Scan Registration Using the Normal Distributions Transform and Point Cloud Clustering Techniques

by

Arun Das

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2013

Abstract

As the capabilities of autonomous vehicles increase, their use in situations that are dangerous or dull for humans is becoming more popular. Autonomous systems are currently being used in several military and civilian domains, including search and rescue operations, disaster relief coordination, infrastructure inspection and surveillance missions. In order to perform high level mission autonomy tasks, a method is required for the vehicle to localize itself, as well as generate a map of the environment. Algorithms which allow the vehicle to concurrently localize and create a map of its surroundings are known as solutions to the Simultaneous Localization and Mapping (SLAM) problem. Certain high level tasks, such as drivability analysis and obstacle avoidance, benefit from the use of a dense map of the environment, and are typically generated with the use of point cloud data. The point cloud data is incorporated into SLAM algorithms with scan registration techniques, which determine the relative transformation between two sufficiently overlapping point clouds. The Normal Distributions Transform (NDT) algorithm is a promising method for scan registration, however many issues with the NDT approach exist, including a poor convergence basin, discontinuities in the NDT cost function, and unreliable pose estimation in sparse, outdoor environments.

This thesis presents methods to overcome the shortcomings of the NDT algorithm, in both 2D and 3D scenarios. To improve the convergence basin of NDT for 2D scan registration, the Multi-Scale k-Means NDT (MSKM-NDT) algorithm is presented, which divides a 2D point cloud using k-means clustering and performs the scan registration optimization over multiple scales of clustering. The k-means clustering approach generates fewer Gaussian distributions when compared to the standard NDT algorithm, allowing for evaluation of the cost function across all Gaussian clusters. Cost evaluation across all the clusters guarantees that the optimization will converge, as it resolves the issue of discontinuities in the cost function found in the standard NDT algorithm. Experiments demonstrate that the MSKM-NDT approach can be used to register partially overlapping scans with large initial transformation error, and that the convergence basin of MSKM-NDT is superior to NDT for the same test data.

As k-means clustering does not scale well to 3D, the Segmented Greedy Cluster NDT (SGC-NDT) method is proposed as an alternative approach to improve and guarantee convergence using 3D point clouds that contain points corresponding to the ground of the environment. The SGC-NDT algorithm segments the ground points using a Gaussian Process (GP) regression model and performs clustering of the non ground points using a greedy method. The greedy clustering extracts natural features in the environment and generates Gaussian clusters to be used within the NDT framework for scan registration.

Segmentation of the ground plane and generation of the Gaussian distributions using natural features results in fewer Gaussian distributions when compared to the standard NDT algorithm. Similar to MSKM-NDT, the cost function can be evaluated across all the clusters in the scan, resulting in a smooth and continuous cost function that guarantees convergence of the optimization. Experiments demonstrate that the SGC-NDT algorithm results in scan registrations with higher accuracy and better convergence properties than other state-of-the-art methods for both urban and forested environments.

Acknowledgements

I would like to express my gratitude, appreciation and respect for my supervisor, Dr. Steven Waslander, who provided me with the opportunity to work in such a rich, interesting and addicting research area. His constant encouragement, drive, and support has enabled me to grow stronger as a person, a researcher, an academic, and as a student.

My life at the Waterloo Autonomous Vehicles Lab would not have been nearly as interesting without all the lab members, past and present, who made this experience memorable. I would like to thank Neil, PJ, Carlos and Yan, who were always there for me in the good times and the bad. A special thanks to James, who was always able to finish experiments on tight deadlines. Finally, thanks to Yassir, Peiyi, Adeel, Kevin, Sirui, Mike S., Mike T., John, Nima, and Sid, who were always willing to discuss new ideas and patiently listen to me as I rambled away on tangents.

I would also like to thank the University of Waterloo Robotics team, for all the work that went into numerous competitions we entered over the years, and the great memories that came along with them. Thanks to the University of Waterloo Micro Aerial Vehicles team for being good sports when we took over their work bay during crunch time.

Finally, I would like to thank my mother, father, family and friends who always gave me their unconditional support throughout this journey.

Dedication

This thesis is dedicated to my father, who fearlessly battled against cancer, and to anyone whose life has been affected by this terrible disease. I would also like to dedicate this work to my brave, strong mother, my sister, and my family and friends who have stood beside us every step of the way.

Table of Contents

List of Tables **ix**

List of Figures **x**

1 Introduction **1**

1.1 SLAM Algorithms . . . 5

1.2 Scan Registration for Graph SLAM . . . 7

1.3 Contributions . . . 11

2 Scan Registration Using Point Cloud Data **14**

2.1 Problem Formulation . . . 14

2.2 The Iterative Closest Point Algorithm . . . 15

2.3 The Generalized Iterative Closest Point Algorithm . . . 18

2.4 The Normal Distributions Transform Algorithm . . . 22

2.5 K-means Clustering . . . 26

3 Multi-Scale K-Means NDT **29**

3.1 Proposed Method: Multi-Scale K-Means NDT (MSKM-NDT) . . . 30

3.2 Experimental Results . . . 32

3.2.1 SLAM Experiments . . . 35

3.2.2 Convergence Tests . . . 40

4 Segmented Greedy Cluster NDT **47**

4.1 Proposed Method 48

4.2 Ground Segmentation 50

4.2.1 Polar Grid Binning 50

4.2.2 Gaussian Process Regression 51

4.2.3 Ground Segmentation Using Gaussian Process Regression and Incremental Sample Consensus 54

4.3 Clustering of the Non-Ground Points 54

4.4 Segmented Greedy Cluster NDT 58

4.5 Experimental Results 60

4.5.1 Justification of Ground Segmentation 62

4.5.2 Accuracy in Outdoor Urban Environments 63

4.5.3 Accuracy in Sparse Forested Environments 68

5 Conclusion **74**

References **78**

List of Tables

3.1 Summary of success rates for registration experiments 41

4.1 Summary of map crispness for urban and forested environments 68

List of Figures

1.1 Sample 2D and 3D point clouds 4

1.2 Reconstructed map for graph SLAM algorithm 8

1.3 Highly connected pose graph and map for based graph SLAM algorithm . 9

2.1 Scan registration using ICP with point correspondences shown 20

2.2 Sample 2D LIDAR scan with a 50cm by 50cm NDT grid. 23

2.3 Probability density of the laser scan from NDT for each cell 24

3.1 Cost surface for 2D NDT registration 31

3.2 Scan clustering at scale $\phi = 3$ 33

3.3 Scan clustering at scale $\phi = 9$ 34

3.4 Scan clustering at scale $\phi = 20$ 35

3.5 Contour plots for cost surfaces with differing cluster scales 36

3.6 Map and vehicle motion using pair-wise ICP SLAM 37

3.7 Map and vehicle motion using pair-wise NDT SLAM 38

3.8 Map and vehicle motion using pair-wise MSKM-NDT SLAM 39

3.9 Convergence results for NDT registration 42

3.10 Convergence results for MSG-NDT registration 43

3.11 Convergence results for MSKM-NDT registration 44

4.1 Ground segmentation results for an outdoor scene 56

4.2 Resulting distributions after performing greedy clustering 59

4.3 Ford data set registration accuracy and run time results for ICP with varying degrees of ground removal 64

4.4 Ford data set registration accuracy and run time results for G-ICP with varying degrees of ground removal 65

4.5 Ford data set registration accuracy and run time results for NDT with varying degrees of ground removal 66

4.6 WPI data set registration crispness and run time results for ICP with varying degrees of ground removal 67

4.7 WPI data set registration crispness and run time results for G-ICP with varying degrees of ground removal 67

4.8 WPI data set registration crispness and run time results for NDT with varying degrees of ground removal 68

4.9 Error distributions of tested algorithms for the Ford campus data set . . . 69

4.10 Maps generated by aggregating scans using ICP and SGC-NDT 70

4.11 Top-down view for the maps generated using SGC-NDT-P2D and G-ICP . 72

4.12 Close up views for the maps generated using SGC-NDT-P2D and G-ICP . 73

Chapter 1

Introduction

Autonomous vehicle technology has the potential to significantly impact a wide range of applications, leading to large scale societal impact. For example, autonomous driving will enable the more efficient use of existing driving infrastructure, reducing the cost and resource requirements for the construction of new roads. The Netherlands Organisation for Applied Scientific Research (TNO) estimates that by 2022, cooperative driving will able to reduce traffic fatalities by 8%, reduce carbon dioxide and fuel consumption by 5%, and reduce the time lost to traffic jams by 50% [59]. Autonomous vehicles also have the potential to greatly improve the safety of humans in dangerous environments. In the United States, from 1900 through 2007, there were at total of 12,823 fatalities from mining related accidents, and the majority of the fatalities involve mine disaster rescue teams who are specially trained for search and rescue in hostile and dangerous environments [40]. In search and rescue situations, mobile autonomous vehicles have great potential, where they can be used to scout ahead of the rescue team and report potentially hazardous conditions [40, 47]. The range of applications for autonomous vehicles is vast, from autonomous space rovers which search for evidence of sustained life on Mars [30, 41], to autonomous air, ground and water vehicles that allow scientists to monitor harmful algal blooms and pollution spread around the Earth [14].

Innovation in autonomous vehicle technology has already proven to be very effective in real-world applications. In the area of autonomous cars, the fleet of Google cars has travelled more than 300,000 km in environments ranging from city traffic, to busy highways and mountainous roads [58]. The VisLab has also performed extensive experiments, where four autonomous vehicles drove from Parma, Italy to Shanghai, China along a 13,000 km route where detailed map and environment information was not available [9]. The Mars Exploration Rovers (MERs) have been continuously improving autonomous capabilities in order to improve the overall activity time of the vehicle [61, 37]. The Mars Science Laboratory (MSL), which is commonly referred to as *Curiosity*, implements parallelized navigation algorithms, allowing for continuous planetary rover navigation and eliminates the need for the vehicle to stop and perceive the environment [20]. The Aeryon Scout Unmanned Aerial Vehicle (UAV) has been deployed in many tactical military, law enforcement and civilian operations to aid in intelligence missions, emergency response and infrastructure inspection. The gradually decreasing cost of robotic systems and their increasing capabilities has brought forth proposed applications within household settings, such as the iRobot Roomba vacuum cleaner and Willow Garage's PR2 personal robot [7].

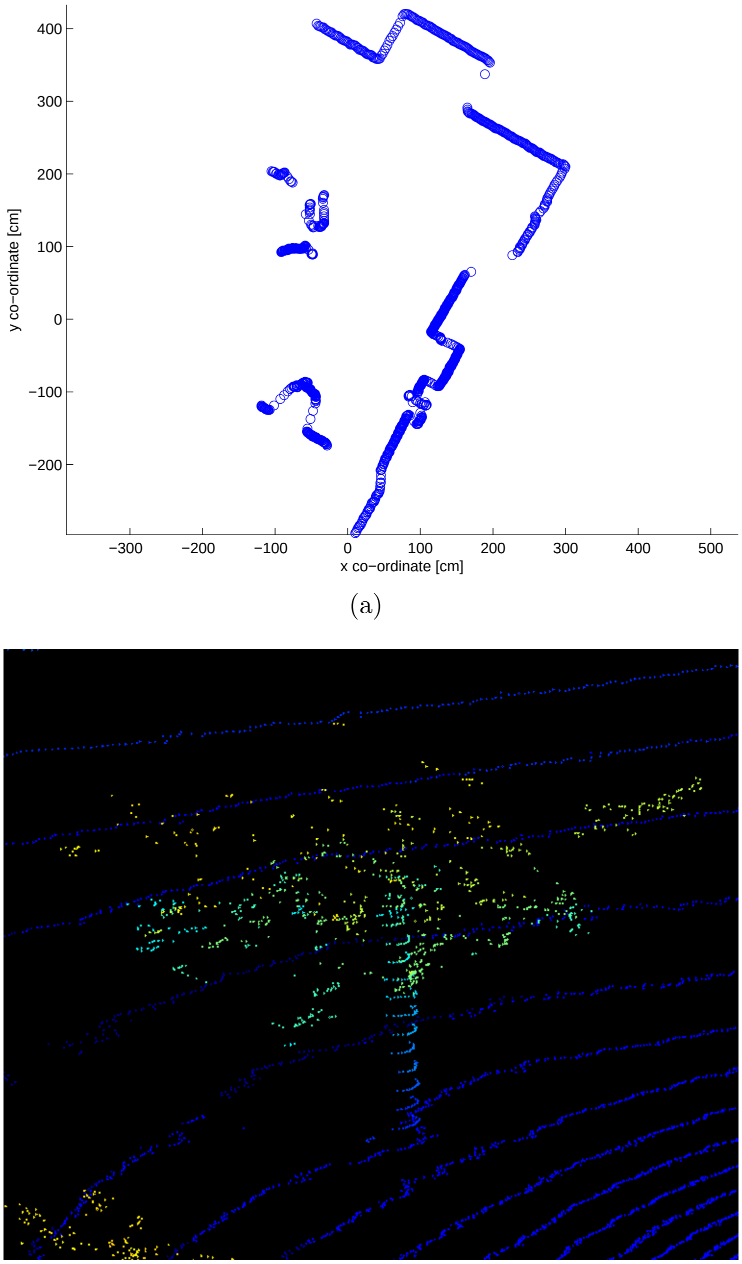
In order to autonomously perform high level tasks such as vehicle path planning, obstacle avoidance and sample return, an autonomous vehicle requires a method to *localize* its position and maintain a *map* of the environment. In the localization problem, the vehicle uses sensors to detect and locate landmarks, or features, within the environment. Through the detection and matching of landmarks, the vehicle can determine its location relative the environment as it travels. In the mapping problem, it is assumed the vehicle pose, its current position and orientation, is known, and the vehicle uses the collected sensor data from its traversal to maintain a representation of its surroundings. When the vehicle is required to traverse the environment and neither the vehicle location nor the map is known, a solution to the *Simultaneous Localization and Mapping*, or SLAM problem, is required. For example, the Mars rover uses a camera based system to detect and map landmark locations in the environment, while simultaneously localizing its own position against the landmarks. The SLAM problem is particularly difficult to solve because it is a chicken or egg problem. That is, an accurate map is required for localization of the vehicle, however, an accurate vehicle pose is also required to generate a consistent map. As the vehicle traverses the environment, errors accumulate due to uncertainty in vehicle motion, which is coupled with sensor error as measurements to landmarks are taken. The accumulation of error has the potential to distort the map and corrupt the vehicle's knowledge of it's true location within the environment.

Although solutions to the SLAM problem concurrently estimate the map and vehicle pose, it is also is possible to perform localization and mapping independently [25]. The naive approach to determine vehicle motion involves using wheel or visual *odometry*, which is the use of data from moving sensors to estimate change in position over time. The integration of odometry information over time is unreliable due to unbounded error and drift in the pose, thus the approach is rarely ever used in isolation. An improved method to determine vehicle motion relies on the combination of Global Positioning System (GPS), inertial measurements and wheel odometry, however these methods also have failure modes due to loss of GPS and vehicle wheel slip. By using the environmental information to maintain a map of the surroundings, the SLAM approach can provide additional robustness and improved localization accuracy over pure pose estimation.

There are two main categories of environmental information that can be used to generate two different types of maps, *sparse* feature based maps and *dense* maps. Monocular cameras can be used to generate sparse data in the form of stable, identifiable landmarks and features, however, dense environment information, such as terrain gradients and obstacle geometry is not readily attainable from camera solutions. The dense environment information is especially key for autonomy tasks such as drivability analysis, path planning and obstacle avoidance, as dense environment information is required to generate paths which are guaranteed to be collision free and traversable by the vehicle. The dense information can be used to perform *dense SLAM* which generates a dense map of the environment, as opposed to a *feature SLAM*, where only certain features are tracked in the map.

In typical robotics applications, dense environment information is provided in a structure called a *point cloud*. A point cloud is a set of points which represent the environment as a collection of samples from 2D or 3D space. Point clouds can be generated using many different sensors, such as stereo cameras, RGB-D sensors, TOF cameras and laser scanners. A stereo camera generates point clouds by identifying corresponding pixels across time synchronized image frames and calculating the depth of the pixels using known camera geometry. In practice, point cloud generation from a stereo camera requires stereo matching algorithms which are computationally expensive and unreliable in low texture environment without sufficient illumination. RGB depth (RGB-D) cameras, such as the Microsoft Kinect, provide camera colour images as well as depth information, which is determined using structured infared (IR) light projectors and 3D reconstruction algorithms. Although RGB-D cameras are able to produce dense, accurate point clouds, they are currently limited to indoor use only. Generally more expensive, the preferred method for the generation of point cloud data in a robotics application is through the use of a LIDAR scanner, which generates point cloud data directly using the time of flight of rotating IR lasers, and is not affected by surface texture or changing illumination. Example point clouds generated from a 2D and 3D LIDAR are provided in Figure 1.1. A related technology to LIDAR is the time of flight (TOF) camera, which uses a modulated IR laser pulse to capture information regarding an entire section of the environment using a single laser, in contrast to a scanning LIDAR which rotates the laser to collect data in a point-by-point fashion.

In order to perform dense SLAM using point clouds, a procedure known as *point cloud scan registration*, or simply *scan registration*, is required. Suppose two point cloud scans, or simply, scans, are collected with the sensor from two different vehicle poses. Let a scan that is collected from some initial vehicle pose be called the *reference scan*, and a



(a)

(b)

Figure 1.1: (a) A 2D point cloud of indoor room environment collected with a Hokuyo LIDAR. (b) A 3D point cloud of a tree within an outdoor environment collected with a Velodyne LIDAR.

scan collected at a different vehicle location be the *scene scan*. Given sufficient overlap between the two scans, it is possible to determine the relative rigid body transformation that aligns the two point clouds. The process of determining the relative transformation between the reference scan and the scene scan is known as *scan registration*. Specifically, a scan registration algorithm will return the transformation which must be applied to the scene scan in order to align it to the coordinate frame of the reference scan. Feature based scan registration methods, such as Normal Aligned Radial Features [53] and Point Feature Histograms [49, 48] can be used, but suffer from difficulties in consistently and uniquely identifying features in the scan data, resulting in less accurate scan registrations when compared to methods that operate on the entire point cloud.

Much work has been done in the area of scan registration, however many issues remain, including the poor convergence properties of the algorithms, high computational complexity, particularly on large point clouds, and inconsistent pose estimation quality when applied to a wide range of environments. The scan registration problem is especially challenging in large, unstructured environments, as the sparsity of the point cloud makes the extraction of useful information from the scan difficult. The main objective of this thesis is to address the issues with current scan registration algorithms by presenting improved methods with better convergence and reduced sensitivity to sparsity of points in the point cloud. The algorithms are evaluated using 2D and 3D point cloud data of indoor and outdoor environments, using both experimentally collected point cloud scans and data from well known, open source repositories within the robotics community.

1.1 SLAM Algorithms

The earliest work in the SLAM area established the requirement for a statistical method for describing relationships between landmark locations and the vehicle position. A key element of the work was to show the correlation between estimates of the location of different landmarks in a map, and that the correlations would evolve with successive observations [15, 51]. In practice, the landmarks can be determined through camera based feature detection and matching methods, such as Scale Invariant Feature Transform (SIFT) [32], Speeded-up Robust Features (SURF) [2] or Center-Surround Extreama (CenSurE) features [1], and alternatives which operate on point cloud data.

The SLAM problem was next solved within the Bayesian estimation framework [55]. The most well known example of this approach used an Extended Kalman Filter (EKF) formulation known as EKF-SLAM, which solves the SLAM problem by estimating the vehicle pose and the positions of landmarks or features as states of the EKF. Although conceptually the EKF-SLAM method is simple, it suffers from many drawbacks including computational complexity, sparse feature maps, and poor robustness due to the linearization step performed at each update of the EKF. Furthermore, the EKF based SLAM approach is susceptible to divergence if the data association between currently measured features and previously visited features is not correct. Computational complexity is a concern, as the complexity of the algorithm is $O(n^3)$, where n is the number of features being mapped. The map generated through the EKF-SLAM is also a sparse map, as only feature locations are tracked, thus the map is not directly usable for other tasks such as path planning and collision avoidance. Computational complexity is somewhat addressed by the Sparse Extended Information Filter (SEIF) SLAM method [56], which is reported to operate at nearly constant time by pruning weak links in the information matrix in order to generate a sparse approximation. The issue of linearization is somewhat addressed with the Sigma Point Kalman Filter (also called the Unscented Kalman Filter or UKF) approach [22], which uses robust statistical sampling methods to linearize highly nonlinear motion and measurement model functions. Although the the concerns of computation and linearization can be addressed through some of the Bayesian SLAM variants, the issues of sparse map generation and divergence due to poor data association remain. Finally, the Bayesian SLAM formulations are *online SLAM* algorithms, meaning that only the current vehicle pose and landmark locations are estimated. In contrast a *full SLAM* algorithm is one which tracks the vehicle and landmark states throughout the traversal history. It should be noted that although Bayesian SLAM is an online method, a correlation between states is maintained that allows for the update all of the landmark positions, however, the linearizations performed at each time step cannot be reversed, thus the state correlations are potentially inaccurate. The distinction between online SLAM and full SLAM is important, as online SLAM algorithms can only correct the vehicle and landmark locations for the current time-step, while full SLAM algorithms have the ability to correct the history of the vehicle and landmark states, and can drastically improve the map and localization accuracy when areas in the environment are revisited.

Current state of the art SLAM algorithms [28, 19, 26, 45, 23, 43] can be categorised as variants of graph based SLAM, or graph SLAM, which poses the SLAM problem within a graph structure. Graph SLAM algorithms are composed of two components, the *front-end* and the *back-end*. The front-end is responsible for generating a *pose graph*, which is a collection of vertices and edges that describe the motion of the vehicle and associated measurements to landmarks. In the pose graph, a vertex represents a pose of the vehicle or landmark, and an edge between any two vertices is a constraint which connects the poses described by the vertices. For example, an edge constraint could be the result of an odometry reading between vehicle poses, or a measurement to a landmark from a given vehicle pose. As the pose graph is populated and as edges are continually added, it is likely that the pose graph may reach an over-constrained state. In the over-constrained case, a graph optimization is formulated as a nonlinear optimization problem. The back-end of the graph SLAM algorithm is responsible for solving the nonlinear optimization problem and minimizes the sum of the errors introduced by the edge constraints by performing an optimization over the graph. In practice, the graph optimization is triggered when a *loop-closure* is detected, or when the vehicle revisits a portion of the map it has seen previously. Through the graph optimization, the graph SLAM algorithm updates the full state of the vehicle and state-of-the art graph SLAM back-ends, such as TORO [18] and g2o [27], are able to perform the graph optimization in a computationally efficient manner using intelligent data structure manipulation and problem parametrization strategies. The ability of the graph SLAM algorithm to update the vehicle pose and the map along its history allows for the generation of accurate, globally consistent maps that are achievable in real time.

