1332, 2016.
- [82] Tan Bui-Thanh, Murali Damodaran, and Karen Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. AIAA journal, 42(8):1505– 1516, 2004.
- [83] Michel A Audette, Frank P Ferrie, and Terry M Peters. An algorithmic overview of surface registration techniques for medical imaging. *Medical image analysis*, 4(3):201–217, 2000.
- [84] Aitor Aldoma, Zoltan-Csaba Marton, Federico Tombari, Walter Wohlkinger, Christian Potthast, Bernhard Zeisl, Radu Bogdan Rusu, Suat Gedikli, and Markus Vincze. Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. *IEEE Robotics & Automation Magazine*, 19(3):80–91, 2012.
- [85] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987.
- [86] Nitin Bhatia et al. Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085*, 2010.
- [87] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. Image and vision computing, 10(3):145–155, 1992.
- [88] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

- [89] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453), volume 3, pages 2743–2748. IEEE, 2003.
- [90] Martin Magnusson. *The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection.* PhD thesis, Örebro universitet, 2009.
- [91] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals* of *Data Science*, 2(2):165–193, 2015.
- [92] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, volume 1, pages 281–297. Oakland, CA, USA, 1967.
- [93] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341, 2009.
- [94] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: an efficient clustering algorithm for large databases. ACM Sigmod record, 27(2):73-84, 1998.
- [95] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5):345–366, 2000.
- [96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226– 231, 1996.
- [97] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.
- [98] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. Science, 344(6191):1492-1496, 2014.
- [99] Rudolf Emil Kalman et al. Contributions to the theory of optimal control. *Bol. soc. mat. mexicana*, 5(2):102–119, 1960.
- [100] Dan Simon. Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons, 2006.
- [101] Jun S Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal* of the American statistical association, 93(443):1032–1044, 1998.

- [102] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.
- [103] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard.
 g 2 0: A general framework for graph optimization. In 2011 IEEE International Conference on Robotics and Automation, pages 3607–3613. IEEE, 2011.
- [104] Jeffrey K Uhlmann. Covariance consistency methods for fault-tolerant distributed data fusion. *Information Fusion*, 4(3):201–215, 2003.
- [105] Erik Nelson. Berkeley localization and mapping (blam).
- [106] Kenji Koide, Jun Miura, and Emanuele Menegatti. A portable three-dimensional lidarbased system for long-term and wide-area people behavior measurement. *International Journal of Advanced Robotic Systems*, 16(2):1729881419841532, 2019.
- [107] Yu Gu, Jared Strader, Nicholas Ohi, Scott Harper, Kyle Lassak, Chizhao Yang, Lisa Kogan, Boyi Hu, Matthew Gramlich, Rahul Kavi, et al. Robot foraging: Autonomous sample return in a large outdoor environment. *IEEE Robotics & Automation Magazine*, 25(3):93–101, 2018.
- [108] Dirk Hahnel, Rudolph Triebel, Wolfram Burgard, and Sebastian Thrun. Map building with mobile robots in dynamic environments. In 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), volume 2, pages 1557–1563. IEEE, 2003.
- [109] Chieh-Chih Wang, Charles Thorpe, Sebastian Thrun, Martial Hebert, and Hugh Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(9):889–916, 2007.
- [110] Huijing Zhao, Masaki Chiba, Ryosuke Shibasaki, Xiaowei Shao, Jinshi Cui, and Hongbin Zha. Slam in a dynamic large outdoor environment using a laser scanner. In 2008 IEEE International Conference on Robotics and Automation, pages 1455–1462. IEEE, 2008.
- [111] Takeo Miyasaka, Yoshihiro Ohama, and Yoshiki Ninomiya. Ego-motion estimation and moving object tracking using multi-layer lidar. In 2009 IEEE intelligent vehicles symposium, pages 151–156. IEEE, 2009.
- [112] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4490–4499, 2018.

- [113] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In Advances in neural information processing systems, pages 820–830, 2018.
- [114] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10529– 10538, 2020.
- [115] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Advances in neural information processing systems, pages 5099-5108, 2017.
- [116] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Jürgen Gall, and Cyrill Stachniss. Towards 3d lidar-based semantic scene understanding of 3d point cloud sequences: The semantickitti dataset. *The International Journal of Robotics Research*, page 02783649211006735, 2021.
- [117] Binghao Li, Thomas Gallagher, Andrew G Dempster, and Chris Rizos. How feasible is the use of magnetic field alone for indoor positioning? In 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pages 1–9. IEEE, 2012.
- [118] Ilari Vallivaara, Janne Haverinen, Anssi Kemppainen, and Juha Röning. Magnetic field-based slam method for solving the localization problem in mobile robot floor-cleaning task. In 2011 15th international conference on advanced robotics (ICAR), pages 198–203. IEEE, 2011.
- [119] Richard J Blakely. *Potential theory in gravity and magnetic applications*. Cambridge university press, 1996.
- [120] Douglas B West. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.
- [121] Dan Simon. Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons, 2006.
- [122] Jared Strader, Yu Gu, Jason N Gross, Matteo De Petrillo, and Jeremy Hardy. Cooperative relative localization for moving uavs with single link range measurements. In 2016 IEEE/ION Position, Location and Navigation Symposium (PLANS), pages 336–343. IEEE, 2016.