As a motivating example of graph SLAM, consider the environment and reconstructed map presented in Figure 1.2. In the example, the vehicle developed for the University of Waterloo's entry to the 2012 NASA Sample Robot Return Challenge navigates a figure eight which covers an area of approximately 400m by 300m. The pose graph is generated using scan registration, and is then refined using pose graph optimization in real-time. Qualitatively, the reconstructed map (shown in black) is well aligned to features in the aerial image such as trees, bushes, buildings and posts. In Figure 1.2, the red dots represent the vertices of the graph, blue lines represent edge constraints between vertices, and the green line is the instantaneous vehicle position as it navigates the environment.

1.2 Scan Registration for Graph SLAM

Scan registration is used in the graph SLAM method to generate edges in the pose graph. Specifically, as the vehicle travels through the environment, a vertex can be added at the current vehicle position and the associated scan for that vertex can be saved. The combination of the graph vertex and its associated scan is commonly referred to as a *keyframe*. Provided there is sufficient point cloud overlap between the scans of any two keyframes, a scan registration can be performed to generate an edge constraint between the two associated vertices. The ability to add edge constraints for vertices where there is sufficient point cloud overlap allows for the generation of highly connected pose graphs, especially in the case where the sensor range is relatively large. An example of a map and a highly connected pose graph is given in Figure 1.3. In Figure 1.3, red dots represent

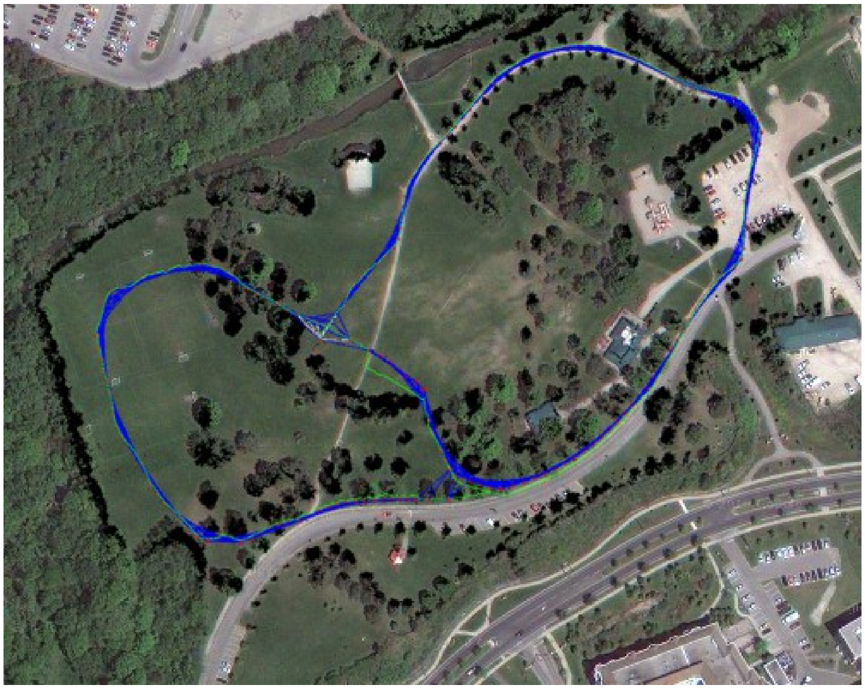


Figure 1.2: Reconstructed map and aerial image overlay for scan registration based graph SLAM algorithm.

the pose graph vertices, blue lines denote the edges of the graph, green represents the instantaneous vehicle path and black denotes the reconstructed map.

A significant amount of work has been done which uses scan registration within the graph SLAM frame work. In indoor environments, the earliest graph SLAM approach was introduced by Lu and Milios, who first proposed the full SLAM formulation as a graph, generated edge constraints using scan registration, and solved the sparse set of equations for the graph optimization [33]. The graph slam approach of Lu and Milios has been extended and refined, namely through methods to improve the efficiency and scalability of the back-end optimization routine [17]. The introduction of the Microsoft Kinect sensor has resulted in numerous innovations in indoor 3D mapping algorithms. The RGB-D SLAM algorithm [19] augments feature tracking using the camera RGB data with scan registration techniques from the point cloud data to perform an offline graph SLAM. The Kinect Fusion algorithm [42] is able to perform dense 3D tracking and reconstruction and achieves real time performance though the use of only scan registration algorithms, implemented on parallelized GPU architecture. Although the standard Kinect Fusion

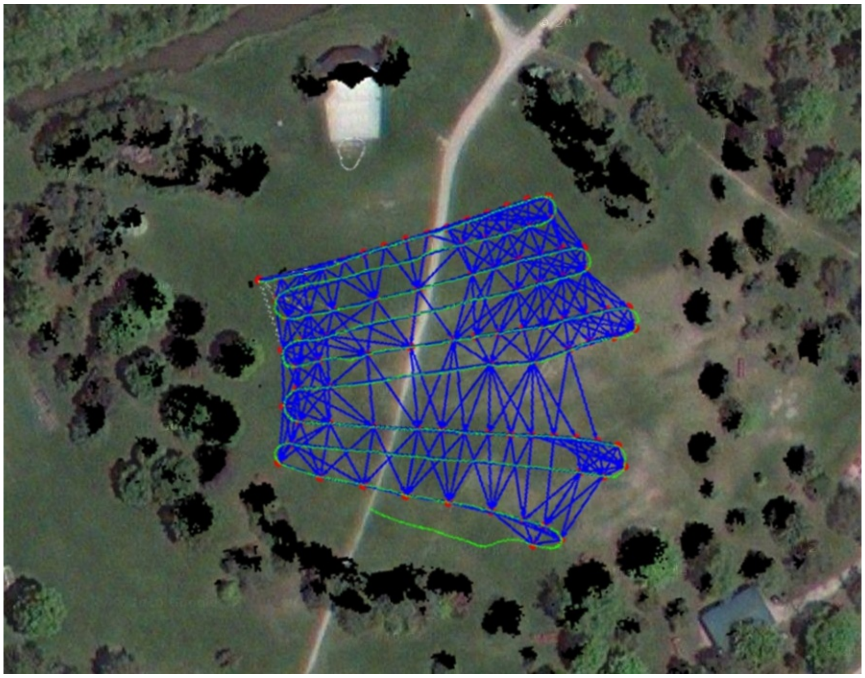


Figure 1.3: Highly connected pose graph and resulting map for scan registration based graph SLAM algorithm.

algorithm is restricted to operate in a limited space, extensions have been proposed which seek to extend the work volume [62]. In addition, the Kinect Fusion algorithm is not able to detect previously visited locations, thus loop closure remains an open problem.

In outdoor environments, high-fidelity and high accuracy maps of urban environments have been generated using 3D scan registration with both high resolution Reigl laser scanners [8], and with lower resolution sensors such as the 32 and 64 plane Velodyne scanners [39, 29]. It should be noted that in urban environments, flat, plane like structures such as walls and roads make the scan registration problem easier. In non-urban and unstructured environments, 3D scan registration techniques have been used to map locations such as mines [35], however a mine environment is relatively well structured with walls and caverns, thus making the scan registration problem easier. Mapping and navigation through forested environments has been performed by an unmanned aerial vehicle (UAV) [52], however success was mainly achieved with the use of a specialized tree-trunk cylinder feature detection method designed to operate in dense forested areas, and likely cannot be applied to generally sparse, outdoor scenes.

It is clear that scan registration is an important aspect of modern graph SLAM algorithms. The best known and most widely used scan registration algorithm is the Iterative Closest Point (ICP) algorithm. ICP was independently introduced by Besl and McKay [4], Chen and Medioni [10], and Zhang [63]. The ICP algorithm attempts to find transform parameters that minimize the Euclidean distance of corresponding *nearest neighbour* points between the reference scan and the scene scan. To compute the nearest neighbour point, a point in the scene scan is queried against the reference scan to determine the point in the reference scan which is the minimum Euclidean distance to the query point. The process of computing point correspondences and minimizing the Euclidean distance between corresponding points is performed iteratively until convergence, when the sum of Euclidean distances between corresponding point does not sufficiently decrease. It should be noted that the ICP algorithm does not take into account the underlying surface of the point cloud, as it operates at the point level. Furthermore, ICP assumes that the two point clouds can overlap perfectly, however in practice this is not the case due to noise and the sampling characteristics of the sensor. To address the shortcomings of ICP, Segal et al. introduced generalized-ICP (G-ICP) [50], which calculates the surface normal at each point, using local neighbourhoods of points, and only includes correspondences between points with similar surface normals between the two scans. The G-ICP cost fuction is structured such that it only includes information perpendicular to the plane defined at the point by the normal, and does not penalize differences in position of points along the plane. G-ICP explicitly takes sensor noise and sampling characteristic into account, thus improving the convergence basin and convergence rate when compared to ICP. The major shortcomings of both the ICP and G-ICP approach is the generation of the nearest neighbour correspondences, which is computationally expensive and has been shown to be the bottleneck of ICP algorithms [8].

A promising approach for scan registration, which has been recently introduced, is the Normal Distributions Transform (NDT). The NDT algorithm was first suggested by Biber and Strasser for scan registration and mapping [5]. It was later expanded for use with 3D point clouds by Magnusson et al. [34]. The NDT method is desirable because it does not require the computation of explicit point correspondences. Instead, the NDT algorithm represents the underlying scan as a set of Gaussian distributions that locally model the surface of the reference scan as a probability density function (PDF). Given a parameter estimate for the desired transformation, a score is assigned which quantifies the amount of overlap between the reference scan and the transformed scene scan. The overlap score is determined by evaluating the transformed scene scan points at the Gaussian distributions which model the reference scan. In order to register the two point clouds, a nonlinear optimization is performed to determine the transformation parameters such that the overlap score is maximized. Stoyanov et al. introduced a *distribution to distribution* matching extension to NDT, which generates Gaussian distributions for both the scene and reference scan and minimizes the L_2 distance between distribution sets of the scene and reference scan in order to perform registration. The distribution to distribution extension was shown to improve the convergence basin, as well as improve computation time. Although it has been shown that the NDT algorithm and its extensions are able to produce higher quality registration results when compared against ICP [36], a major shortcoming of the NDT algorithm is its poor convergence basin [11]. Furthermore, as the Gaussian distributions are generated by partitioning the point cloud with rectilinear grid cells, the nonlinear cost function used to determine the transform parameters is not well defined as points cross cell boundaries during the optimization. As a result, convergence of the optimization is not guaranteed.

It should be noted that due to planar structures and high object density, scan registration algorithms generally perform well in structured urban environments. A large, forested environment makes laser scan registration based approaches especially difficult since a typical point cloud from a laser scanner such as the Velodyne-HDL32E sensor is sparse and relatively noisy in such a setting. The state of the art scan registration algorithms generally make assumptions about point cloud data which are not valid in forested environments. For example, ICP methods require a high point data density in order to provide accurate correspondences for nearest neighbour search [44]. G-ICP requires the computation of surface normal data, which is difficult to perform accurately with noisy point cloud data of grass, trees and shrubs, all of which are typical of a forested environment. Conceptually, the NDT approach is suitable for a forested area, as the sparsity issue can be resolved through the modelling of the point cloud with Gaussian distributions. It is clear that the NDT algorithm is a promising new approach to perform scan registration, which is a fundamental aspect of modern SLAM algorithms. Although promising, the NDT algorithm suffers from three major shortcomings: the convergence basin of the NDT algorithm is poor, the optimization of the NDT cost function is not guaranteed to converge, and the partitioning of the point cloud using rectilinear cells is naive and results in poor registration accuracy in sparse environments.

1.3 Contributions

The goal of this research is to address the issue of poor convergence of the NDT algorithms through multi-scale optimization methods, and address the accuracy of the NDT algorithm in sparse environments through improved point cloud clustering techniques.

The Multi-Scale-K-Means NDT (MSKM-NDT) algorithm is proposed, which divides a 2D point cloud using k-means clustering and performs the optimization step at multiple scales of clustering. The k-means clustering approach guarantees that the optimization will converge, as it resolves the issue of discontinuities in the cost function found in the standard NDT algorithm. The optimization step of the NDT algorithm is performed over a decreasing scale, which greatly improves the basin of convergence. Experiments show that the MSKM-NDT approach can be used to register partially overlapping scans with large initial transformation error, and that the convergence basin of MSKM-NDT is superior when compared against NDT for the same test data. It is also determined through experiment that the MSKM-NDT algorithm provides a viable solution for scan registration in a 2D, indoor, SLAM scenario.

As the k-means clustering does not scale well to 3D, an alternative approach to improve and guarantee convergence using 3D point clouds is proposed. In order to generate clusters in the 3D point cloud, the ground is first segmented using a Gaussian Process (GP) regression model, and the non ground points are then clustered using a greedy clustering method. The proposed method, Segmented Greedy Cluster NDT (SGC-NDT) uses the natural features in the environment to generate Gaussian clusters and performs scan registration using the clusters within the NDT framework. By segmenting the ground plane and clustering the remaining features, the SGC-NDT approach results in a smooth and continuous cost function which guarantees that the optimization will converge. Experiments demonstrate that the SGC-NDT algorithm results in scan registrations with higher accuracy and better convergence properties when compared against other state-of-the-art methods, such as ICP, G-ICP and NDT, for both urban and forested environments.

In summary, the main results presented in this thesis are:

- A multi-scale optimization NDT algorithm which improves the convergence basin of the NDT algorithm,
- Clustering techniques which can be applied within the NDT framework that modify the NDT cost function such that the optimization is well defined and guaranteed to converge,
- A greedy clustering based NDT algorithm which uses natural features in the environments to improve registration accuracy,
- Experimental results demonstrating the performance gains of the proposed algorithms in indoors, urban, and sparse forested environments.

The proposed MSKM-NDT algorithm and SGC-NDT algorithm are verified methods that address the shortcomings of the NDT algorithm. Through multi-scale optimization and modifications made to the NDT cost function, the proposed work improves the convergence basin of NDT and guarantees convergence of the nonlinear optimization. Through improved clustering methods of 3D point clouds, the proposed work demonstrates significant performance gains over NDT for scan registration in sparse, forested environments. With the improved robustness of the MSKM-NDT and SGC-NDT algorithms over standard NDT, opportunities exist that allow for the deployment of these algorithms on consumer grade hardware in a variety of autonomous vehicle applications.

The remainder of this thesis is organized as follows. Chapter 2 provides further background information on scan registration and presents specific details on the ICP, G-ICP and NDT algorithms. Chapter 3 presents the MSKM-NDT algorithm and experimental results which validate the approach for indoor settings. Chapter 4 presents the SGC-NDT algorithm, detailed information on GP based ground segmentation, and experimental results which validates the SGC-NDT approach in both urban and forested environments. Finally, Chapter 5 provides a conclusion of the presented work, as well as future directions for this research.

Chapter 2

Scan Registration Using Point Cloud Data

The goal of scan registration is to determine the transformation which can be applied to the scene scan, such that it best overlaps with the reference scan. To determine the transformation parameters which result in the best overlap, scan registration algorithms define a cost function which quantifies the amount of overlap between the point clouds. As the optimization variables consist of translation and rotation parameters, the cost functions for the scan registration algorithms are nonlinear. As a result, the assumption for scan registration is that the global minimum of the cost function yields the optimal transformation parameters that best align the two point clouds. This chapter formulates the scan registration problem for three competing algorithms: Iterative Closest Point (ICP), Generalized ICP (G-ICP) and the Normal Distributions Transform (NDT).

2.1 Problem Formulation

Define a point cloud as the set of points $P = \{p_1, \ldots, p_{N_P}\}$ where $p_i \in \mathbb{R}^d$ for $i \in \{1, \ldots, N_P\}$. Let d denote the dimensionality of the point cloud data, where $d = 2$ for 2D data and $d = 3$ for 3D data. In the 3D case, a point $p_j \in P$ consists of three components, $p_j = \{p_j^x, p_j^y, p_j^z\}$, which refer to the x, y, and z components of the point, respectively. Similarly in 2D, the point $p_j \in P$ consists of two components, $p_j = \{p_j^x, p_j^y, \}$, which refer to the x and y components of the point, respectively.

To transform a point from the coordinate frame of the scene scan to the coordinate frame of the reference scan, transformation parameters are required. In 2D, these transform parameters are given by $\mathcal{E} = [t_x, t_y, t_\theta]^T \in \mathbb{SE}(2)$ and in 3D, the transform parameters are given by $\mathcal{E} = [t_x, t_y, t_z, t_\phi, t_\theta, t_\psi]^T \in \mathbb{SE}(3)$. Note that the rotations will be defined by a $3-2-1$ Euler angle parametrization. In 3D, the transformation function, $T_\mathcal{E} : \mathbb{R}^3 \mapsto \mathbb{R}^3$, that maps a point from the scene scan, $p \in P^S$, into the coordinate frame of the reference scan, using a parameter estimate, $\mathcal{E}$, is given as

$$T_\mathcal{E}(p) = \left[\begin{array}{c} R_\phi R_\theta R\psi \end{array} \right] p + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{2.1}$$

where R_ϕ is the rotation matrix for a rotation about the x axis by angle ϕ, R_θ is the rotation matrix for a rotation about the y axis by angle θ, R_ψ is the rotation matrix for a rotation about the z axis by angle ψ, and $[t_x, t_y, t_z]^T$ is the translation vector between the origins of the two frames. Note that although the mapping $T_\mathcal{E}$ as presented is for the 3D case where $d = 3$, a similar transformation function can be derived for the 2D case. For notational simplicity, the mapping, $T_\mathcal{E}$, is applied for both the 2D and 3D case.

Scan registration algorithms seek to find the optimal transformation between two point cloud scans, a *reference scan*, P^R, and a *scene scan*, P^S. The goal is to determine transformation parameters which best align the two, such that the two scans overlap as much as possible. To quantify the amount of overlap between the two scans, a score function is required, and is denoted by the mapping $\Lambda : \mathbb{SE}(d) \mapsto \mathbb{R}$. Finally, the *scan registration problem* can be defined as determining the optimal transform parameters, $\mathcal{E}^*$, such that

$$\mathcal{E}^* = \underset{\mathcal{E} \in \mathbb{SE}(d)}{\operatorname{argmin}} \Lambda(\mathcal{E}) \tag{2.2}$$

Note that the score function is algorithm specific. The remainder of this section will introduce competing scan registration algorithms found within the literature, including ICP, G-ICP and NDT.

2.2 The Iterative Closest Point Algorithm

Given a scene scan P^S, and a reference scan P^R, the ICP algorithm seeks to find the parameters $\mathcal{E}$ which minimize a cost function based on corresponding points between the reference scan and the scene scan. For ICP, the point correspondence for point p_i in the reference scan is its nearest neighbour point p_j within the scene scan. Define the point to set distance between a point p and a point cloud P as

$$\bar{d}(p)_P := \min_{\hat{p}\in P} \|p - \hat{p}\| \tag{2.3}$$

Using the definition of the point to set distance, a correspondence term w can be defined. For a point in the scene scan $p_j \in P^S$ that has been transformed by parameter estimate $\mathcal{E}$, denote w_j as the respective correspondence term to it's nearest neighbour in the reference scan P^R. The correspondence term w_j is given as

$$w_j = \begin{cases} 1, & \text{if } \bar{d}(T_{\mathcal{E}}(p_j))_{P^R} < \delta_{max}, \\ 0, & \text{otherwise.} \end{cases}$$

where δ_{max} denotes the maximum allowable neighbourhood to consider for a point correspondence. In practice, the point clouds are partially overlapping, thus a point in the scene scan may not have a corresponding point in the reference scan. To ensure that outlier points are not corresponded, the maximum allowable neighbourhood should be appropriately determined. If the δ_{max} is too large, potential outlier point correspondences will be allowed, which reduces the accuracy of the scan registration. If the δ_{max} parameter is not sufficiently large, correct point correspondences may be rejected, which could result in the optimization converging to local minimum. Define $p_j^{\mathcal{E}}$ as the point in the reference scan that is the nearest neighbour to point p_j in the scene scan that has been transformed by $\mathcal{E}$, or

$$p_j^{\mathcal{E}} = \arg\min_{\hat{p}\in P^R}(\|T_{\mathcal{E}}(p_j) - \hat{p}\|)$$

Finally, the score function for ICP minimization can be presented as,

$$\Lambda_{icp}(\mathcal{E}) = \sum_{j=1}^{|P^S|} w_j \|p_j^{\mathcal{E}} - T_{\mathcal{E}}(p_j)\|^2 \tag{2.4}$$

The ICP algorithm iteratively calculates the transformation parameters $\mathcal{E}$ by selecting the nearest neighbour points between the reference scan and incoming scan as correspondences, then minimizing Equation 2.4. Let $M_{\mathcal{E}}$ be the total number of point correspondences between the point clouds, given the current transform parameters, $\mathcal{E}$. Then, denote the point clouds of the reference scan and the scene scan containing only corresponding points as $\overline{P^R} \subseteq P^R$ and $\overline{P^S} \subseteq P^S$, where $|\overline{P^R}| = |\overline{P^S}| = M_{\mathcal{E}}$. Further assume that the point clouds $\overline{P^R}$ and $\overline{P^R}$ have been re-ordered such that corresponding points between the scans are indexed in order. It should be noted that while the nearest neighbour point correspondence assumption is somewhat valid for dense point cloud data, in situations where the point cloud data is relatively sparse, the assumption is poor.

Once the corresponding points for the current iteration are known, the optimal solution for the iteration can be determined through numerical methods such as Gauss-Newton or Levenberg-Marquardt, or closed form methods [44]. For the closed form solution, the computation of the optimal rotation and optimal translation can be decoupled [44]. Denote the rotation matrix and translation vector corresponding to the optimal transformation parameters as $\mathcal{E}_r^* \in \mathbb{R}^{d \times d}$, and $\mathcal{E}_t^* \in \mathbb{R}^d$, respectively. To determine the optimal parameters, first the centroid of each scan is calculated,

$$\mathcal{C}^R = \frac{1}{M_{\mathcal{E}}} \sum_{p^R \in \overline{P^R}} p^R$$

$$\mathcal{C}^S = \frac{1}{M_{\mathcal{E}}} \sum_{p^S \in \overline{P^S}} p^S$$

where $\mathcal{C}^R \in \mathbb{R}^d$ is the centroid for the reference scan $\overline{P^R}$ and $\mathcal{C}^S \in \mathbb{R}^d$ is the centroid for the scene scan $\overline{P^S}$. Once the centroids have been calculated, the point clouds can be redefined as deviations from their centroid values. The deviated reference and scene point clouds, $\hat{P^R}$ and $\hat{P^S}$, are defined as

$$\hat{P^R} = \{\hat{p} \in \mathbb{R}^d : \hat{p} = p - \mathcal{C}^R \,\forall\, p \in \overline{P^R}\}$$
$$\hat{P^S} = \{\hat{p} \in \mathbb{R}^d : \hat{p} = p - \mathcal{C}^S \,\forall\, p \in \overline{P^S}\}$$

Using the centroid deviated point clouds, a correlation matrix, $H \in \mathbb{R}^{d \times d}$, can be constructed using the points $\hat{p}^R \in \hat{P^R}$ and $\hat{p}^S \in \hat{P^S}$. The correlation matrix is computed as

$$H = \sum_{i=1}^{M_{\mathcal{E}}} \hat{p}_i^R \otimes \hat{p}_i^S$$

where $\otimes$ denotes the vector outer product. As described in [44], from the correlation matrix H, the optimal rotation is given as

$$\mathcal{E}_r^* = VU^T$$

where $V \in \mathbb{R}^{d \times d}$ and $U \in \mathbb{R}^{d \times d}$ are unitary matrices given by the singular value decomposition $H = U\mathcal{D}V^T$. Once the optimal rotation has been calculated, the optimal translation is given by

$$\mathcal{E}_t^* = \mathcal{C}^R - \mathcal{E}_r^* \mathcal{C}^S \tag{2.5}$$

The full proof for the optimal close form solution is presented in [44]. It should be noted that the nearest neighbour search is generally the most computationally expensive component of the ICP algorithm. To decrease the computation time required by nearest neighbour search, methods such as the kd-tree look-up have been developed, which can significantly improve the nearest neighbour query time. For a full description of nearest neighbour look up methods, the reader is directed to [44].

For the current iteration, denoted by I, the ICP algorithm calculates the optimal transform parameters, $\mathcal{E}^*$, using the corresponding nearest neighbour points of the current iteration, as well as the respective score value for the iteration, $E_I = \Lambda_{icp}(\mathcal{E}^*)$. The algorithm terminates either after I_{max} iterations or if the decrease in the score is not sufficient, $\|E_{I-1} - E_I\| < \epsilon_{icp}$, where ϵ_{icp} is a user defined threshold value. Figure 2.1 illustrates the correspondences between two typical 2D scans. The ICP method is summarized in Algorithm 1.

There are also many variants of the ICP algorithm which compute additional properties of the point cloud and use these to improve the registration result. A popular variant of the ICP algorithm is the point-to-plane ICP algorithm. In the point to plane algorithm, the surface normal of the scan at a particular point is computed, and using this information the ICP cost function is modified to minimize the error along the surface normal direction. The point to plane version of the ICP algorithm has been shown to have a wider basin of convergence, at the cost of increased computational complexity associated with having to determine the surface normal direction at every point in the point cloud [44].

2.3 The Generalized Iterative Closest Point Algorithm

Through the minimization of the Euclidean distance between nearest neighbour points, the ICP algorithm fundamentally assumes that the application of the optimal transformation parameters to the scene scan will result in perfect overlap between the reference scan

Algorithm 1 Register scene scan P^S to reference scan P^R using ICP and initial parameter estimate $\hat{\mathcal{E}}$

```
for $I = 0$ to $I_{max}$ do
    $\overline{P^R} \leftarrow \emptyset$
    $\overline{P^S} \leftarrow \emptyset$
    $\mathcal{E}^* \leftarrow \hat{\mathcal{E}}$
    Perform nearest neighbour point correspondence:
    for all $p \in P^S$ do
        if $\bar{d}(T_{\mathcal{E}^*}(p))_{P^R} < \delta_{max}$ then
            $\overline{P^S} \leftarrow \overline{P^S} \cup T_{\mathcal{E}^*}(p)$
            $p^* = \underset{\hat{p} \in P^R}{\arg\min}(\|T_{\mathcal{E}^*}(p) - \hat{p}\|)$
            $\overline{P^R} \leftarrow \overline{P^R} \cup p^*$
        end if
    end for
    Minimize ICP cost function:
    $M_{\mathcal{E}} \leftarrow |\overline{P^S}|$
    $\mathcal{C}^R = \frac{1}{M_{\mathcal{E}}} \sum_{p^R \in \overline{P^R}} p^R$
    $\mathcal{C}^S = \frac{1}{M_{\mathcal{E}}} \sum_{p^S \in \overline{P^R}} p^S$
    $\hat{P^R} \leftarrow \{\hat{p} \in \mathbb{R}^d : \hat{p} = p - \mathcal{C}^R \, \forall \, p \in \overline{P^R}\}$
    $\hat{P^R} \leftarrow \{\hat{p} \in \mathbb{R}^d : \hat{p} = p - \mathcal{C}^S \, \forall \, p \in \overline{P^S}\}$
    $H = \sum_{i=1}^{M_{\mathcal{E}}} \hat{p}_i^R \otimes \hat{p}_i^S$
    $U\mathcal{D}V^T \leftarrow \mathrm{svd}(H)$
    $\mathcal{E}_r^* \leftarrow VU^T$
    $\mathcal{E}_t^* \leftarrow \mathcal{C}^R - (\mathcal{E}_r^*)\mathcal{C}^S$
    $E_I = \Lambda_{icp}(\mathcal{E}^*)$
    if $\|E_{I-1} - E_I\| < \epsilon_{icp}$ then
        terminate
    end if
end for
```

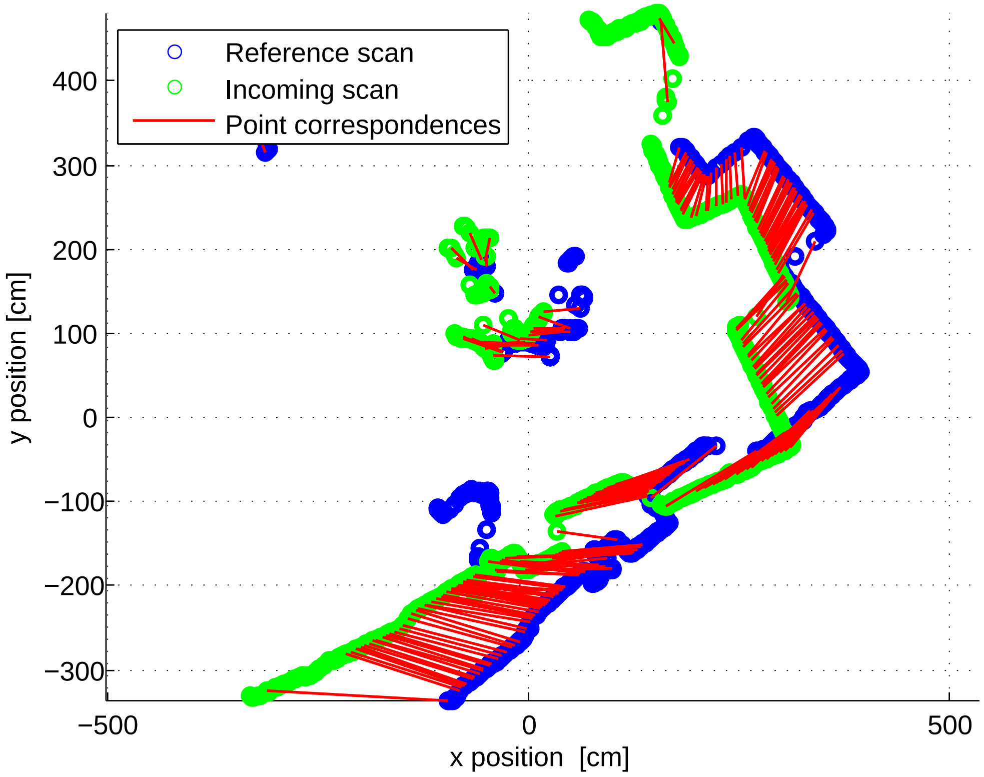


Figure 2.1: Scan registration using ICP with point correspondences shown.

and the scene scan. In practice, perfect overlap is an incorrect assumption, as due to sensor noise and vehicle movement, even scans taken from the same vehicle position will not overlap perfectly. Instead, a laser scanner can be seen as a sensor which samples the surface locally, and the scan registration algorithm can explicitly take this information into account.

One attribute of local surface structure which can aid in the scan registration process is the surface normal. In practice, surface normals are estimated using principle component analysis (PCA) for a neighbourhood of points. The point-to-plane ICP algorithm somewhat addresses the issue of local point sampling, by modelling the point cloud locally as planes using the surface normal, then minimizing the distance of a point in the scene scan to its associated plane in the reference scan. The point-to-plane approach however, does not take into account any local surface information of the scene scan, and it further assumes that the points in the scene scan are noiseless.

The G-ICP algorithm can be seen as a plane-to-plane matching algorithm. For G-ICP, it is assumed that every point in the point clouds of the reference and scene scan have been sampled from a Gaussian distribution. The Gaussian distribution has a covariance such that there is confidence about the point's distance from the surface along the normal direction, but less confidence about where the point lies along the surface. The Gaussian distribution is generated using the G-ICP surface covariance matrix $\Sigma_\varepsilon \in \mathbb{R}^{d\times d}$, where for d-dimensional points, the matrix Σ_ε has a small value, ε, as the first element, a value of one for the remaining elements along the diagonal, and zero for all other elements. In order to capture the surface information in the correct orientation, the G-ICP surface covariance matrix is rotated according to the surface normal directions for corresponding points of the scene and reference scan. The surface covariance matrix is given as

$$\Sigma_\varepsilon = \begin{bmatrix} \varepsilon & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \tag{2.6}$$

where $\varepsilon < 1$. For the G-ICP algorithm, the process of using nearest neighbours for point correspondences is used. By doing so, all of the well developed methods for nearest neighbour look-up such as kd-trees can also be used for G-ICP. For the i^{th} pair of corresponding points, denote the residual vector for the point in the reference scan, $p_i^R \in \overline{P^R}$, and the point in the scene scan $p_i^S \in \overline{P^S}$, as

$$\alpha_i = p_i^R - T_{\mathcal{E}}(p_i^S) \tag{2.7}$$

Let n_i^R and n_i^S be the surface normal for the i^{th} point correspondence from the reference and scene scan, respectively. Denote the function $\mathcal{R} : \mathbb{R}^d \mapsto \mathbb{R}^{d\times d}$, which accepts the surface normal and returns a rotation matrix that provides the rotation between the co-ordinate frame of the scan and the surface normal direction. Then, denote the rotated G-ICP surface covariance matrices, $\mathcal{S}_i^R$ and $\mathcal{S}_i^S$, for the reference scan and scene scan, respectively as

$$\mathcal{S}_i^R = \mathcal{R}(n_i^R)\Sigma_\varepsilon \mathcal{R}(n_i^R)^T$$
$$\mathcal{S}_i^S = \mathcal{R}(n_i^S)\Sigma_\varepsilon \mathcal{R}(n_i^S)^T$$

Denoting $\mathcal{E}_r$ as the rotation matrix constructed from the rotation parameters from parameter estimate $\mathcal{E}$, the G-ICP score function is given as

$$\Lambda_{gicp}(\mathcal{E}) = \sum_{i=1}^{M_{\mathcal{E}}} {\alpha_i}^T (\mathcal{S}_i^R + \mathcal{E}_r \mathcal{S}_i^S {\mathcal{E}_r}^T)^{-1} \alpha_i$$

The G-ICP score function can be used in place of the ICP score function when performing G-ICP based scan registration using a procedure similar to Algorithm 1. Intuitively, the G-ICP score function penalizes the divergence between the Gaussian distributions modelling the surface at corresponding points in the scene and reference scan. For a given point, the G-ICP algorithm models an associated Gaussian distribution such that there is high uncertainty in the point location perpendicular to the surface normal direction and low uncertainty in the location of the point in the normal direction. Through distribution to distribution matching, the G-ICP algorithm takes the sensor noise associated with both the reference and scene scan into account, whereas the point to plane ICP algorithm accounts for the sensor noise in only the reference scan. It is also evident that the G-ICP algorithm requires accurate surface normal information in order to generate the Gaussian distribution associated with each point. It should be noted that in practice, surface normal information is difficult to compute, especially in outdoor environments where trees, shrubs and bushes are present.

2.4 The Normal Distributions Transform Algorithm

The Normal Distributions Transform (NDT) is a method by which sections of a point cloud are represented as Gaussian distributions within a grid structure. The transform maps a point cloud to a smooth surface representation described as a set of local distributions which capture the shape of the surface. Similar to an occupancy grid map, the NDT generates a subdivided representation of the environment. However, the occupancy grid represents the probability that the cell is occupied, whereas the NDT represents the probability of measuring a point cloud sample for each position within a given cell.

The NDT registration algorithm begins by subdividing the space occupied by the reference scan into fixed size, rectilinear grid cells, $c \subseteq \mathbb{R}^d$, where the set of all cells in the reference scan is denoted as C^R. Denote the collection of the reference scan points in cell c_i as $P^R_{c_i} = \{p \in P^R : p \in c_i\}$. Given the i^{th} cell, the points of the reference scan occupying that cell are used to generate a mean, μ_i, and a covariance matrix, Σ_i, for a representative Gaussian distribution, $\mathcal{N}(\mu_i, \Sigma_i)$. The Gaussian distribution can be interpreted as a generative process that models the local surface points $P^R_{c_i}$ within the cell. In other words, it assumes that $P^R_{c_i}$ is generated by drawing from the Gaussian distribution $\mathcal{N}^R_i(\mu^R_i, \Sigma^R_i)$. Assuming that the locations of the reference scan surface points are drawn from this distribution, the likelihood of having measured a point p, within cell c_i, can be modelled as

$$\rho_{c_i}(p) = \exp\left(-\frac{(p - \mu_i^T)\Sigma_i^{-1}(p - \mu_i)}{2}\right) \tag{2.8}$$

A sample scan transformed by the NDT and a corresponding probability distribution plot is given in Figures 2.2 and 2.3.

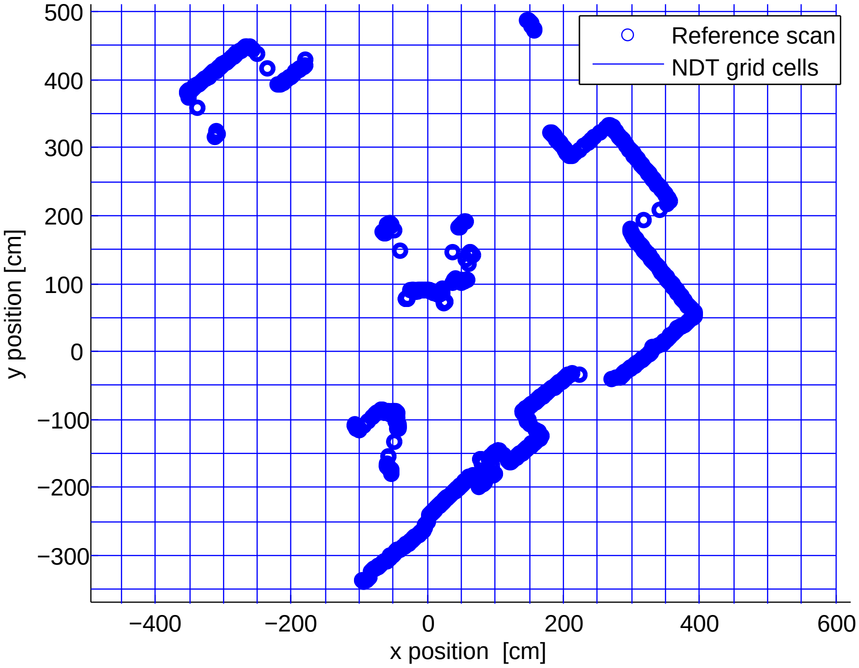


Figure 2.2: Sample 2D LIDAR scan with a 50cm by 50cm NDT grid.

Since the point cloud is modelled by a piecewise continuous and piecewise differentiable summation of Gaussians, numerical optimization tools can be used in order to register the scene scan with the reference scan. A fitness score can be calculated which quantifies the measure of overlap between the reference scan and the scene scan transformed by parameter estimate $\mathcal{E}$. The original work by Magnusson for point to distribution NDT registration calculates the cost by evaluating each point in the transformed scan at the distribution corresponding to the NDT cell which each transformed point occupies [34]. The NDT point-to-distribution (p2d) score is given as

$$\Lambda_{ndt}^{p2d}(\mathcal{E}) = -\sum_{p \in P^S} \rho_{c_{cur}}(T_{\mathcal{E}}(p)) \tag{2.9}$$

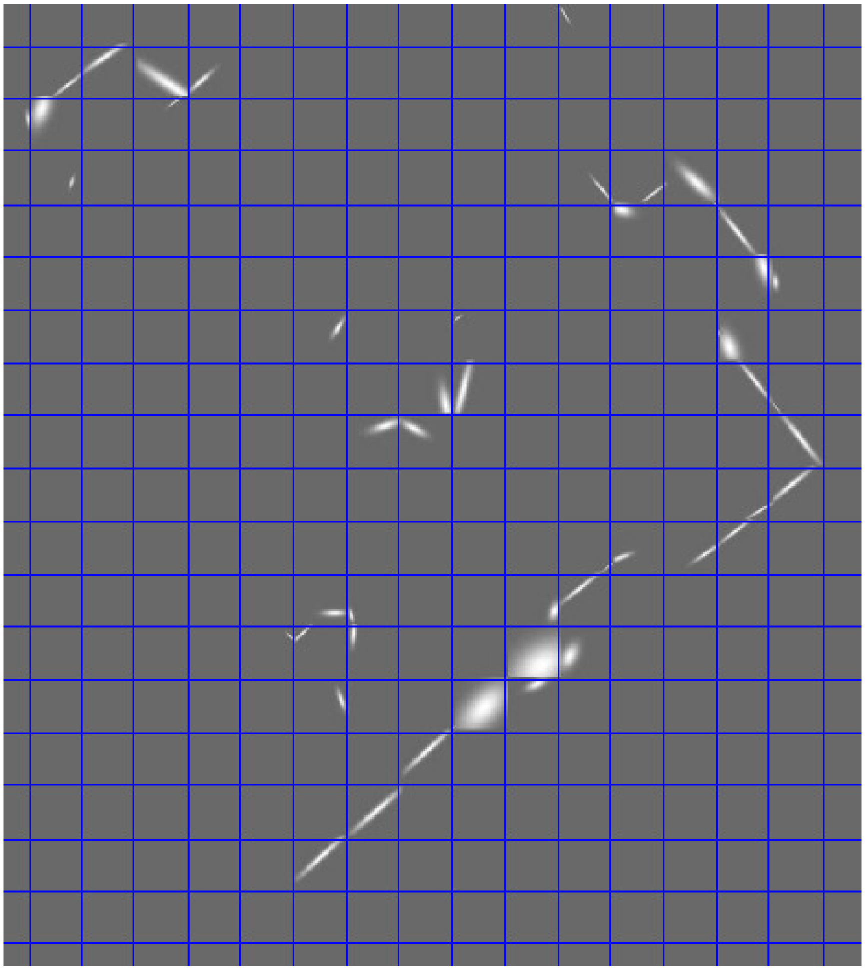


Figure 2.3: Probability density of the laser scan from NDT for each cell. Lighter areas indicate higher probabilities of sample points

where c_{cur} denotes the cell which point p occupies when transformed by parameter estimate $\mathcal{E}$. Stoyanov et al. introduced a modified version of NDT which generates an NDT representation of both scans, then compares the distributions of the scans in order to align them [54]. Thus, the scene scan P^S is also divided into a set of cells, C^S, where the points contained in the i^{th} cell are used to model the distribution $\mathcal{N}_i^S(\mu_i^S, \Sigma_i^S)$. To align the scans, an optimization is formulated to minimize the L_2-distance between the sets of Gaussian distributions of the reference and scene scan. Denote the difference of mean values for the i^{th} Gaussian distribution from the reference scan and the j^{th} Gaussian distribution of the scene scan as

$$\beta_{ij} = \left[T_{\mathcal{E}}(\mu_j^S) - \mu_i^R\right]$$

Using the difference of means, the NDT distribution-to-distribution (d2d) score function is given as

$$\Lambda_{ndt}^{d2d}(\mathcal{E}) = -\sum_{i=1}^{|C^R|}\sum_{j=1}^{|C^S|} \exp(-\frac{1}{2}\beta_{ij}^T \left[\mathcal{E}_r^T \Sigma_j^S \mathcal{E}_r + \Sigma_i^R\right]^{-1} \beta_{ij}) \tag{2.10}$$

The distribution to distribution cost function evaluates all pairwise Gaussian components for both the scene and the reference scan. In Stoyanov's work however, the cost function is only evaluated at the nearest Gaussian component in order to reduce computation. Evaluation at the nearest Gaussian results in discontinuities in the overall cost function, which implies that the cost function is non-smooth and the gradient and Hessian do not exist in the transformation space when the correspondences between mean points change. The issue of the non-smooth and discontinuous cost function is also present in point-to-distribution NDT, when scene scan points cross cell boundaries. Magnusson et al. address the discontinuous cost function using a tri-linear interpolation scheme which takes distributions from neighbouring cells into account when calculating the gradient and Hessian contributions [34]. However, the tri-linear interpolation does not fully solve the issue, as the cost functions are still discontinuous.

It should be noted that it is possible to compute an analytic gradient, g, and Hessian H, for both NDT cost functions, which can be used for improving the performance of the nonlinear optimization method selected. For point to distribution matching, denote the functions $\Pi_g(p_k, \mathcal{E})$, and $\Pi_H(p_k, \mathcal{E})$, which use a point in the scene scan and the transformation parameters to calculate an associated gradient and Hessian contribution for point p_k. The optimization is initialized from a parameter estimate $\hat{\mathcal{E}}$, and terminates once the norm of the gradient, $\|g\|$, is less than a user specified threshold, ϵ_{ndt}. Biber et al. use a Newton method with line search to perform the optimization [5], however other methods such as Levenberg-Marquardt or Broyden-Fletcher-Goldfarb-Shanno (BFGS) method could be used. The NDT point to distribution registration algorithm is summarized in Algorithm 2. The NDT distribution to distribution registration algorithm is performed analogous to Algorithm 2, except the NDT is performed for both the scene and reference scan, and the distribution to distribution cost function given by Equation 2.10 is used in place of Equation 2.9.