- [123] John Clifford Gower. Properties of euclidean and non-euclidean distance matrices. *Linear* Algebra and its Applications, 67:81–97, 1985.
- [124] Ivan Dokmanic, Reza Parhizkar, Juri Ranieri, and Martin Vetterli. Euclidean distance matrices: essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 32(6):12–30, 2015.
- [125] Richard J Blakely. *Potential theory in gravity and magnetic applications*. Cambridge University Press, 1996.
- [126] Mehran Khaghani and Jan Skaloud. Autonomous vehicle dynamic model-based navigation for small uavs. *Navigation: Journal of The Institute of Navigation*, 63(3):345–358, 2016.
- [127] Reuben Strydom, Saul Thurrowgood, and Mandyam V Srinivasan. Visual odometry: autonomous uav navigation using optic flow and stereo. In *Proceedings of Australasian conference on robotics and automation*, 2014.
- [128] Fernando Caballero, Luis Merino, Joaquin Ferruz, and Anibal Ollero. Vision-based odometry and slam for medium and high altitude flying uavs. *Journal of Intelligent and Robotic Systems*, 54(1-3):137–161, 2009.
- [129] Peter Corke. *Robotics, vision and control: fundamental algorithms In MATLAB® second, completely revised,* volume 118. Springer, 2017.
- [130] Chizhao Yang, Jared Strader, and Yu Gu. A scalable framework for map matching based cooperative localization. *Sensors*, 21(19):6400, 2021.
- [131] Jeff Snyder. Doppler velocity log (dvl) navigation for observation-class rovs. In OCEANS 2010 MTS/IEEE SEATTLE, pages 1–9. IEEE, 2010.
- [132] Luis C Carrillo-Arce, Esha D Nerurkar, José L Gordillo, and Stergios I Roumeliotis. Decentralized multi-robot cooperative localization using covariance intersection. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1412–1417. IEEE, 2013.
- [133] Dietrich Franken and Andreas Hupper. Improved fast covariance intersection for distributed data fusion. In 2005 7th International Conference on Information Fusion, volume 1, pages 7-pp. IEEE, 2005.
- [134] Zheyuan Lu, Zhencheng Hu, and Keiichi Uchimura. Slam estimation in dynamic outdoor environments: A review. In International Conference on Intelligent Robotics and Applications, pages 255–267. Springer, 2009.
- [135] Wei Xu and Fu Zhang. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317-3324, 2021.
- [136] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5135– 5142. IEEE, 2020.
- [137] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.
- [138] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidarinertial odometry. *arXiv preprint arXiv:2107.06829*, 2021.
- [139] Chao Qin, Haoyang Ye, Christian E Pranata, Jun Han, Shuyang Zhang, and Ming Liu. Lins: A lidar-inertial state estimator for robust and efficient navigation. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 8899–8906. IEEE, 2020.
- [140] Sam Roweis. Levenberg-marquardt optimization. *Notes, University Of Toronto,* 1996.
- [141] Nicholas Ohi, Kyle Lassak, Ryan Watson, Jared Strader, Yixin Du, Chizhao Yang, Gabrielle Hedrick, Jennifer Nguyen, Scott Harper, Dylan Reynolds, et al. Design of an autonomous precision pollination robot. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 7711–7718. IEEE, 2018.
- [142] Jared Strader, Jennifer Nguyen, Christopher Tatsch, Yixin Du, Kyle Lassak, Benjamin Buzzo, Ryan Watson, Henry Cerbone, Nicholas Ohi, Chizhao Yang, et al. Flower interaction subsystem for a precision pollination robot. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5534–5541. IEEE, 2019.
- [143] Chizhao Yang, Ryan M Watson, Jason N Gross, and Yu Gu. Localization algorithm design and evaluation for an autonomous pollination robot. ION, 2019.
- [144] Cagri Kilic, Nicholas Ohi, Yu Gu, and Jason N Gross. Slip-based autonomous zupt through gaussian process to improve planetary rover localization. *IEEE Robotics and Automation Letters*, 6(3):4782–4789, 2021.
- [145] Cagri Kilic, Jason N Gross, Nicholas Ohi, Ryan Watson, Jared Strader, Thomas Swiger, Scott Harper, and Yu Gu. Improved planetary rover inertial navigation and wheel odometry performance through periodic use of zero-type constraints. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 552–559. IEEE, 2019.

[146] Patrick Pfreundschuh, Hubertus Franciscus Cornelis Hendrikx, Victor Reijgwart, Renaud Dubé, Roland Siegwart, and Andrei Cramariuc. Dynamic object aware lidar slam based on automatic generation of training data. arXiv preprint arXiv:2104.03657, 2021.