2.5 K-means Clustering

K-means clustering is a well known data partitioning technique [31] which, given a set of data points, seeks to divide the data points k sets, $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_k\}$. The associated clustering metric is the squared distance between the points in the data set and their associated closest cluster mean, μ_i. K-means seeks to divide the data points into k sets such that the function

$$f_{km} = \sum_{i=1}^{k} \sum_{p \in \gamma_i} \|p - \mu_i\|^2 \tag{2.11}$$

is minimized. The k-means algorithm is typically performed using an iterative method and has two main steps. Given a set of initial k mean locations, the first step assigns each of the data points in point cloud, P, to the cluster with the closest mean value. In the second step, the means are re-calculated as the mean value of the data points assigned to each cluster. These steps are repeated until the change in the vector of mean values, $\mathcal{M}$, between t and $t-1$ iterations, δ, falls below a certain threshold, ϵ_{km}. The k-means algorithm for clustering is summarized in Algorithm 3.

Algorithm 2 Register scene scan P^S to reference scan P^R using point to distribution NDT

```
Initialization of NDT:
allocate NDT cells $C^R$, for reference scan $P^R$
for all $c_i \in C^R$ do
    $P^R_{c_i} = \{p \in P^R : p \in c_i\}$
    $\mu_i \leftarrow \frac{1}{|P^R_{c_i}|} \sum_{p \in P^R_{c_i}} p$
    $\Sigma_i \leftarrow \frac{1}{|P^R_{c_i}|-1} \sum_{p \in P^R_{c_i}} (p - \mu_i)(p - \mu_i)^T$
end for
Registration of scene scan with the reference NDT:
$\mathcal{E} \leftarrow \hat{\mathcal{E}}$
$I \leftarrow 0$
while $| g | > \epsilon_{ndt}$ and $I < I_{max}$ do
    $s \leftarrow 0$
    $g \leftarrow 0$
    $H \leftarrow 0$
    for all $p \in P^R$ do
        $\bar{p} \leftarrow T_{\mathcal{E}}(p)$
        $c_{cur} = \{c \in C^R : \bar{p} \in c\}$
        $s \leftarrow s + \rho_{c_{cur}}(\bar{p})$
        $g \leftarrow g + \Pi_g(\bar{p}, \mathcal{E})$
        $H \leftarrow H + \Pi_H(\bar{p}, \mathcal{E})$
    end for
    $\Delta\mathcal{E} = (H)^{-1}(-g)$
    $\mathcal{E} \leftarrow \mathcal{E} + \Delta\mathcal{E}$
end while
$\mathcal{E}^* \leftarrow \mathcal{E}$
```

Algorithm 3 Algorithm for k-means clustering of point cloud P

```
Randomly choose $k$ mean locations:
$\mathcal{M}_0 \leftarrow \{\mu_1^0, \mu_2^0, \ldots, \mu_k^0\}$
while $\delta > \epsilon_{km}$ do
    Perform assignment step:
    for all $\gamma_i \in \Gamma$ do
        $\gamma_i \leftarrow \{p \in P : \|p - \mu_i^t\| \leq \|p - \mu_j^t\| \ \forall \ j \in \{1, \ldots, k\}$
    end for
    Perform mean update step:
    for all $i \in \{1, \ldots, k\}$ do
        $\mu_i^t \leftarrow \frac{1}{|\gamma_i|} \sum_{p \in \gamma_i} p$
    end for
    Check for convergence:
    $\mathcal{M}_t \leftarrow [\mu_1^t \ \mu_2^t \ \ldots \ \mu_k^t]^T$
    $\delta \leftarrow \|\mathcal{M}_{t-1} - \mathcal{M}_t\|$
end while
```

Chapter 3

Multi-Scale K-Means NDT

The NDT approach in 2D seeks the transform parameters x, y and θ such that the transformed scene scan is best aligned with the reference scan according to the NDT cost function. The 2D NDT algorithm divides the reference scan of the environment using a rectilinear grid and models the points contained in each grid cell as a Gaussian distribution, which results in a nonlinear optimization that is not guaranteed to converge, and is susceptible to local minima if not initialized sufficiently close to the minimum.

To demonstrate the weak convergence characteristics of the standard NDT algorithm, a projection of the 3D cost function onto the x and θ plane for NDT is presented in Figure 3.1, which illustrates numerous local minima of the NDT cost function surrounding the global minimum. It is clear that if the optimization is not initialized close to the global minimum, the optimization may converge to a local minimum. In this chapter, a method to improve the convergence basin of NDT, the Multi-Scale K-Means NDT (MSKM-NDT), is presented. The grid based division of the scan is replaced with a multi-scale k-means clustering technique. The number of clusters is selected such that, as the registration scale decreases, the number of cluster increases. In order to increase the basin of convergence, the optimization step is performed across varying scales, as the optimization for the current scale is initialized with the solution from the previous scale. The multi-scale method avoids local minima and is experimentally shown to reliably converge to the global minimum for large initial transformation errors. Experimental results also demonstrate that the MSKM-NDT algorithm can be used in a SLAM scenario to accurately reconstruct the environment map and vehicle path from partially overlapping laser scans.

Another issue with the standard NDT formulation is that arbitrarily small changes in the parameter estimate could cause the scene scan points to cross cell boundaries, resulting in discontinuities in the overall cost function. As a result, the cost function is non-smooth and the gradient and Hessian do not exist in the transformation space at cell boundaries. To remove the discontinuities with the standard NDT formulation, every point of the scene scan would have to be evaluated at the Gaussian distribution of *every* grid cell, which is generally not computationally feasible as even modestly sized environments results in many grid cells, depending on the discretization resolution. Magnusson et al. address the discontinuities using a tri-linear interpolation scheme which takes the influence of neighbouring cells into account when calculating the gradient and Hessian contribution from a single scene point [36]. However, the interpolation approach merely reduces, but does not entirely resolve the issue of cost function discontinuity, as a boundary crossing could still cause a finite change in the cost function for an arbitrarily small change in the transformation parameters. Thus, even with the use of tri-linear interpolation, the cost function remains discontinuous and therefore convergence of the optimization is not guaranteed. The MSKM-NDT approach, on the other hand, performs k-means clustering of the scan, generating comparatively fewer Gaussian distributions compared to the grid based approach. Since fewer Gaussian distributions are generated, the MSKM-NDT approach can feasibly evaluate the points of the scene scan at every distribution for a given scale, which removes discontinuities in the cost function caused by the cell boundaries found in the standard NDT algorithm. Evaluation of the scene scan points at all Gaussian distributions guarantees that the nonlinear optimization is well defined and all standard nonlinear optimization algorithms will converge to a solution satisfying necessary conditions for optimality.

3.1 Proposed Method: Multi-Scale K-Means NDT (MSKM-NDT)

In order to overcome the poor convergence basin of the NDT algorithm, a multi-scale registration approach is proposed. The proposed approach uses k-means clustering to cluster the scan at different scales and perform the optimization at successively decreasing scales. The k-means clustering results in a smooth cost function and removes the discontinuities in the gradient and Hessian associated with the standard NDT algorithm. The multi-scale optimization improves the convergence basin with a course-to-fine approach that avoids local minima. The standard NDT algorithm is modified in the following ways:

1. The grid based approach in NDT for the division of the scan is replaced with the k-means algorithm for clustering the reference scan, P^R, into k clusters. Let the

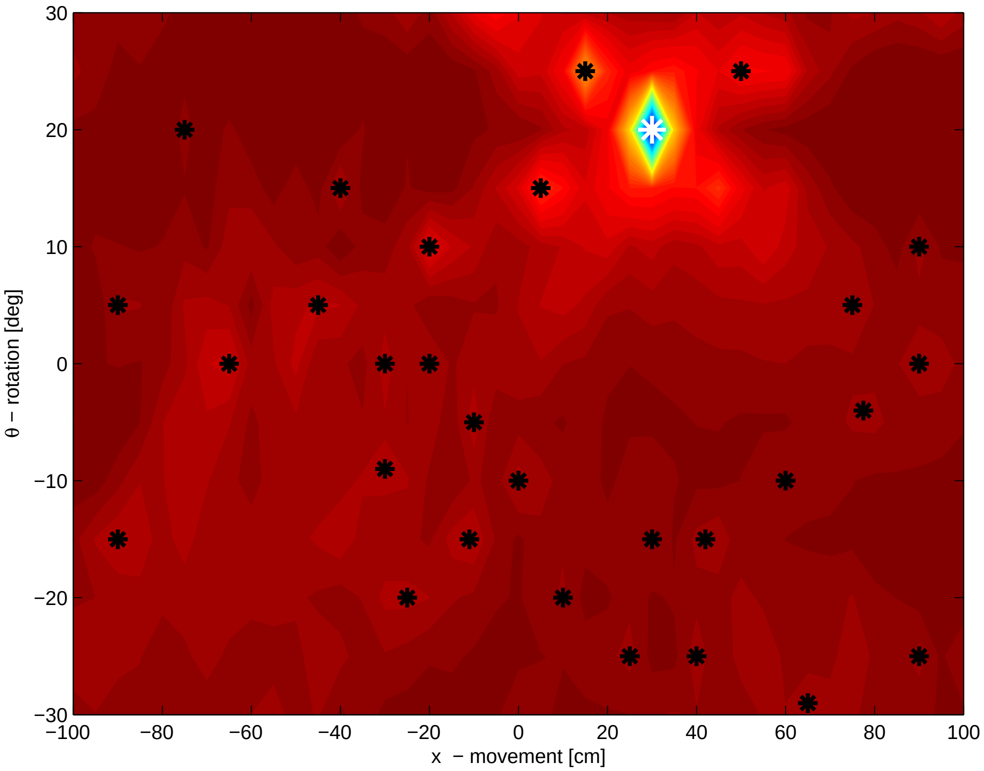


Figure 3.1: Cost surface for NDT registration, for varying x and θ parameters. The black markers denote the local minima and the white marker denotes the global minimum. The true transform parameters are $x = 30$ cm and $\theta = 20$ deg.

set of all clusters be $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_k\}$, where each cluster has N_{γ_j} points and $\sum_{j=1}^{k} N_{\gamma_j} = |P^R|$. Where standard NDT models the distribution of points within each cell as a Gaussian distribution, MSKM-NDT models each k-means cluster as a Gaussian distribution.

2. The standard NDT algorithm performs the optimization step for one fixed grid spacing, while MSKM-NDT performs the optimization step at decreasing scales, increasing the number of clusters as the scale is decreased. Each scale represents the number of clusters the reference scan is divided into, using k-means. The set of scales, $\Phi = \{\phi_1, \phi_2, \ldots, \phi_N\}$, is selected in descending order, resulting in a course-to-fine optimization scheme which improves convergence to the global minimum of the cost function.

3. To accommodate the clustering of the reference scan, the standard NDT cost calculation is modified. Standard NDT evaluates the Gaussian distribution within the cell that a transformed point, $T_{\mathcal{E}}(p)$, falls into. For MSKM-NDT, each transformed point is evaluated with respect to all cluster distributions in the current scale, ϕ_i, with the strongest contributions coming from the most likely clusters. For each cluster in the current scale, the transformed point, $\bar{p}$, is evaluated using the score function associated with the current cluster, γ_j, denoted as

$$\rho_{\gamma_j}(\bar{p}) = \exp\left(-\frac{(\bar{p} - \mu_j^T)\Sigma_j^{-1}(\bar{p} - \mu_j)}{2}\right) \tag{3.1}$$

   where μ_j and Σ_j are the mean and covariance for cluster γ_j, respectively. Evaluation of the point at all of the clusters of the current scale results in a continuous and differentiable cost function at each stage of the multi-scale optimization.

An example of a reference scan clustered using k-means at three different scales is presented in Figures 3.2, 3.3 and 3.4. The red ellipses represent one standard deviation for the Gaussian distributions of the cluster points. The figures demonstrate that clustering at larger scales captures very coarse features within the scan and clustering at smaller scales captures the finer details. It is this phenomenon which MSKM-NDT exploits. At smaller scales, the cost function has a wider basin of convergence, however due to the coarseness of the clustering, the minimum at that scale may not correspond to the true transformation parameters. At higher scales, the cost function has a much narrower basin of convergence, however due to the details captured at higher scales, the global minimum is likely to correspond to the true transformation parameters. An example of the cost surface projection over multiple scales is depicted in Figure 3.5. In the multi-scale approach, the optimization for a scale, ϕ_i, is initialized with the parameter solution from the previous scale, ϕ_{i-1}. The end result is a method which effectively widens the basin of convergence for the NDT algorithm, making it more robust to poor initializations. The MSKM-NDT algorithm is summarized in Algorithm 4.

3.2 Experimental Results

To validate the MSKM-NDT approach, two sets of experiments are performed. In the first experiment, SLAM is performed in an indoor environment using a series of 2D laser scans and three scan registration techniques: ICP, NDT and MSKM-NDT. The environment

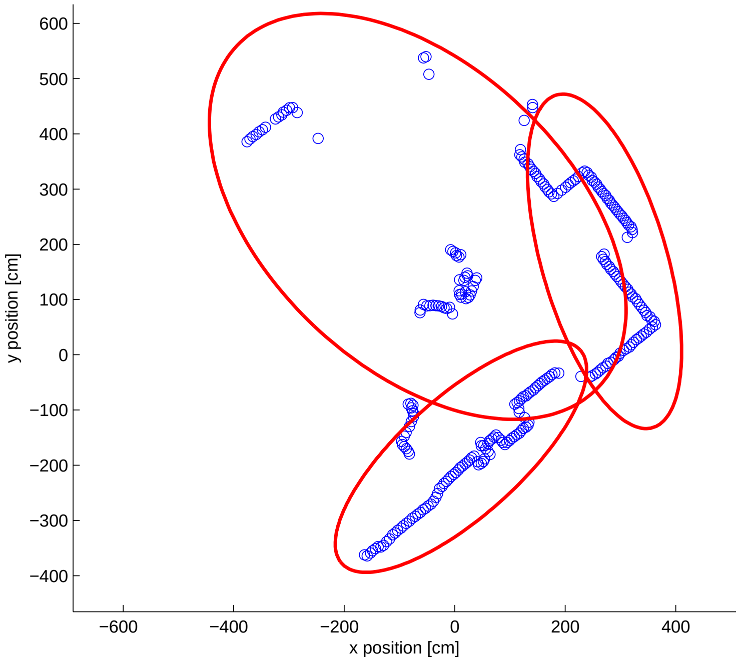


Figure 3.2: Reference scan clustering at scale $\phi = 3$

map and resulting vehicle motion is determined through pair-wise scan registration SLAM, where the entirety of the map and vehicle motion is determined through the integration of incremental pose updates provided through scan registration of successive laser scans. To validate scan registration accuracy, the experiment is performed using the three scan registration techniques. The resulting maps are compared qualitatively based on knowledge of the indoor environment, and also based on the resulting vehicle motion ground truth provided by an indoor positioning system. To demonstrate the convergence properties of the MSKM-NDT algorithm compared to the standard NDT algorithm, a second experiment is performed where a known relative transformation between a pair of sequential laser scans is applied and compared to the transform parameters resulting from a given scan registration algorithm. The initial relative transformation applied to each scan pair is systematically varied such that scan registration is attempted over a wide range of initial parameter errors, resulting in a pattern for the convergence basin of each tested algorithm.

To collect data for the experiment, a *Hokuyo UTM-30LX* LIDAR with a 40Hz update rate is mounted to a *Clearpath Robotics A100 Husky* chassis and driven through the environment, which is approximately 5m by 6m in size. Although the LIDAR scan range is 30m, the view of the entire room is occluded due to large obstacles present in the centre of

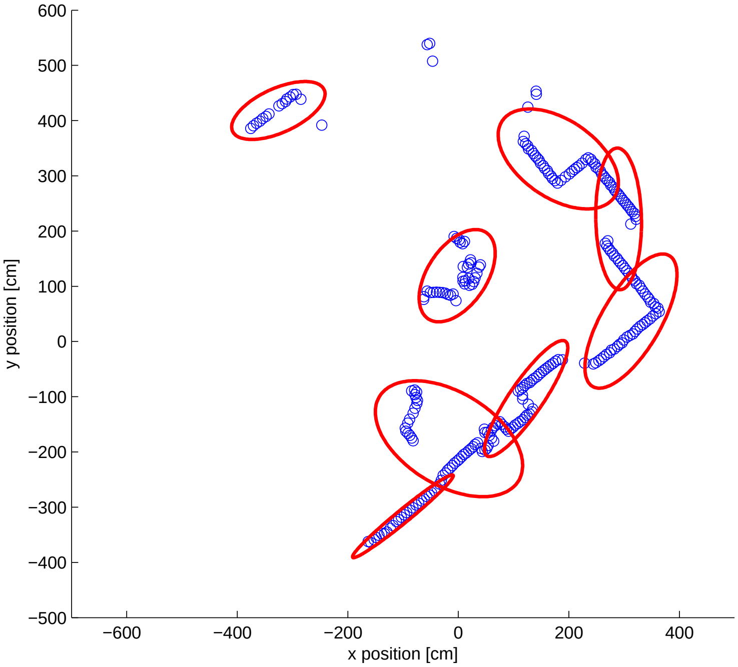


Figure 3.3: Reference scan clustering at scale $\phi = 9$

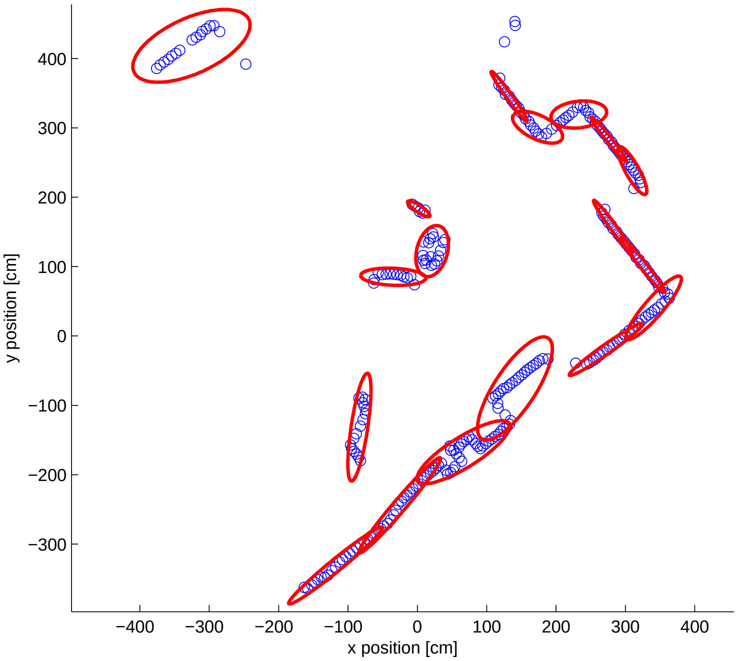


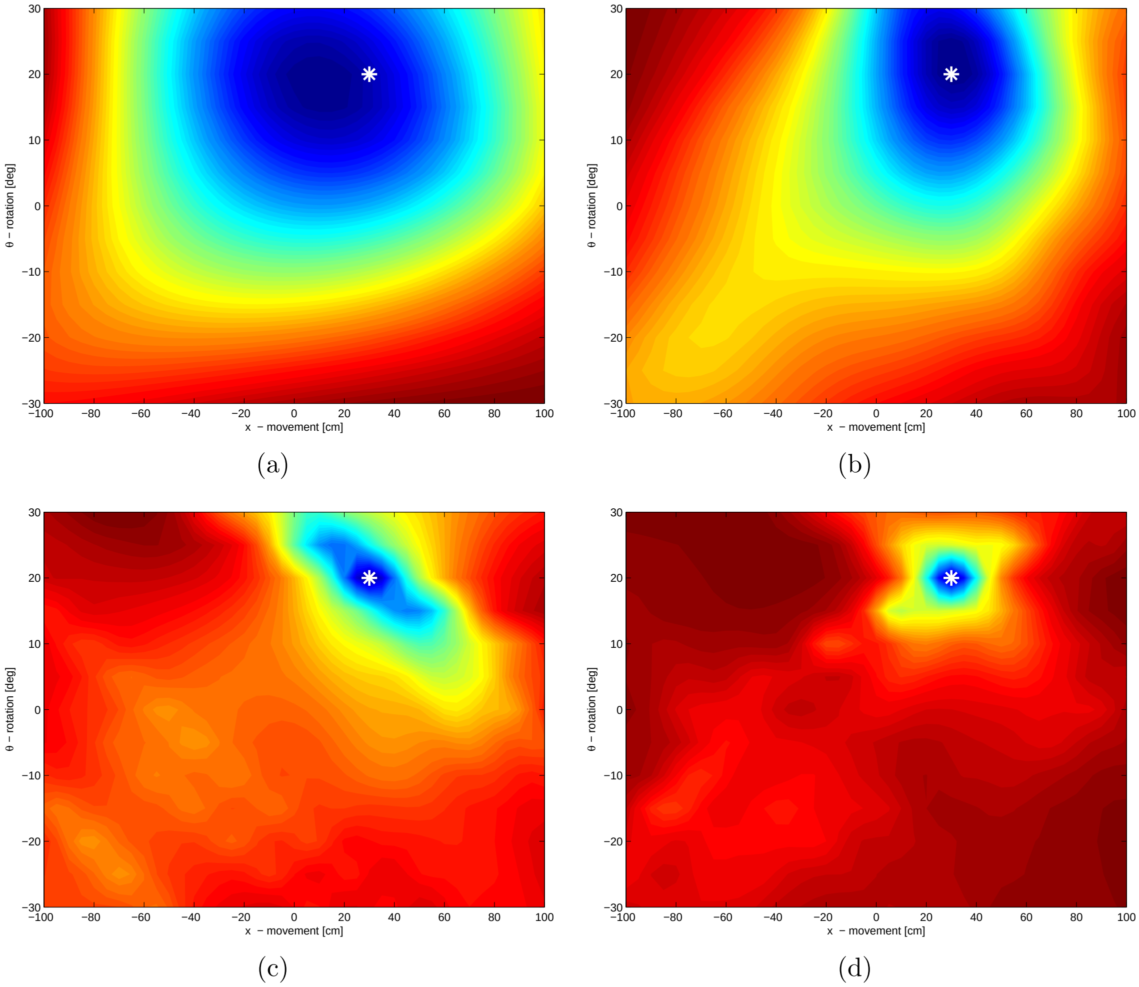
Figure 3.4: Reference scan clustering at scale $\phi = 20$

the room. Vehicle motion ground truth is collected with an OptiTrack indoor positioning system (IPS) that is capable of millimeter translation accuracy and sub degree rotation accuracy.

Although comparison of the proposed method against G-ICP registration is highly desirable in the SLAM scenario, the G-ICP algorithm is currently implemented for 3D point clouds only, and a full implementation of 2D G-ICP and comparison against the MKSM-NDT approach is left as an area of future work.

3.2.1 SLAM Experiments

For the first set of experiments, an indoor environment is reconstructed from a series of overlapping 2D laser scans, taken from multiple poses in an indoor lab environment, using pair-wise SLAM. The ICP, NDT and MSKM-NDT algorithms are compared to one another for qualitative correctness of the resulting map, and the RMS error of the vehicle motion over its trajectory. Note that for the experiment, no vehicle odometry is used and the registration is instead initialized with a parameter estimate of $\hat{\mathcal{E}} = \{0, 0, 0\}$. The resulting maps and vehicle motion are illustrated in Figures 3.6, 3.7 and 3.8, respectively.



(a)

(b)

(c)

(d)

Figure 3.5: Contour plots for cost surfaces with differing numbers of clusters. Contour plots of cost surfaces with (a) 3 clusters. (b) 6 clusters. (c) 9 clusters. (d) 15 clusters. This example shows the cost functions for two scans which have only been transformed in x and θ. The white marker indicates the true transformation parameters, $x = 30$ and $\theta = 20$. Using successive optimizations at decreasing scales, the overall optimization of the cost function avoids the local minima and converges to the global minimum.

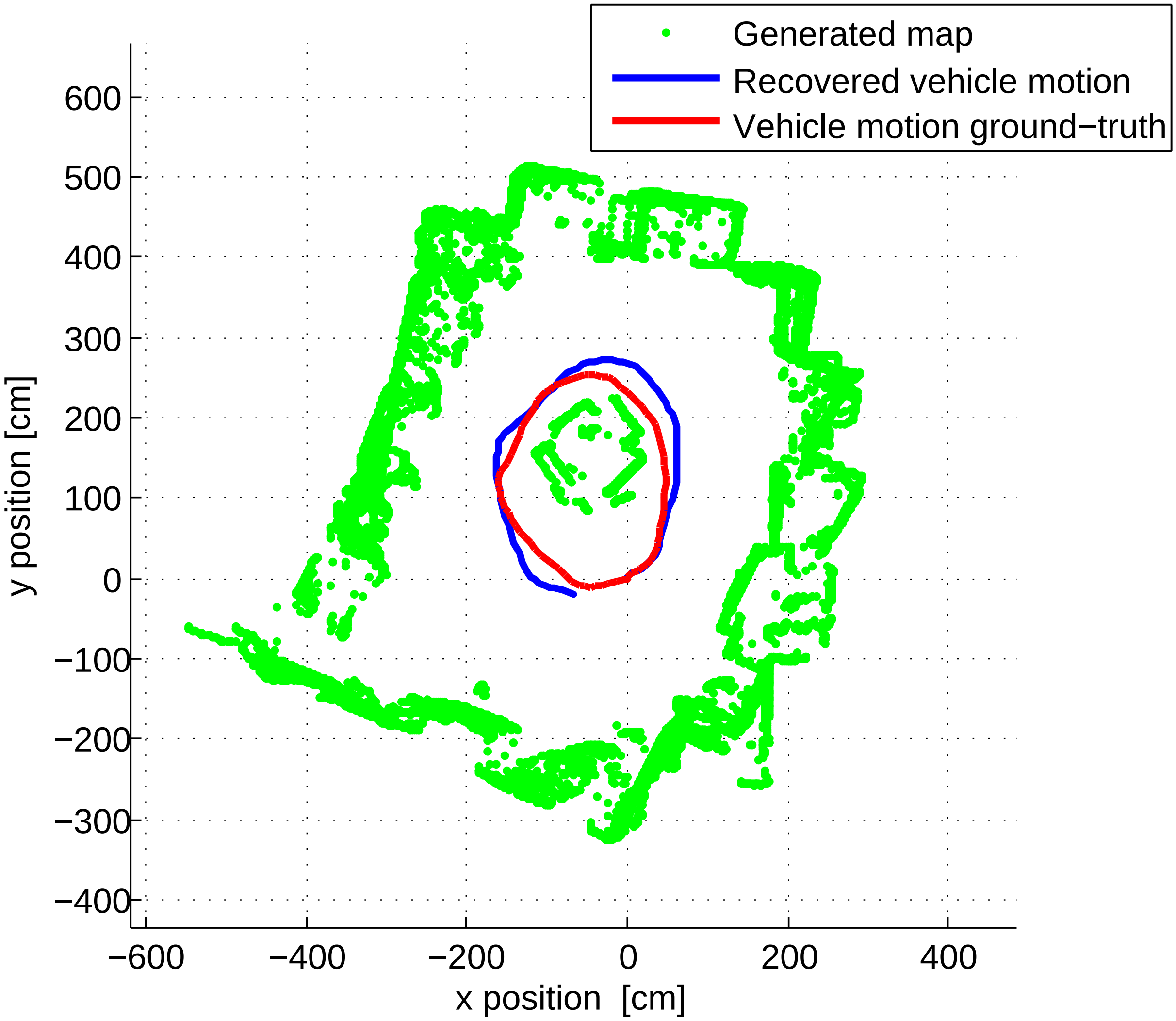


Figure 3.6: Map and vehicle motion using the ICP algorithm for a pair-wise registration SLAM solution. Note the drifting of the map due to the accumulation of registration error.

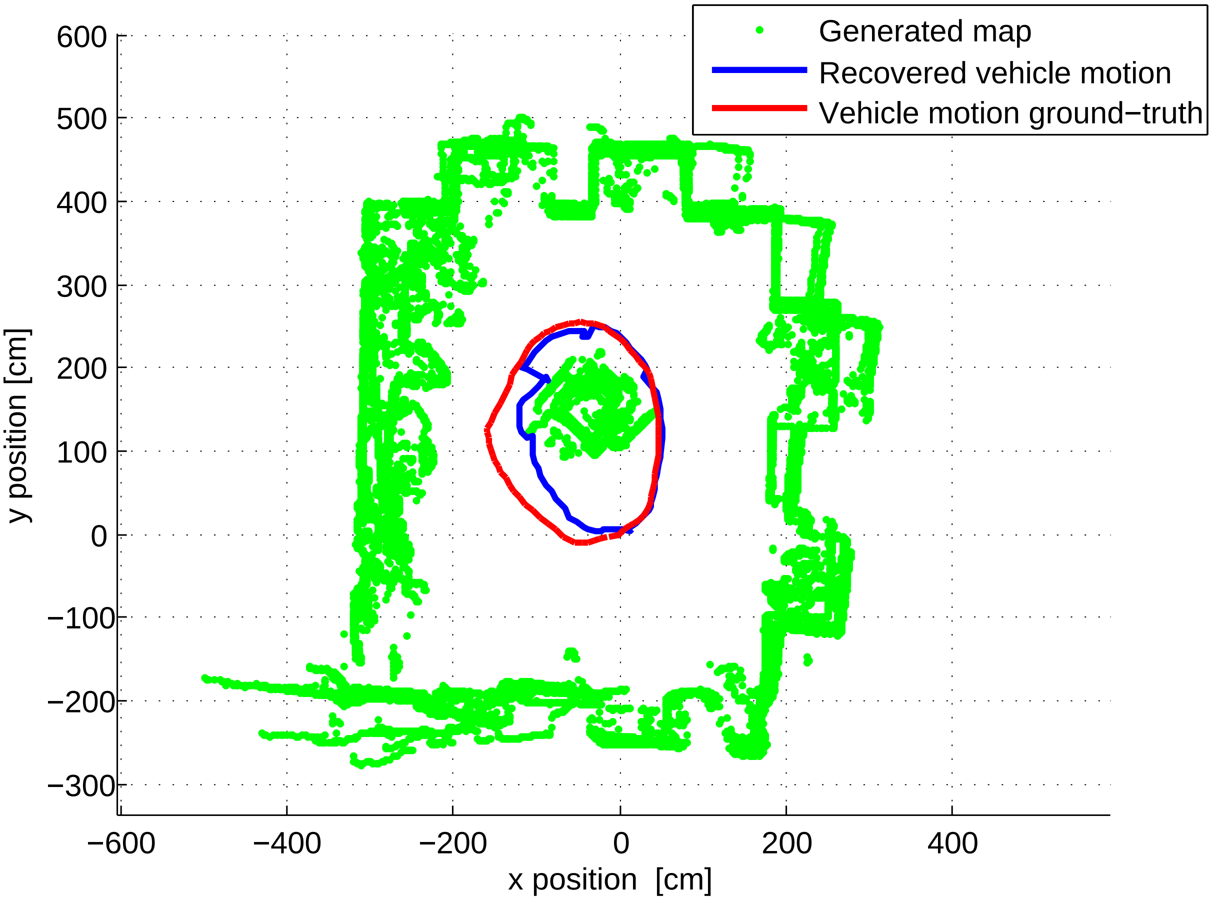


Figure 3.7: Map and vehicle motion using the NDT algorithm for a pair-wise registration SLAM solution. Note the poor map quality due to the NDT algorithm converging to local minima.

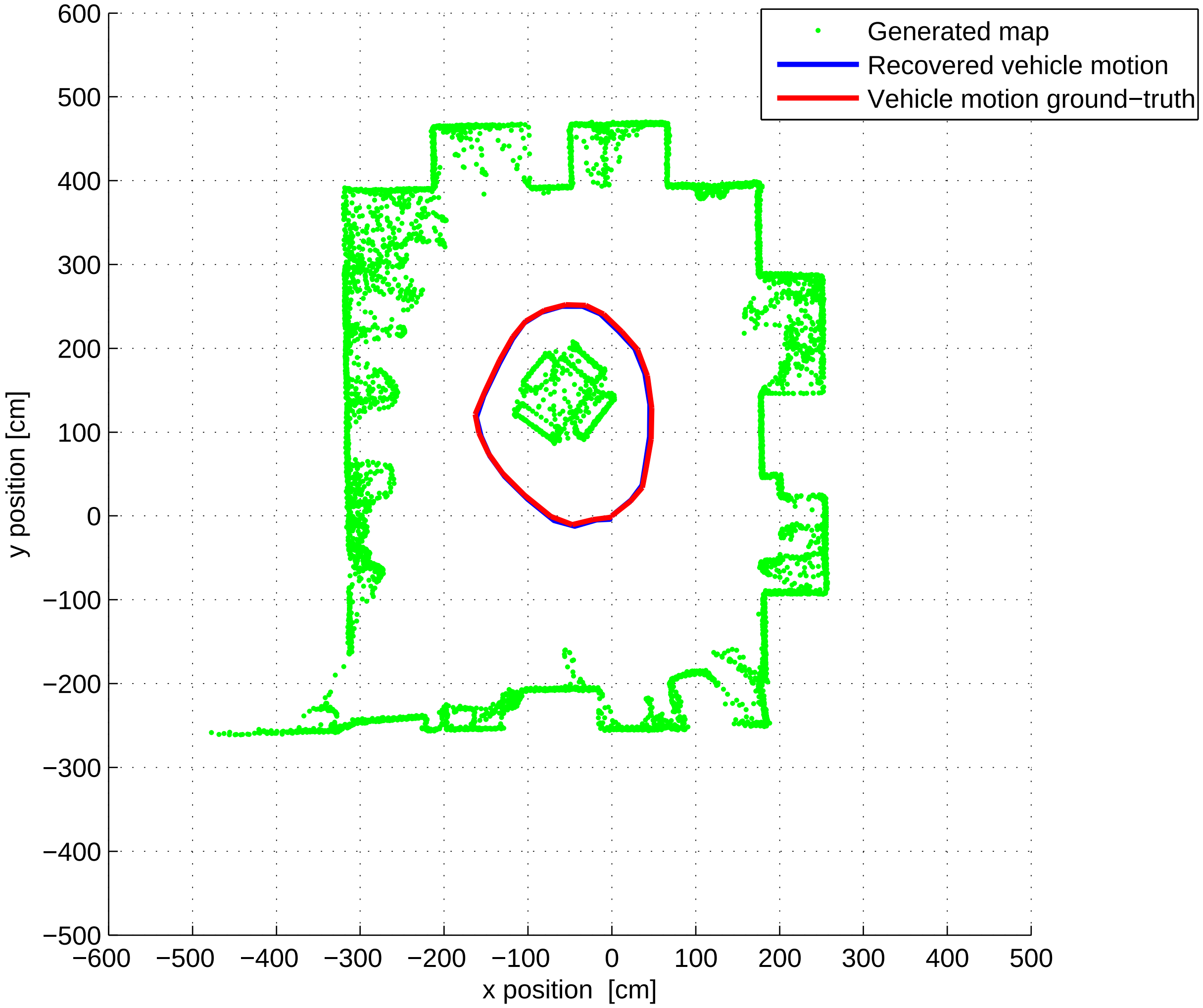


Figure 3.8: Map and vehicle motion using MSKM-NDT algorithm for a pair-wise registration SLAM solution. Note the noise in some of the corners of the room due to the LIDAR beam glancing between edges of objects and registering false readings. The MSKM-NDT algorithm is able to generate a high quality map, even with this noise present.

It can be seen that the structure of the room, such as the corners and walls, are better represented using the MSKM-NDT approach, and the recovered map using MSKM-NDT shows accurate scan registration, as verified by the strong overlap between the recovered vehicle path and the IPS ground truth path. In contrast, the ICP based SLAM is prone to drift over time due to the accumulation of pairwise registration errors, while the NDT SLAM solution produces a warped map due to the optimization converging at local minima. The RMS position error for the MSKM-NDT is significantly less compared to the other approaches, as the RMS position error over the experiment is 38.47 cm using ICP SLAM, 9.48 cm using the NDT SLAM, and 1.16 cm using MKSM-NDT SLAM.

To further illustrate the robustness of the MSKM-NDT algorithm in a SLAM scenario, a similar experiment with the same data set is performed, however, using *every 50th* scan in the sequence of laser data, as opposed to the previous experiment where every scan was used. The use of every 50th scan represents a much larger initial transformation between the scans. As with the previous experiment, no vehicle odometry is used to initialize the registration and is initialized with a parameter estimate of $\hat{\mathcal{E}} = \{0, 0, 0\}$. The resulting map using the MSKM-NDT algorithm with every 50th laser scan is visually indistinguishable from the map generated using MKSM-NDT with every laser scan, and is thus not displayed. The calculated RMS error over the entire run is calculated to be approximately 1.34 cm, which is comparable to the results of the MKSM-NDT approach using every laser scan. It should be noted that the SLAM results using the ICP and NDT algorithms with every 50th laser scan are omitted, as the scan registrations failed to converge. It is evident that the MSKM-NDT method is robust to initial transformation errors, as it is able to generate accurate maps in a SLAM scenario using subsequent, partially overlapping scans, which greatly differ in pose.

3.2.2 Convergence Tests

In order to test the convergence basin of the MSKM-NDT approach, a pair of sequential scans are transformed relative to each other by a known amount and a registration algorithm is performed in order to determine the transformation parameters. If each of the the resulting transformation parameters are within 5% of the true transformation parameters, the registration is deemed successful. To test a wide case of scenarios, the scans are tested over an x-position range of -200cm to 200cm (with a 50cm resolution), a y-position range of -200cm to 200cm (with a 50cm resolution) and a rotation range of -30 degrees to 30 degrees (with a 15 degree resolution), resulting in a total of 405 combinations.

Three scan registration approaches are compared: the standard NDT algorithm (NDT), a multi-scale grid discretization NDT (MSG-NDT) and the proposed method, a multi-scale

	NDT	MSG-NDT	MSKM-NDT
Full overlap	4.9%	22.2%	94.3%
Partial overlap	3.0%	20.0%	75.9%

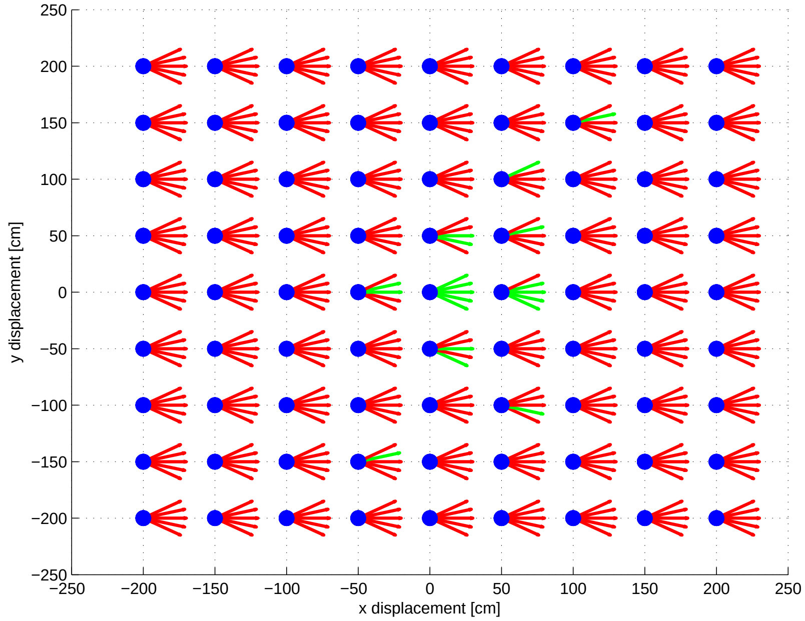
Table 3.1: Summary of success rates for registration experiments

k-means clustering NDT (MSKM-NDT). The multi-scale grid discretization approach is an intuitive scale based optimization approach suggested by Magnusson et al. [34], where the scale size corresponds to the spacing between grid cells. A large scale denotes large grid spacing, while a small scale denotes finer grid spacing. For the experiment a cluster scale setting of $\Phi = \{3, 6, 9, 15\}$ for the MSKM-NDT is selected and a grid scale setting of $\Phi = \{400, 200, 100, 50\}$ (distance between grid cells in [cm]) for MSG-NDT, is selected. The scale sizes are determined experimentally through rough testing of scale sequences that perform best for each algorithm. Although the experimental selection of the scale sequence sufficiently demonstrates the behaviour of each algorithm, a more intelligent, automated approach for scale selection is left as future work.

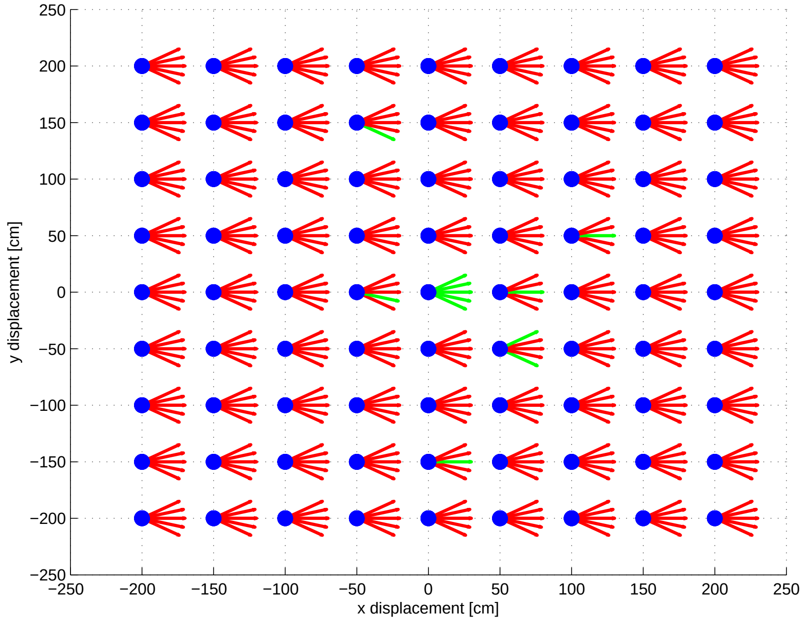
The experiment is performed for two test cases. In the first case, the two scans to be registered are fully overlapping, which is, in general, the best case scenario for scan registration. In the second case, the scans to be registered are partially overlapping, which is more realistic scenario for a mobile robotics application. The results are shown in Figures 3.9 to 3.11. For Figures 3.9 to 3.11, the blue dot indicates the x, y displacement between scans and the arrow shows the rotations which were tested at that x, y location, describing a full parameter set for the transformation. A red arrow indicates the registration failed for that parameter set, while a green arrow indicates the registration succeeded.

Table 3.1 summarizes the results of the registration tests. The NDT algorithm has the smallest basin of convergence, as it is essentially only able to converge to correct solutions for small initialization errors for both the fully and partially overlapping case. The MSG-NDT algorithm has a wider basin of convergence, but still generally fails for large translations in the x and y directions. The MSKM-NDT algorithm had the highest success rate and suggests the widest valley of convergence. Furthermore, there is no visible trend to when the MSKM-NDT algorithm fails. The failures for MSKM-NDT could be geometry specific and could be rectified by performing the registration with additional scales.

It can be seen that in general, the MSKM-NDT algorithm converges for significantly more initial parameter errors compared to the NDT or MSG-NDT algorithms. While the MSG-NDT is also a multi-scale algorithm, the grid based discretization produces rapid changes in the cost surface as points pass between cell boundaries, resulting in a discontinu-

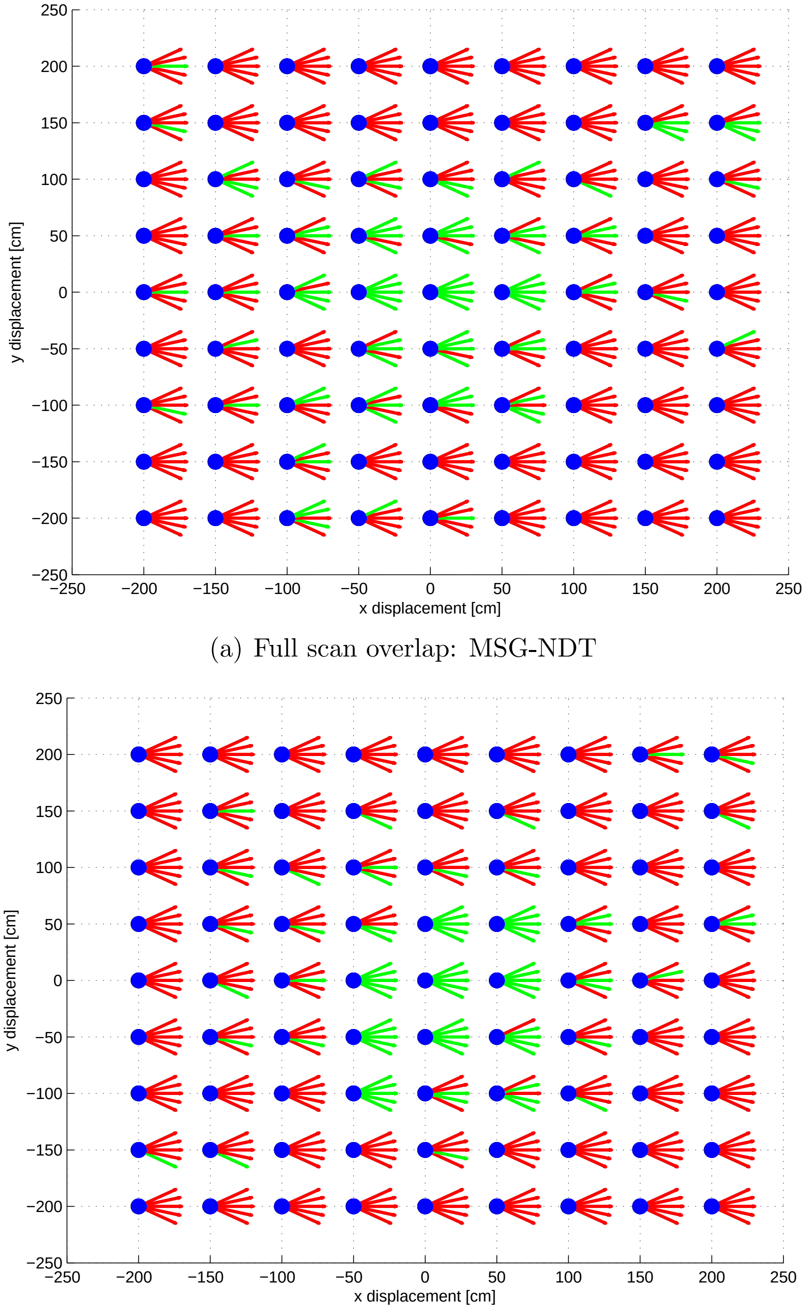


(a) Full scan overlap: NDT



(b) Partial scan overlap: NDT

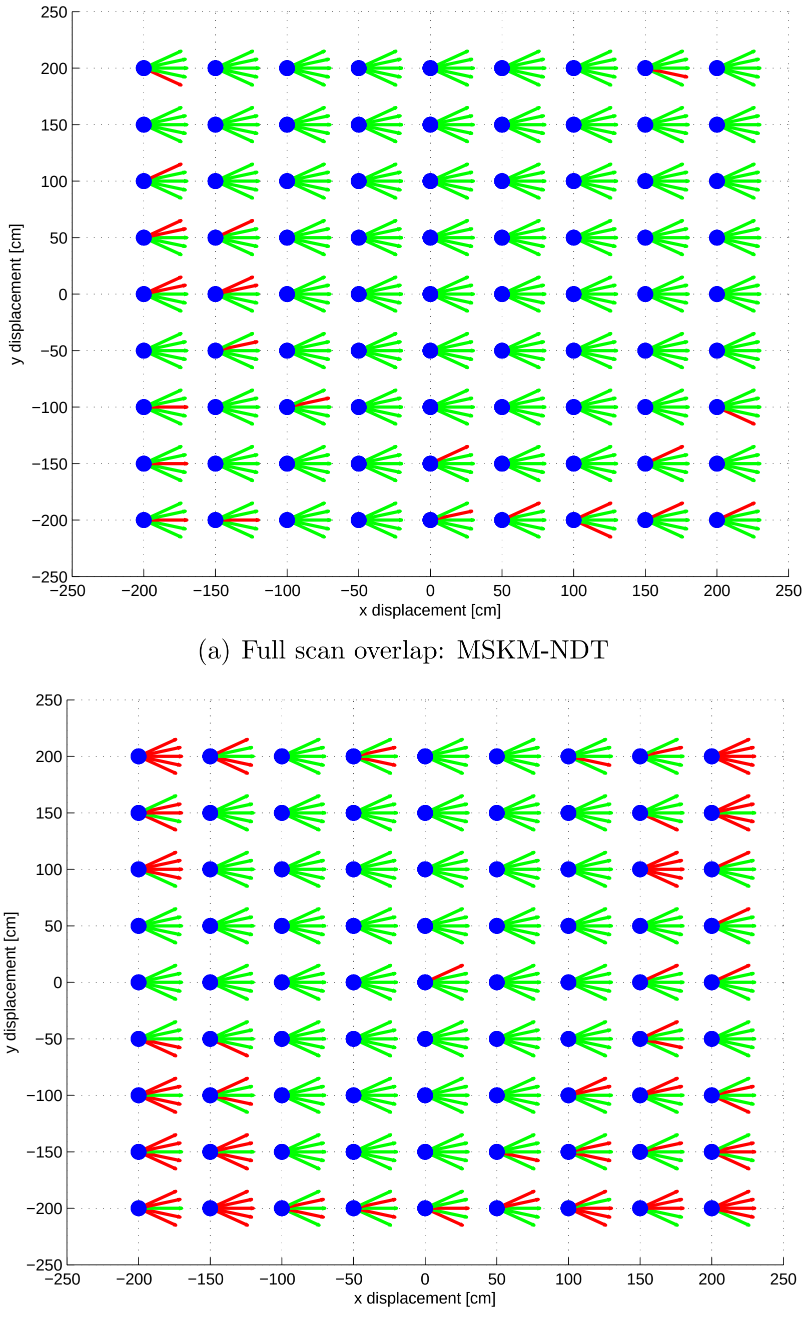
Figure 3.9: Plots illustrating the registration results for the NDT algorithm using a pair of scans from various initial transformations.



(a) Full scan overlap: MSG-NDT

(b) Partial scan overlap: MSG-NDT

Figure 3.10: Plots illustrating the registration results for MSG-NDT algorithm using a pair of scans from various initial transformations.



(a) Full scan overlap: MSKM-NDT

(b) Partial scan overlap: MSKM-NDT

Figure 3.11: Plots illustrating the registration results for the MSKM-NDT algorithm using a pair of scans from various initial transformations.

ous and non-smooth cost function that is difficult to optimize. In contrast, the MSKM-NDT algorithm does not discretize the scan using a voxel grid, but instead performs k-means clustering which ultimately results in a smooth and differentiable cost function. As a result, the MSKM-NDT yields better registration results, even though both approaches are multi-scale.

Algorithm 4 Register scene scan P^S to reference scan P^R using multi-scale NDT with k-means clustering (MSKM-NDT)

```
$\mathcal{E} \leftarrow \hat{\mathcal{E}}$
for all $\phi_i \in \Phi$ do
    Perform k-means clustering of scan:
    $\Gamma \leftarrow \text{kmeans}(P^R, \phi_i)$
    for all $\gamma_j \in \Gamma$ do
        $\mu_{\gamma_j} \leftarrow \frac{1}{|\gamma_j|} \sum_{p \in \gamma_j} p$
        $\Sigma_{\gamma_j} \leftarrow \frac{1}{|\gamma_j|-1} \sum_{p \in \gamma_j} (p - \mu_{\gamma_j})(p - \mu_{\gamma_j})^T$
    end for
    Registration of scene scan:
    while $|\, g \,| > \epsilon_{ndt}$ do
        $s \leftarrow 0$
        $g \leftarrow 0$
        $H \leftarrow 0$
        for all $p \in P^R$ do
            $\bar{p} \leftarrow T_{\mathcal{E}}(p)$
            for all $\gamma_j \in \Gamma$ do
                $s \leftarrow s + \rho_{\gamma_j}(\bar{p})$
                $g \leftarrow g + \Pi_g(\bar{p}, \mathcal{E})$
                $H \leftarrow H + \Pi_H(\bar{p}, \mathcal{E})$
            end for
        end for
        $\Delta\mathcal{E} = (H)^{-1}(-g)$
        $\mathcal{E} \leftarrow \mathcal{E} + \Delta\mathcal{E}$
    end while
end for
$\mathcal{E}^* \leftarrow \mathcal{E}$
```

Chapter 4

Segmented Greedy Cluster NDT

The voxel based discretization of NDT generates Gaussian distributions which do not necessarily model the environment accurately. The distributions only locally model the points within each cell, and may not capture broader features present within the scan. Furthermore, the crossing of points between cell boundaries during the optimization process results in a cost function which is discontinuous. The MSKM-NDT method partitions a 2D laser scan using k-means clustering over multiple scales. The sets of clusters resulting from the partitioning are used in a course-to-fine optimization method in order to avoid local minima and improve convergence. The MSKM-NDT approach results in a smooth and continuous cost function, but requires fixing the number of partitions *a priori*. Furthermore, the use of k-means clustering for 3D point clouds is not practical, as the computational complexity for the algorithm is on the order of $O(n^{dk+1} logn)$, where k is the number of clusters, n is the number of points in the point cloud, and d is dimensionality of the data [21]. In addition to the clear increase of dimensionality when transitioning from 2D to 3D data, 3D point clouds also typically contain a larger number of points, causing significant growth in computational complexity.

The idea of partitioning 3D point clouds has previously been proposed by Moosmann et al., who used a graph based approach to segment a laser scan based on local convexity criterion [38]. The graph based partitioning method computes surface normal information in order to partition the scan, however good quality surface normal information is difficult to generate for sparse laser scans, or when the environment consists of objects such as trees, brush, and foliage. Douillard et al. demonstrated scan partitioning of urban environments using Gaussian processes and incremental sample consensus [13]. The GP segmentation method was then applied to ICP scan registration, where correspondences between neighbouring points were constrained to belong to corresponding segments [12].

Thus, the notion of using point cloud partitioning to aid scan registration is valid, however a different approach is required for less structured environments, such as sparse forested areas.

The contribution of this chapter is the application of a segmentation and clustering algorithm to improve the accuracy and convergence of the NDT algorithm, specifically for large, open, outdoor environments with sparse point-clouds. Ground segmentation is performed in order to separate sparse, natural features in the environment, and are used to generate Gaussian clusters for the NDT algorithm. Clustering according to natural features is desirable since it allows for accurate modelling of the environment using significantly fewer distributions compared to the standard NDT algorithm. The application of ground segmentation also results in the decomposition of the environment into ground and obstacle partitions which can be used for higher level mission execution.

To perform ground segmentation, a Gaussian process regression [60] and incremental sample consensus [13] is applied to the 3D laser scan. A greedy clustering algorithm which partitions the non-ground points into clusters is then proposed. Assuming that the non-ground points are fully removed by the ground segmentation, the greedy clustering algorithm generates clusters which group together points belonging to features in the environment, such as trees, bushes, etc., and the clusters are used to construct Gaussian distributions. The NDT algorithm is then modified to evaluate the cost function for all generated clusters, which guarantees that the nonlinear optimization is well defined and all standard nonlinear optimization algorithms will converge to a solution which satisfies the necessary conditions for optimality [3]. Evaluation at all Gaussian clusters is performed for both the point-to-distribution and distribution-to-distribution variants of the NDT algorithm. Finally, the proposed method is evaluated for accuracy and convergence using laser data obtained from the Ford campus data set [46] and a sparse forested park at Worcester Polytechnic Institute.

4.1 Proposed Method

In order to improve the accuracy and convergence basin of the NDT algorithm, the Segmented Greedy Cluster NDT (SGC-NDT) method is proposed. The SGC-NDT method segments the ground points and performs clustering on the remaining points. Segmenting of the ground point and application of the greedy cluster algorithm is applied to both the point-to-distribution and distribution-to-distribution registration method for NDT.

The NDT algorithm is modified in the following three ways. First, the ground points of the scan are removed using the ground segmentation method described in Section 4.2. In large, outdoor, field-like environments, the ground points typically do not provide as much information for the purpose of scan registration as other environmental features such as large trees, fences, etc., and in fact, can introduce convergence issues for ICP and NDT. Scan point density also becomes an issue when dealing with the ground, as the ground points very close to the laser scan are far denser than those farther away. Non-uniform density of the point cloud is not desirable for NDT, since the high density of points near the vehicle will have a tendency to generate Gaussian distributions biased towards the sensor origin. The non-uniform density of a scan can be somewhat mitigated using a down-sampling voxel based filter for the scan, however the use of a voxel filter adds additional computation to the scan registration, and removes information from the entire scan due to the down-sampling process. Although down sampling the scan will remove the biased distribution issue for NDT, it does not solve the problem of the discontinuous cost function.

Second, the non-ground points are clustered using the greedy clustering method described in Section 4.3. By performing the clustering step, natural features in the environment, such as trees and bushes, are clustered to form the Gaussian distributions for SGC-NDT. In contrast, NDT performs a voxel based discretization, generating truncated Gaussian distributions which do not necessarily correctly model the environment. The greedy clustering method can be seen as a way to capture representative features from the environment, assuming they are spatially separated once the ground points have been removed and are distributed in a configuration that allows for the formation of clusters that are sufficient for NDT registration.

Finally, to accommodate the greedy clustering of the scan, the NDT cost calculation is modified as described in Section 4.4. The SGC-NDT algorithm evaluates the transformed point with respect to the set of all the Gaussian distributions for the clusters identified by the greedy clustering method. Similarly, for distribution-to-distribution matching NDT, each Gaussian distribution in the scene scan is scored against all Gaussian distributions from the reference scan. Evaluation at all of the distributions is computationally feasible, since the number of clusters identified in the scan is typically significantly smaller than the number of Gaussian distributions that would result from voxel based discretization. Calculating the cost based on all Gaussian distributions in the scans allows for the strongest contributions to come from the most likely clusters and provides a continuous and differentiable cost function for the optimization.

4.2 Ground Segmentation

The basis for the ground segmentation algorithm used for this work was first demonstrated by Douillard et al. [13]. Their method performs ground segmentation based on a 2D Gaussian process model of the sparse point clouds. The use of Gaussian processes provides a probabilistic framework to identify ground points using an incremental sample consensus (INSAC) method. A modified version of Douillard's ground segmentation algorithm was introduced by Tongtong et al. [57]. To reduce computation, Tongtong divides the scan into sectors based on a polar grid binning, and applies the ground segmentation to each sector separately. The Gaussian process regression is formulated for an approximate signal, based on the points contained in each sector. Tongtong's method was shown to run more quickly than the 2D method, with comparable results in segmentation quality. The 1D approximation method for ground segmentation from [57] is used in this work, and is briefly summarized in this section.

4.2.1 Polar Grid Binning

In order to perform the polar grid binning, the $x - y$ plane of the laser scan, in the laser frame, is first segmented into N_a angular sectors. The angle each sector covers, τ_a, is given by

$$\tau_a = \frac{2\pi}{N_a}$$

Each sector is further sub-divided into N_l linear range based bins. The distance each linear range bin covers, τ_l, is given by

$$\tau_i = \frac{R_{max}}{N_b}$$

where R_{max} is the maximum range measurement expected for the given scan. Denote the cell for the i^{th} sector and j^{th} linear range bin as $\xi_{ij} \subseteq \mathbb{R}^2$, and the set of all bins as Ξ where $|\Xi| = N_a N_l$. Define the points from point cloud P whose projection falls within cell ξ_{ij} as

$$P_{ij} = \{p \in P : (p^x, p^y) \in \xi_{ij}\}$$

To perform rapid ground segmentation, a set of tuples is constructed from the cell point sets, P_{ij}. In order to construct the set of tuples, a prototype point is determined for the points within each cell. For the case of ground segmentation, it is desirable to model the prototype point in each cell such that it best represents the ground. The prototype point for the points P_{ij} within each cell ξ_{ij} is selected as the point with the lowest z co-ordinate, and is given as

$$x_{ij} = \underset{p \in P_{ij}}{\operatorname{argmin}}(p^z)$$

Finally, using the prototype point for the points in each cell, the set of tuples for each angular sector can be defined. Denote a tuple associated with the prototype point x_{ij} of a cell as $y_{ij} = (r_{ij}, h_{ij}) \in \mathbb{R}^2$. For the tuple notation, the first element of the tuple $r_{ij} \in \mathbb{R}$ denotes the range component, and $h_{ij} \in \mathbb{R}$ denotes the height component. Using the prototype point for the cell, x_{ij}, the range and height components for the tuple are calculated as

$$\begin{aligned} r_{ij} &= \sqrt{(x_{ij}^x)^2 + (x_{ij}^y)^2} \\ h_{ij} &= x_{ij}^z \end{aligned}$$

The range element, r_{ij}, is the Euclidean distance of the x and y components from the prototype point, x_{ij}, to the sensor origin, and the height element, h_{ij}, is simply the z component of the prototype point. The set of tuples, Y_i , for angular sector i is defined as

$$Y_i = \{y_{ij} \, \forall \, j \in \{1 \dots N_l\}\}$$

For the ground segmentation, a *seed-set* must also be defined. The seed set is the set of points which is used to initialize the ground segmentation INSAC algorithm. For a given sector, the seed points are defined as the range-height tuple points that are within a threshold distance of the laser scan origin, δ_o. The seed points for each sector, Y_i^{seed}, are given as

$$Y_i^{seed} = \{(r, h) \in Y_i : r < \delta_o\} \tag{4.1}$$

4.2.2 Gaussian Process Regression

In this work, ground segmentation is done by performing Gaussian Process (GP) regression. For a thorough exploration of the usage of GP's for ground terrain modelling, the reader is directed to the work of Vasudevan et al. [60]. In this work, the approximate GP regression method as described in [57] is applied.

The GP regression provides a probabilistic framework to predict a set of height values for a set of query inputs, given a set of training data. Suppose the GP model is to be trained with a subset of the range-height tuple set, denoted as $\bar{Y}_i \subseteq Y_i$. A covariance function is used to quantify the correlation among data points used for the regression. In this work, a *squared-exponential covariance function*, $\kappa : \mathbb{R}^2 \mapsto \mathbb{R}$ is used. Suppose the correlation between two scalar data points, $b_1 \in \mathbb{R}$ and $b_2 \in \mathbb{R}$ is desired. Using the squared-exponential covariance function, the correlation is given by given by

$$\kappa(b_1, b_2) = \sigma_f^2 \exp(-\frac{1}{2l^2}(b_1 - b_2)^2) \tag{4.2}$$

where $l \in \mathbb{R}$ is the length-scale parameter, and $\sigma_f \in \mathbb{R}$ is the input variance, which are known as the hyper-parameters for the GP model. Intuitively, from the covariance function it can be seen that points which have similar values will be highly correlated, while points which are farther apart are less correlated. Given the covariance function, a regression problem for ground segmentation can be posed which seeks to predict the height values for the associated range values from tuple set, Y_i, given that the GP model has been trained using the data from the training set, $\bar{Y}_i$. Using the covariance function, define four matrices which correlate the query and training range data. Denote $\mathcal{K}_{qq} \in \mathbb{R}^{|Y_i| \times |Y_i|}$ as the auto-covariance matrix for the query set. Let the notation $\mathcal{K}(j, k)$ access element (j, k) of the matrix $\mathcal{K}(j, k)$. To simplify notation, let the range and height values of a tuple element $y_k \in Y_i$ be denoted as r_k and h_k, respectively. Then, each element of the matrix $\mathcal{K}_{qq}$ is populated using the squared exponential covariance function given by Equation 4.2,

$$\mathcal{K}_{qq}(j, k) = \kappa(r_j, r_k) \,\forall\, y_j, y_k \in Y_i$$

Similarly, the cross-covariance matrix between the query and training data, $\mathcal{K}_{qt} \in \mathbb{R}^{|Y_i| \times |\bar{Y}_i|}$, and the auto-covariance matrix for the training data, $\mathcal{K}_{tt} \in \mathbb{R}^{|\bar{Y}_i| \times |\bar{Y}_i|}$, can be defined as

$$\begin{aligned} \mathcal{K}_{qt}(j, k) &= \kappa(r_j, r_k) \,\forall\, y_j \in \bar{Y}_i \,\text{and}\, y_k \in Y_i \\ \mathcal{K}_{tq} &= \mathcal{K}_{qt}^T \\ \mathcal{K}_{tt}(j, k) &= \kappa(r_j, r_k) \,\forall\, y_j, y_k \in \bar{Y}_i \end{aligned}$$

Using the auto and cross correlation matrices, the Gaussian process models the relationship between the input and training data as a joint distribution. Denote the vector of height values from the training set as the vector $\lambda = [h_1 \dots h_{|\bar{Y}_i|}]^T$, where $h \in \bar{Y}_i$. Similarly, denote the vector λ^* as height values to be inferred using the GP regression. The GP is then modelled as the joint distribution

$$\begin{bmatrix} \lambda \\ \lambda^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathcal{K}_{tt} + \sigma_n^2 I & \mathcal{K}_{tq} \\ \mathcal{K}_{qt} & \mathcal{K}_{qq} \end{bmatrix}\right)$$

where σ_n^2 is an additional hyper-parameter, the process noise variance.

From the joint distribution, an expression for the conditional mean and conditional variance can be expressed. The conditional mean and covariance expressions result in the *predictive equations*. In other words, using the training data and the predictive equations, a vector of ground height values, λ^*, and associated covariances, V_{λ^*}, can be predicted for a set of queried range values. The conditional mean and covariance predictive equations are given as

$$\begin{aligned} \lambda^* &= \mathcal{K}_{qt} \left[\mathcal{K}_{tt} + \sigma_n^2 I\right]^{-1} \lambda \\ V_{\lambda^*} &= \mathcal{K}_{qq} - \mathcal{K}_{qt} \left[\mathcal{K}_{tt} + \sigma_n^2 I\right]^{-1} \mathcal{K}_{tq} \end{aligned}$$

Denote the method $\Upsilon(Y_i, \bar{Y}_i)$, that generates the four correlation matrices $\mathcal{K}_{tt}$, $\mathcal{K}_{tq}$, $\mathcal{K}_{qt}$, and $\mathcal{K}_{qq}$ using the training data, Y_i, and query points, $\bar{Y}_i$, for the angular sector, i.

From the predictive equations, the estimated ground height for a given range value from the set of query data can be generated. Using index notation, denote λ_k^* as the k^{th} element of the query height vector, λ^*, and similarly denote λ_k as the k^{th} element of the vector of training heights, λ. Further denote N_λ^* as the number of elements in the vector, λ^* and note that $N_\lambda^* = |Y_i|$. In order to determine if the query point is an inlier according to the GP model, the predicted ground height, λ_k^*, can be compared against the actual ground height from the tuple, $y_k = (r_k, h_k) \in Y_i$, for the query point. To be considered an inlier with the GP model, two criteria must be fulfilled:

$$\begin{aligned} V_{\lambda_k^*} &< \delta_{model} \\ \frac{h_k - \lambda_k^*}{\sqrt{\sigma_n^2 + V_{\lambda_k^*}}} &< \delta_{data} \end{aligned}$$

where δ_{model} defines the threshold of covariance for the test point and δ_{data} defines a normalized distance of the test point to its expected value from the GP model. The comparison against δ_{model} ensures that there is sufficient confidence in the GP model to allow for inlier classification, and the comparison against δ_{data} uses the distance between the predicted ground height and measured ground height to classify the query point as an inlier to the GP model.

4.2.3 Ground Segmentation Using Gaussian Process Regression and Incremental Sample Consensus

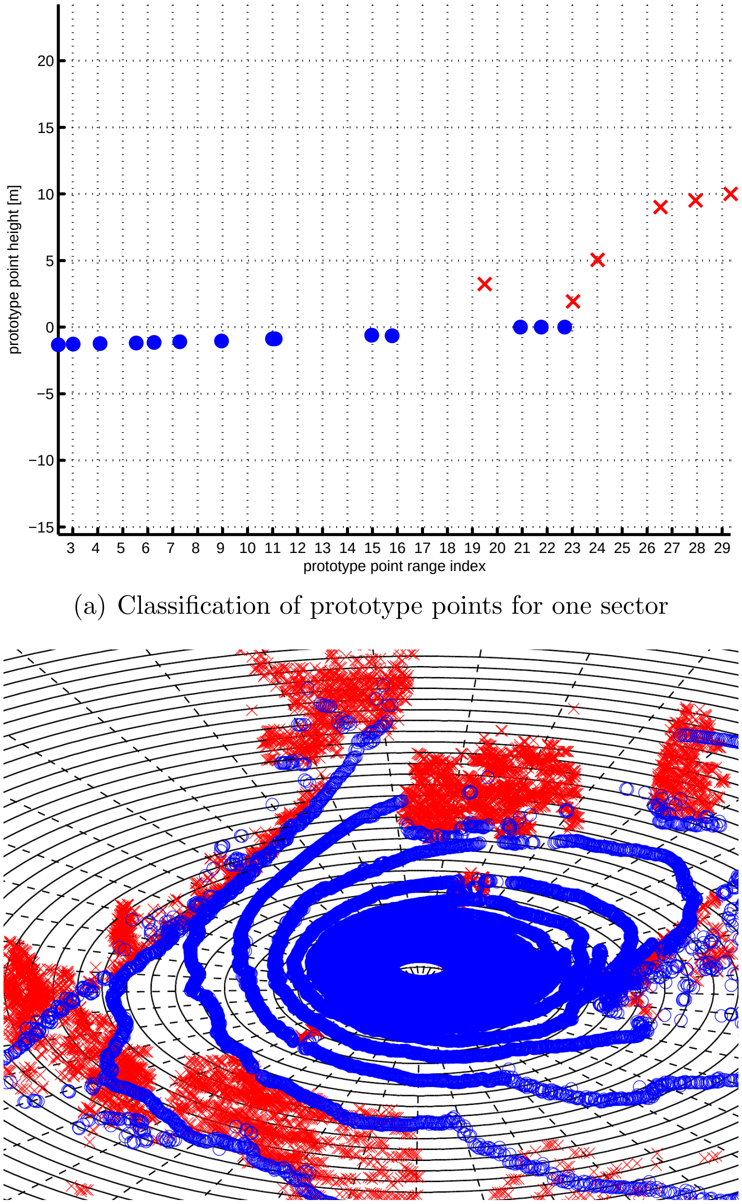
The INSAC algorithm is used in conjunction with the GP regression model to identify all of the ground points in the laser scan. The ground points are separately identified for each sector of the polar grid bins. The 3D scan is first binned into sectors as discussed in section 4.2.1. For each sector, the training points for the GP model, $\bar{Y}_i$, are selected as the sector's seed points. The remainder of the prototype point tuples in the query set, Y_i, are evaluated against the GP model. The inlier tuples for the current iteration are added to the inlier set, Y_{in}, and a new GP model is produced. The process of adding inliers and re-generating the GP model is repeated until no further inliers can be added to the set. Since there are no remaining inliers, all the prototype points for the bins have been classified as ground or not ground. To classify the remaining points in each ground bin, the height component of each point is compared against the height component of the ground prototype point. If the absolute difference in height is less than a user defined threshold, δ_g, the points are also classified as ground points. Denote the method $P^g \leftarrow \vartheta(P)$, which accepts point cloud P and returns the ground points P^g. The ground segmentation process is summarized in Algorithm 5. Ground segmentation results for an example laser scan are given in Figure 4.1.

4.3 Clustering of the Non-Ground Points

Once the ground points have been segmented, the remaining non-ground points must be clustered in order to apply the SGC-NDT algorithm. Clustering of point cloud data is an active area of research, as Golovinskiy et al. [16] demonstrated 3D point cloud clustering using a hierarchical approach, and Klasing et al. [24] performed clustering on 3D data using a radially bounded nearest neighbour (RBNN) graph. Although the demonstrated segmentation performance for the existing approaches is strong, the algorithms are evaluated on very dense point clouds, and intuitively, the methods would not perform well when the point clouds are sparse. In order to perform clustering of the non-ground points of a

Algorithm 5 $\vartheta(P)$: Given point cloud P, find the ground points, P^g, using GP-INSAC

```
$P^g \leftarrow \emptyset$
**for all** $i \in \{1 \ldots N_a\}$ **do**
    $Y_i \leftarrow \emptyset$
    **for all** $j \in \{1 \ldots N_l\}$ **do**
        $x_{ij} = \underset{p \in P_{ij}}{\text{argmin}}(p^z)$
        $r_{ij} = \sqrt{(x_{ij}^x)^2 + (x_{ij}^y)^2}$
        $h_{ij} = x_{ij}^z$
        $y_{ij} = (r_{ij}, h_{ij})$
        $Y_i \leftarrow Y_i \cup y_{ij}$
    **end for**
**end for**
**for all** $i \in \{1 \ldots N_a\}$ **do**
    $Y_i^{seed} \leftarrow \{(r, h) \in Y_i : r < \delta_o\}$
    $\bar{Y}_i \leftarrow Y_i^{seed}$
    $Y_{in} \leftarrow \bar{Y}_i$
    $Y_i \leftarrow Y_i \backslash \bar{Y}_i$
    **while** $Y_{in} \neq \emptyset$ **do**
        $Y_{in} \leftarrow \emptyset$
        $[\mathcal{K}_{qq}, \mathcal{K}_{qt}, \mathcal{K}_{tq}, \mathcal{K}_{tt}] \leftarrow \Upsilon(Y_i, \bar{Y}_i)$
        $\lambda^* \leftarrow \mathcal{K}_{qt} \left[\mathcal{K}_{tt} + \sigma_n^2 I\right]^{-1} \lambda$
        $V_{\lambda^*} \leftarrow \mathcal{K}_{qq} - \mathcal{K}_{qt} \left[\mathcal{K}_{tt} + \sigma_n^2 I\right]^{-1} \mathcal{K}_{tq}$
        **for all** $y_k \in Y_i$ **do**
            **if** $V_{\lambda_k^*} < \delta_{model}$ and $\frac{h_k - \lambda_k^*}{\sqrt{\sigma_n^2 + V_{\lambda_k^*}}} < \delta_{data}\}$ **then**
                $Y_{in} \leftarrow Y_{in} \cup y_k$
            **end if**
        **end for**
        $\bar{Y}_i \leftarrow \bar{Y}_i \cup Y_{in}$
        $Y_i \leftarrow Y_i \backslash Y_{in}$
        **for all** $j \in \{1 \ldots N_l\}$ **do**
            **if** $(r_{ij}, h_{ij}) \in \bar{Y}_i$ **then**
                $P_{in} \leftarrow \{p \in P_{ij} : |p^z - h_{ij}| < \delta_g$
                $P^g \leftarrow P^g \cup P_{in}$
            **end if**
        **end for**
    **end while**
**end for**
```



(a) Classification of prototype points for one sector

(b) Ground segmentation result for 3D laser Scan

Figure 4.1: Ground segmentation results for an outdoor scene with large trees and shrubs. Blue points have been classified as ground, and red X points are classified as non-ground. Figure (a) shows the classification results for a single sector after performing the GP based ground segmentation. The x-axis shows the division of bins for the sector. Figure (b) shows the classification results for the entire 3D laser scan. The scan has been binned with the parameters $\tau_a = 8$ degrees and $\tau_l = 1.875$ meters.

sparse point cloud, greedy clustering technique based on bins of points is used. Using a similar polar grid technique as described in Section 4.2.1, except using a single index for notational simplicity, denote a polar grid cell as $\xi_i \subset \mathbb{R}^2$ and the set of all such cells as Ξ. For a point cloud of non-ground points, P^{ng}, denote the points which fall within cell ξ_i as

$$P_i^{ng} = \{p \in P^{ng} : (p^x, p^y) \in \xi_i\}$$

and the set of all cell point sets as $\mathcal{P} = \{P_1^{ng} \dots P_{|\Xi|}^{ng}\}$ where $|\mathcal{P}| = |\Xi|$. Denote a cluster of points as $\gamma \subseteq P^{ng}$, and the set of clusters as $\Gamma = \{\gamma_1 \dots \gamma_{N_\Gamma}\}$, where N_Γ is the total number of clusters, which is initially unknown. The clusters are generated by first randomly sampling an initial bin index, i, from a random uniform distribution, $\mathcal{U}$, such that $i \sim \mathcal{U}(i \in \{1 \dots |\Xi|\})$. Letting $\mu_i \in \mathbb{R}^d$ be the mean of point set P_i, denote the distance function $d : \mathcal{P} \times \mathcal{P} \mapsto \mathbb{R}$ return the distance between point sets P_i and P_j as

$$d(P_i, P_j) = \|\mu_i - \mu_j\|_m \tag{4.3}$$

where m denotes the type of norm to be used. Using Equation 4.3, the nearest neighbour set of point set Ω, to point set P_i can be given as

$$\Omega = \{P \in \mathcal{P} : d(P, P_i) < \delta_{nn}\} \backslash P_i \tag{4.4}$$

where δ_{nn} determines the size of neighbour to consider for point set P_i.

For the randomly selected point set P_i, the nearest neighbours are generated and added to the nearest neighbour set, Ω. An element from the nearest neighbour set, $P_j \in \Omega$, can be compared against P_i to determine if the two can be combined. Merging of the two clusters is decided using the function $\varrho : \mathcal{P} \times \mathcal{P} \mapsto \{0, 1\}$. The function $\varrho(P_i, P_j)$ returns 1 if the two point sets P_i and P_j can be merged, and 0 otherwise. In the merging function, different metrics can be used to compare the point sets, such as Euclidean distance between the mean of the points, the L_2 distance between Gaussian distributions formed by the point sets, or a test for Gaussian fit. If the bins can be combined, the neighbour bin is added to an open list of point sets, Q_o, where the notation Q_o^k denotes the k^{th} element of Q_o. The neighbour point set is also removed from the set of all point sets and is merged into the points of the current cluster, γ_{cur}. The nearest neighbours of the bins from Q_o are explored and evaluated. When a bin is explored, its nearest neighbours are also added to Q_o, based on the evaluation function, ϱ. Once Q_o is empty, it means no further nearest neighbours can be assigned to the current cluster so a new point set from $\mathcal{P}$ is selected, and the process is repeated. The process continues until all non empty point sets in $\mathcal{P}$ have been assigned to a cluster. The greedy-cluster algorithm is presented in Algorithm 6. Denote the method

$\Gamma \leftarrow \varsigma(P^{ng})$ from Algorithm 6, which accepts the point cloud of non ground points and returns the set of clusters determined by the greedy clustering algorithm. An example of the resulting Gaussian distributions after the greedy clustering has been completed is presented in Figure 4.2.

Algorithm 6 $\varsigma(P^{ng})$: Given the point set for the non ground point cloud, $\mathcal{P}$, perform Greedy Clustering and return the set of clusters Γ.

```
while $\mathcal{P} \neq \emptyset$ do
    $i \sim \mathcal{U}(i \in \{1 \ldots |\Xi|\})$
    $\mathcal{P} \leftarrow \mathcal{P} \backslash P_i$
    $\gamma_{cur} \leftarrow \emptyset$
    if $P_i \neq \emptyset$ then
        $\gamma_{cur} \leftarrow P_i$
        $Q_o \leftarrow \{P_i\}$
        while $Q_o \neq \emptyset$ do
            $\tilde{P} \leftarrow Q_o^1$
            $Q_o \leftarrow Q_o \backslash Q_o^1$
            $\Omega = \{P \in \mathcal{P} : d(P, \tilde{P}) < \delta_{nn}\} \backslash \tilde{P}$
            for all $P \in \Omega$ do
                if $\varrho(P, \tilde{P})$ then
                    $Q_o \leftarrow Q_o \cup \{P\}$
                    $\gamma_{cur} \leftarrow \gamma_{cur} \cup P$
                    $\mathcal{P} \leftarrow \mathcal{P} \backslash P$
                end if
            end for
        end while
        $\Gamma \leftarrow \Gamma \cup \{\gamma_{cur}\}$
    end if
end while
```

4.4 Segmented Greedy Cluster NDT

Using the Gaussian distributions generated by the greedy clustering, the NDT algorithm is modified to evaluate the cost function using all of the Guassians. Recall that the standard NDT registration algorithm subdivides the space occupied by the reference scan into fixed size, rectilinear grid cells, and models the points occupying each cell with a mean and

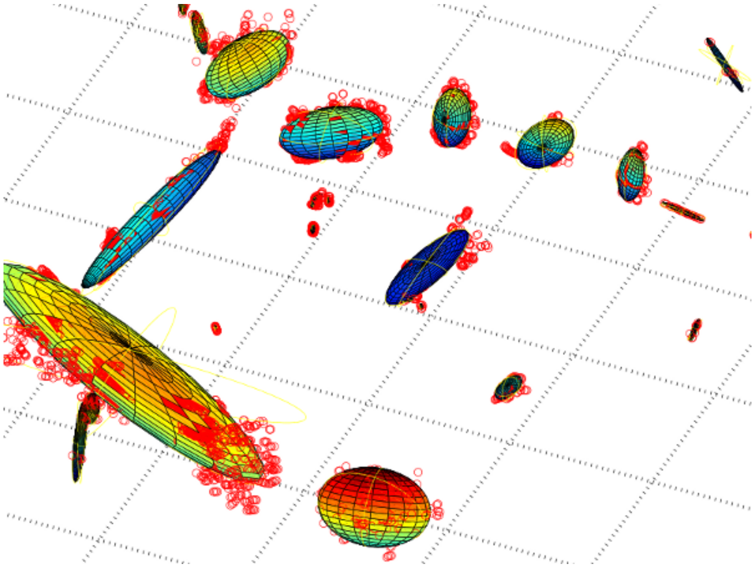


Figure 4.2: The resulting Gaussian distributions after performing the greedy cluster algorithm on the non-ground point laser scan. One-sigma Gaussian distributions are shown.

covariance for a representative Gaussian distribution. For SGC-NDT, the reference scan is clustered using the greedy cluster method, and the resulting clusters of points are used directly to generate the Gaussian distributions which model the reference scan. Denote a point in the scene scan that has been transformed by parameter estimate $\mathcal{E}$, as $\bar{p} = T_{\mathcal{E}}(p)$. The transformed point can be scored against a Gaussian cluster, γ_j, using the NDT score function, ρ, and is given as

$$\rho_{\gamma_j}(\bar{p}) = \exp\left(-\frac{(\bar{p} - \mu_j^T)\Sigma_j^{-1}(\bar{p} - \mu_j)}{2}\right) \tag{4.5}$$

The points in the scene scan are scored against all of the Gaussian clusters from the reference scan, which allows for the strongest contributions to come from the most likely clusters and provides a continuous and differentiable cost function for the optimization. Similar to the standard NDT algorithm, it is possible to compute an analytic gradient, g, and Hessian H, for the cost function, which can be used for improving the performance of the nonlinear optimization. For point to distribution matching, denote the functions $\Pi_g(\bar{p}, \mathcal{E})$, and $\Pi_H(\bar{p}, \mathcal{E})$, which use a point in the scene scan and the transformation parameters to calculate an associated gradient and Hessian contribution for point $\bar{p}$. The optimization is initialized from a parameter estimate $\hat{\mathcal{E}}$, and terminates once the norm of the gradient $\|g\|$

is less than a user specified threshold, ϵ_{ndt}.

A summary of the SGC-NDT-P2D is given in Algorithm 7. For SGC-NDT-D2D, a method similar to the SGC-NDT-P2D version is applied, except both laser scans are segmented and clustered and the set of resulting Gaussian distributions for each scan are compared using the distribution to distribution NDT cost function. For SGC-NDT-D2D, the L_2-distance between the sets of Gaussian distributions of the reference and scene scan are minimized, and the cost function evaluates all pair-wise Gaussian components for both the scene and the reference scan.

4.5 Experimental Results

The SGC-NDT approach is evaluated using two data sets of different environments and is compared against other scan registration algorithms. The algorithms used for comparison are ICP, NDT and G-ICP, and for the purposes of comparison, the Point Cloud Library (PCL) implementations of these algorithms are used [6]. The ICP algorithms are implemented using a maximum correspondence distance of 10m and the NDT algorithm was implemented with a grid size of 3m. For all optimization based approaches, the optimization is terminated when the norm of the gradient or the norm of the step size falls below 10^{-6}. The criteria for merging bins in the greedy cluster algorithm is a test for Gaussian fit, which is also used by Stoyanov to partition NDT cells [54].

In order to test the accuracy of SGC-NDT against other scan registration methods, the Ford Campus Vision and LIDAR data set [46] and the Worcester Polytechnic Institute (WPI) data set are used. The Ford data set was generated using a Velodyne HDL-64E LIDAR with ground truth acquired by integrating the high quality velocity and acceleration measurements produced by a Applanix POS-LV 420 INS with Trimble GPS. When scans were aligned using the ground truth measurements, the alignment was seen to have on the order of sub-degree accuracy in orientation and decimeter accuracy in position. To determine the robustness of the proposed method, in a sparse, forested environment, the WPI data set is used. The WPI data was collected as part of the *NASA Sample Return Robot Challenge*, at Worcester Polytechnic Institute (WPI). The laser scans were collected using a Velodyne HDL-32E LIDAR, mounted to a custom made chassis designed for the challenge. The environment consists mainly of trees and other foliage, as well as a single small gazebo structure. Forested environments are particularly challenging for registration because foliage, grass and bush tend to create irregular, noisy point clouds with inconsistent local surface normals, making point correspondences and accurate surface normal calculations difficult.

Algorithm 7 Register scene scan P^S with reference scan P^R using SGC-NDT-P2D Algorithm and initial parameter estimate $\hat{\mathcal{E}}$

```
$\mathcal{E} \leftarrow \hat{\mathcal{E}}$
$P^g \leftarrow \vartheta(P^R)$
$P^{ng} \leftarrow P^R \backslash P^g$
$\Gamma \leftarrow \varsigma(P^{ng})$
for all $\gamma_j \in \Gamma$ do
    $\mu_{\gamma_j} \leftarrow \frac{1}{|\gamma_j|} \sum_{p \in \gamma_j} p$
    $\Sigma_{\gamma_j} \leftarrow \frac{1}{|\gamma_j| - 1} \sum_{p \in \gamma_j} (p - \mu_{\gamma_j})(p - \mu_{\gamma_j})^T$
end for
while $|\, g \,| > \epsilon_{ndt}$ do
    $s \leftarrow 0$
    $g \leftarrow 0$
    $H \leftarrow 0$
    for all $p \in P^S$ do
        $\bar{p} \leftarrow T_{\mathcal{E}}(p)$
        for all $\gamma_j \in \Gamma$ do
            $s \leftarrow s + \rho_{\gamma_j}(\bar{p})$
            $g \leftarrow g + \Pi_g(\bar{p}, \mathcal{E})$
            $H \leftarrow H + \Pi_H(\bar{p}, \mathcal{E})$
        end for
    end for
    $\Delta\mathcal{E} = (H)^{-1}(-g)$
    $\mathcal{E} \leftarrow \mathcal{E} + \Delta\mathcal{E}$
end while
$\mathcal{E}^* \leftarrow \mathcal{E}$
```

To further assess the scan registration quality for the tested algorithms, the pair-wise registered scans are aggregated into a global map. The maps are evaluated using a *crispness* measure, which is also used for map evaluation in [12]. The crispness of a map is evaluated by discretizing the 3D space into voxels and recording the number of occupied voxels. For the experiment, a voxel size of 0.1 m is used. A map which evaluates to a lower number is said to be crisper than a map which evaluates to a higher number. Intuitively, the crispness quality measure captures the blurring of the map caused by incremental pair-wise registration errors as the scans are aggregated.

4.5.1 Justification of Ground Segmentation

To demonstrate the effect of ground segmentation on scan registration quality, an experiment is conducted where ground segmentation is performed to varying degrees prior to scan registration. For the Ford data set, a set of scans is composed using every 5th scan from the data set. Every 5th scan is selected since the differential pose between scans is relatively small, thus allowing for convergence of the scan registration to the correct parameters. For every scan in the scan set, the ground is segmented using the GP-INSAC method. Once the ground is segmented, a portion of the ground points is combined with the non ground points, and the portion of ground points is combined by first passing the ground points through a voxel grid filter. The proportion of ground points combined with the non ground points is controlled by the parameters of the voxel grid filter, as the larger the size of the voxel grid, the fewer the number of ground points included in the registration process. In the experiments, the proportion of ground points combined with non ground points range from *full ground (FG)* to *no ground (NG)*, with increasing voxel grid filter size between the FG and NG range. For each scan set, the scans within each set are pair-wise registered using the ICP, G-ICP and NDT algorithms. The normalized crispness measure (normalized to the crispness of the aggregated map with full ground) is calculated, as well as the translation and rotation registration errors as compared against the data set's ground truth. The computation time for the registration is also recorded. The results of the experiment for the Ford data set, comparing ICP, G-ICP, and NDT are presented in Figures 4.3, 4.4 and 4.5. A similar experiment is performed using the the WPI data set, however since no ground truth data is available, only the normalized crispness measure and run time are calculated. The results of the experiment using the WPI data set, comparing the ICP, G-ICP and NDT algorithms are presented in Figures 4.6, 4.7 and 4.8. It should be noted that the upper adjacent and lower adjacent whiskers of the box-plots represent 1.5× the interquartile range of the data.

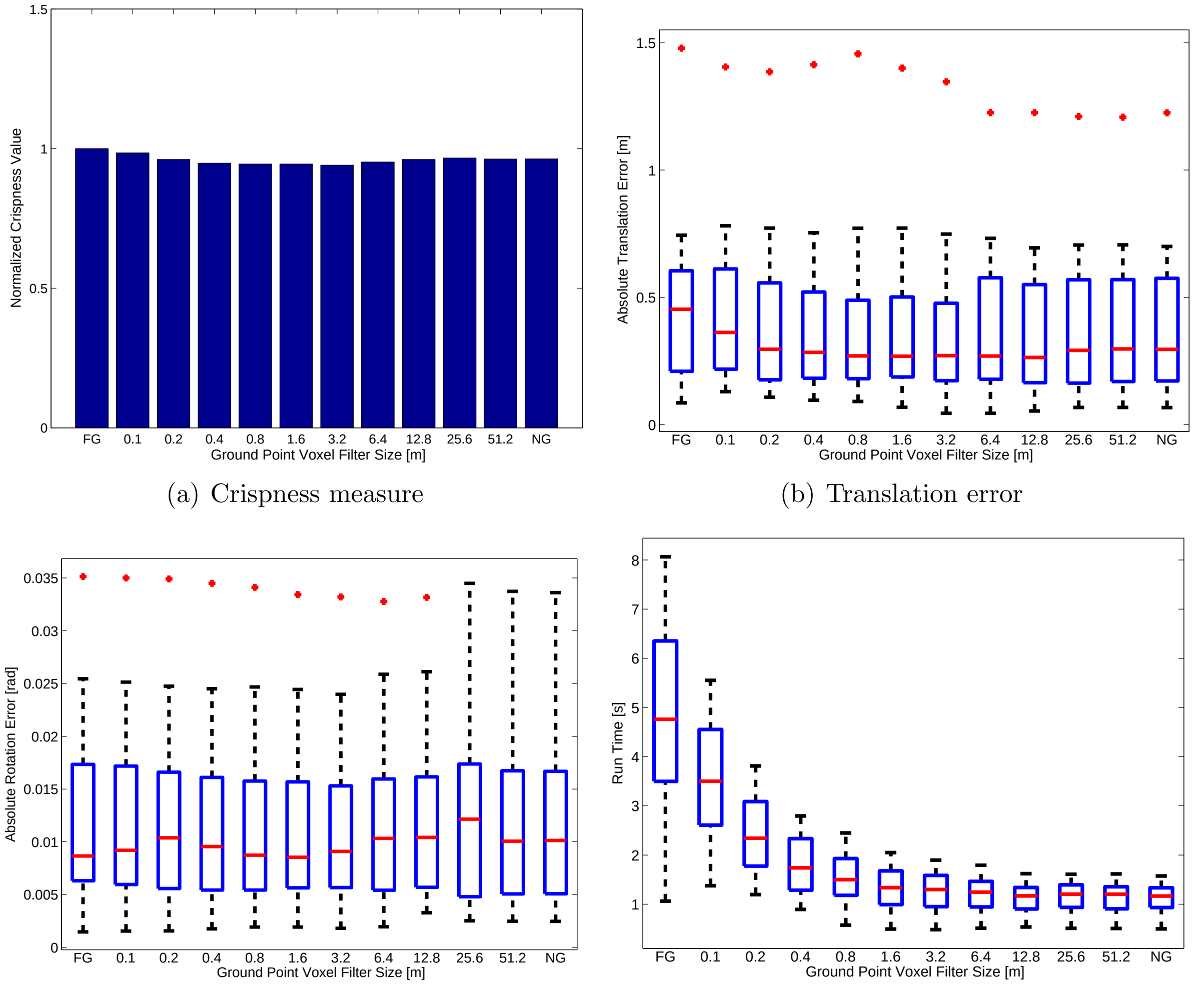
From the results of the experiments, it is evident that the removal of the ground points does not significantly affect the quality of the scan registration. For the aggregated map using the ICP, G-ICP and NDT algorithms, the normalized crispness measure from the aggregated map constructed using the down sampled ground point does not deviate significantly from the crispness result generated using scans containing the full ground. From the Ford data set, it is seen that the error in the translation and rotation parameters, as compared to the ground truth, is relatively constant across the varying degrees of ground removal, suggesting that the removal of the ground does not negatively affect the scan registration quality and that the ground points do not contribute any significant information to the registration process. The execution time plot illustrates that as increasing percentages of the ground points are removed, the registration run time decreases. The decrease in run time is expected, as the reduction of points from the ground removal will inherently decrease scan registration time, especially for the ICP and G-ICP algorithms where nearest neighbour correspondences are required. Overall, the experiment illustrates that the removal of the ground does not negatively affect the quality of the scan registration, while decreasing the run time for the scan registration algorithm. Similar results are illustrated by the experiment conducted using the WPI data set, where the crispness measure remains relatively constant across varying degrees of ground removal, while the required registration run time is decreased.

4.5.2 Accuracy in Outdoor Urban Environments

In order to test the accuracy of the scan registration in an urban environment, each algorithm performs pair-wise scan registration using every 10^{th} scan produced by the LIDAR from the Ford data set. The registration is initialized with a parameter estimate of zero translation and zero rotation. The error in the translation and rotation of registration is then compared against the ground truth measurements. The translation and rotational error distributions are displayed in Figure 4.9.

The error distributions from Figure 4.9 demonstrate that both SGC-NDT algorithms produce accurate results when compared against the ground truth, as does the G-ICP algorithm. The large translational error distribution for NDT are a result of the algorithm's poor convergence characteristics, as the optimization has a tendency to converge to local minima. The ICP algorithm also shows fairly low translation and rotation errors, however, these do not appear to be as accurate as SGC-NDT or G-ICP.

The crispness measure is evaluated for SGC-NDT, ICP and G-ICP. A crispness for NDT is not evaluated since the registration did not converge to the global minimum on numerous occasions. The crispness measures comparing SGC-NDT, ICP and G-ICP are presented in Table 4.1.



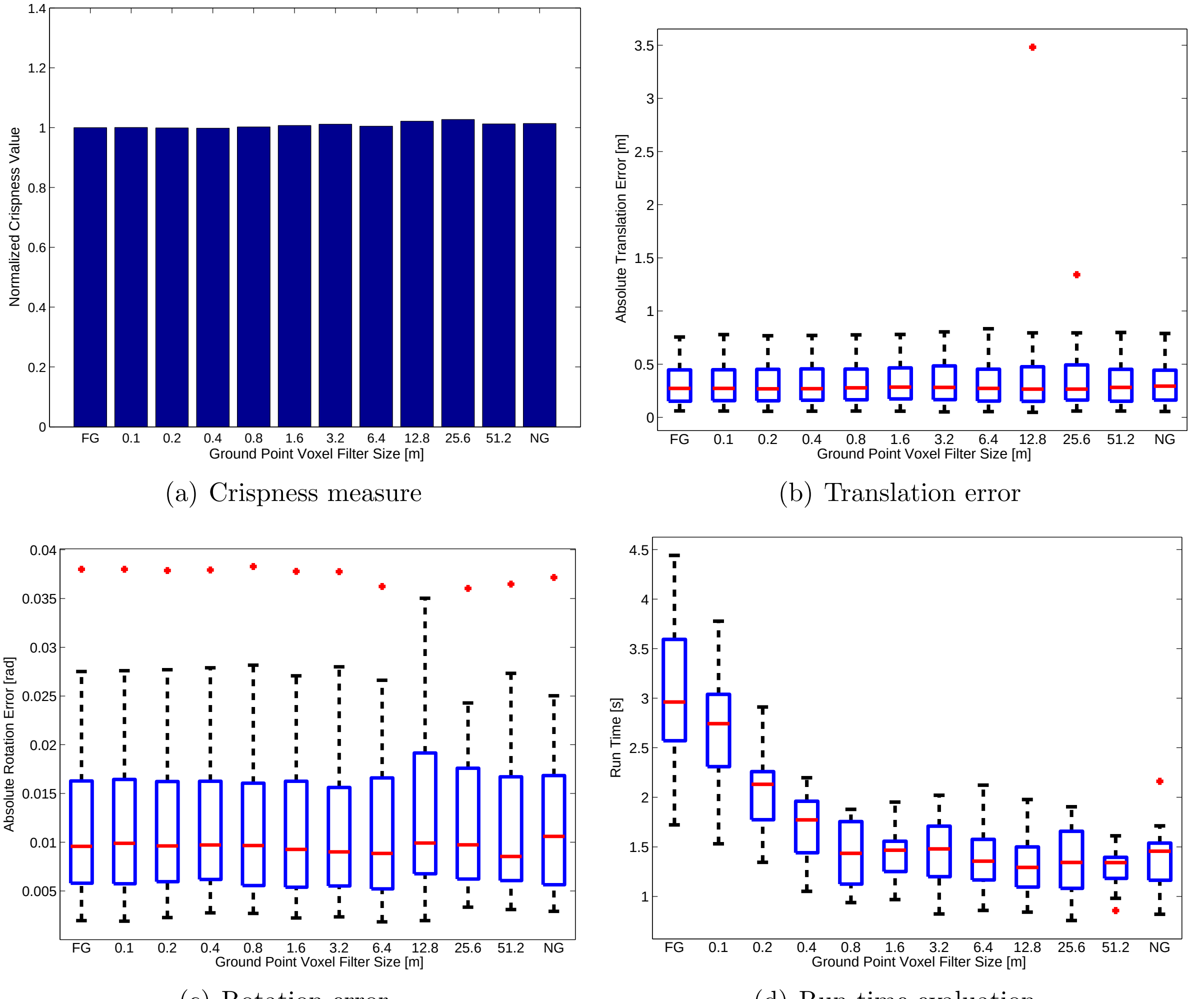
(a) Crispness measure

(b) Translation error

(c) Rotation error

(d) Run time evaluation

Figure 4.3: Ford data set: Results of scan registration accuracy and run time for ICP algorithm with varying degrees of ground removal.



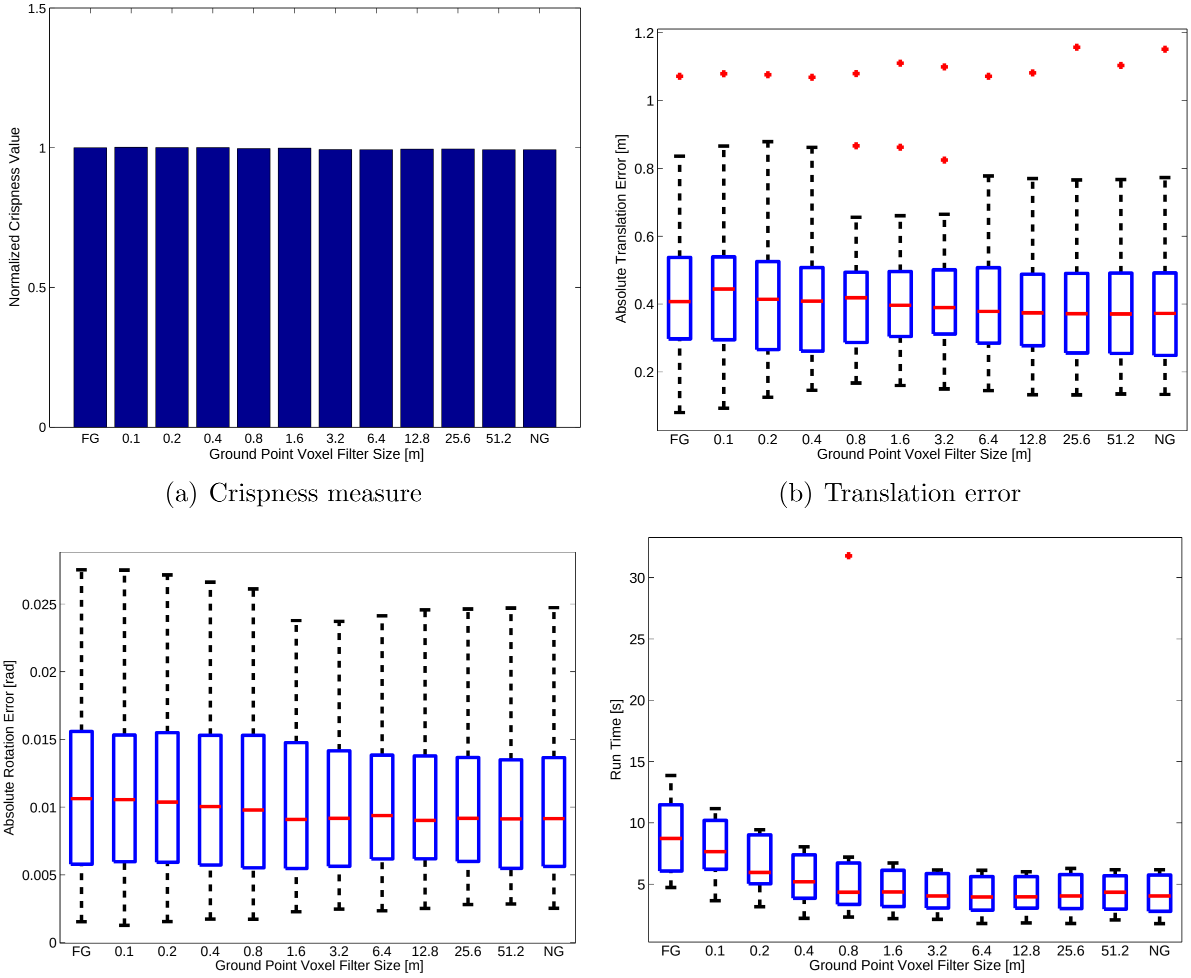
(a) Crispness measure

(b) Translation error

(c) Rotation error

(d) Run time evaluation

Figure 4.4: Ford data set: Results of scan registration accuracy and run time for GICP algorithm with varying degrees of ground removal.



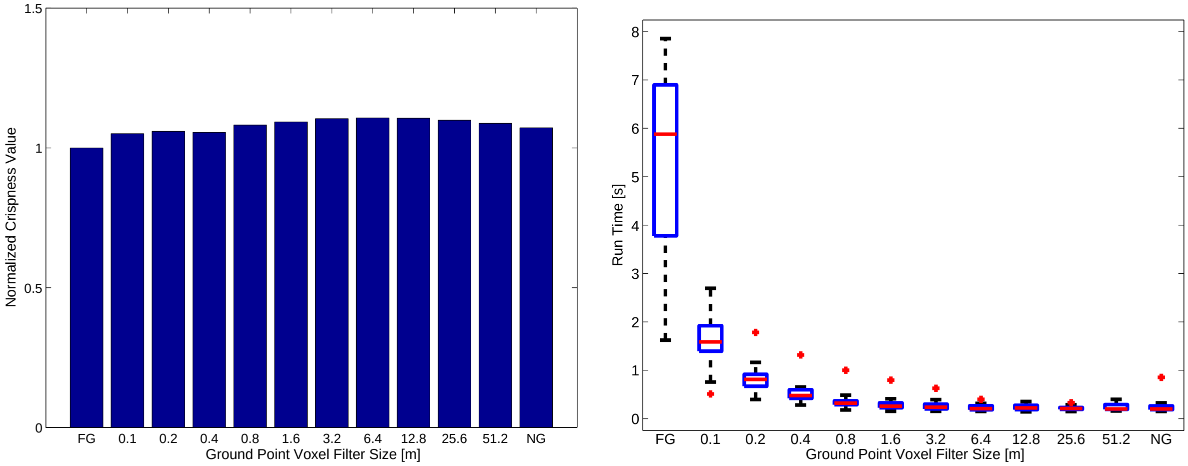
(a) Crispness measure

(b) Translation error

(c) Rotation error

(d) Run time evaluation

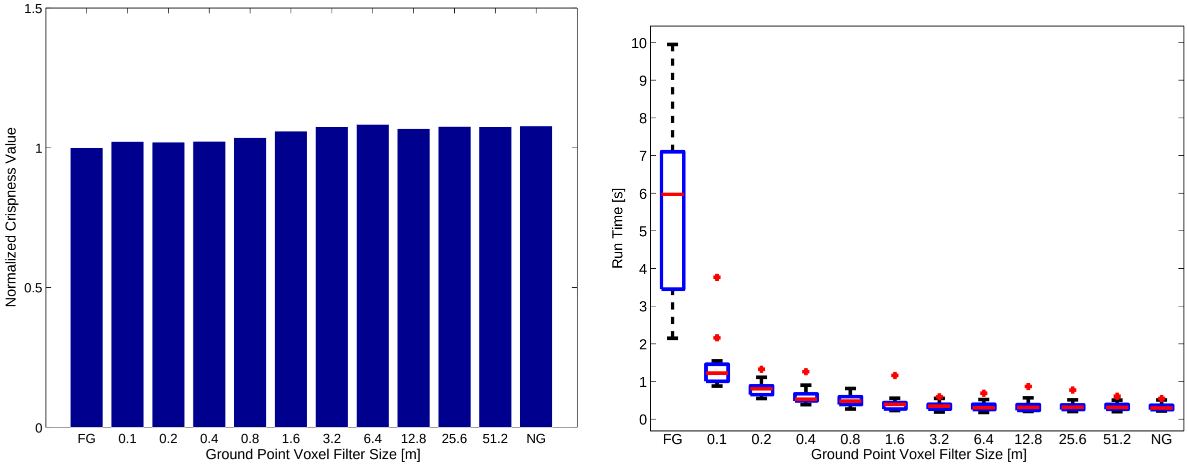
Figure 4.5: Ford data set: Results of scan registration accuracy and run time for NDT algorithm with varying degrees of ground removal.



(a) Crispness measure

(b) Run time evaluation

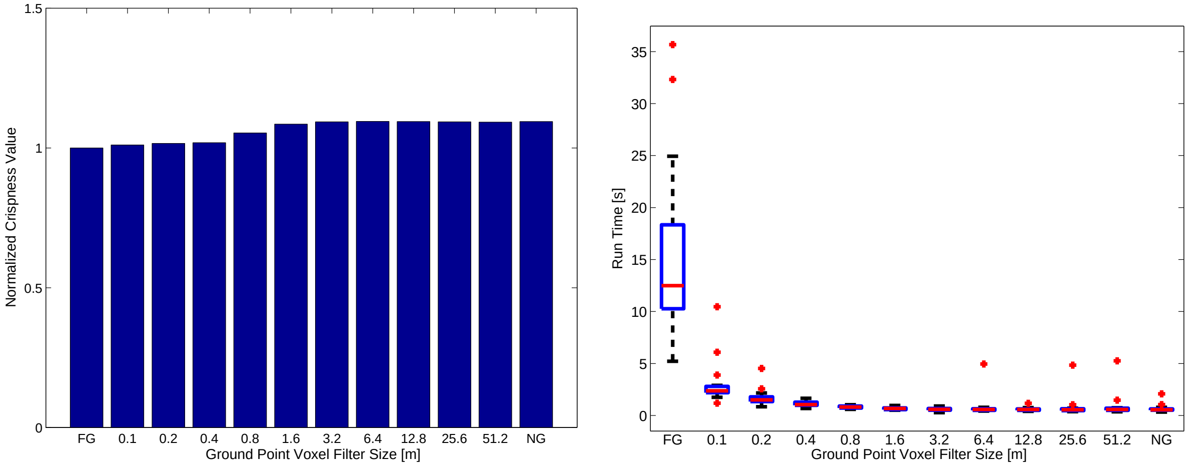
Figure 4.6: WPI data set: Results of scan registration crispness and run time for ICP algorithm with varying degrees of ground removal.



(a) Crispness measure

(b) Run time evaluation

Figure 4.7: WPI data set: Results of scan registration crispness and run time for G-ICP algorithm with varying degrees of ground removal.



(a) Crispness measure

(b) Run time evaluation

Figure 4.8: WPI data set: Results of scan registration crispness and run time for NDT algorithm with varying degrees of ground removal.

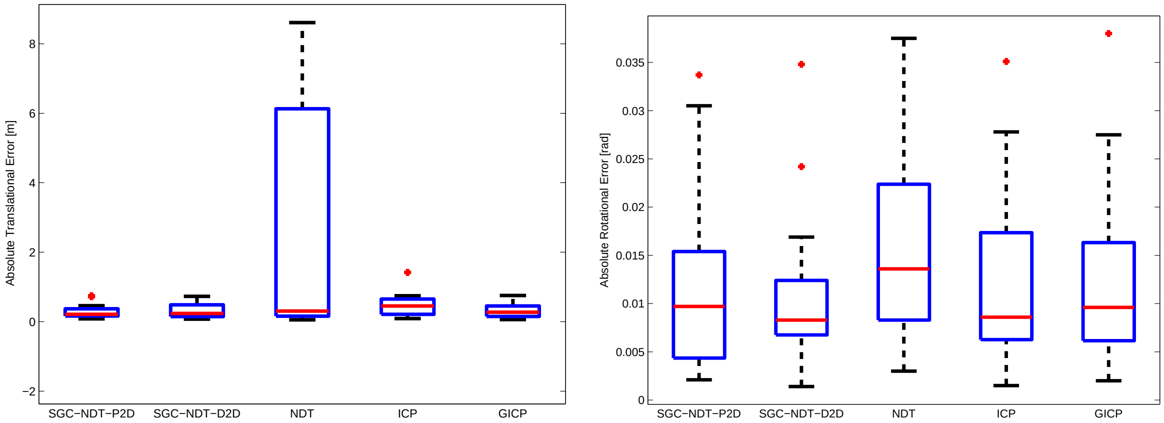
	ICP	G-ICP	SGC-NDT-P2D	SGC-NDT-D2D
Ford	132507	124911	125767	125221
WPI	-	44609	41884	40560

Table 4.1: Summary of crispness measures for registration tests in urban and forested environments. Note that the crispness for ICP with the WPI data set is not presented as the registration did not converge to the global minimum on numerous occasions.

The crispness measures indicate that both SGC-NDT algorithms and G-ICP generate high quality maps, while ICP produces a lower quality map due to the accumulation of registration error, as illustrated in Figure 4.10. The global map of G-ICP is visually indistinguishable from that of SGC-NDT and is therefore not included in the figure. The qualitative representation of the crispness measure is demonstrated by visually comparing the two maps. The maps produced by SGC-NDT and G-ICP show very little blurring of features such as the building walls and cars. Conversely, the map generated by ICP has a blurred appearance and less sharp features.

4.5.3 Accuracy in Sparse Forested Environments

To determine robustness of the proposed scan registration method in a forested environment, an experiment analogous to the one described in Section 4.5.2 is performed using



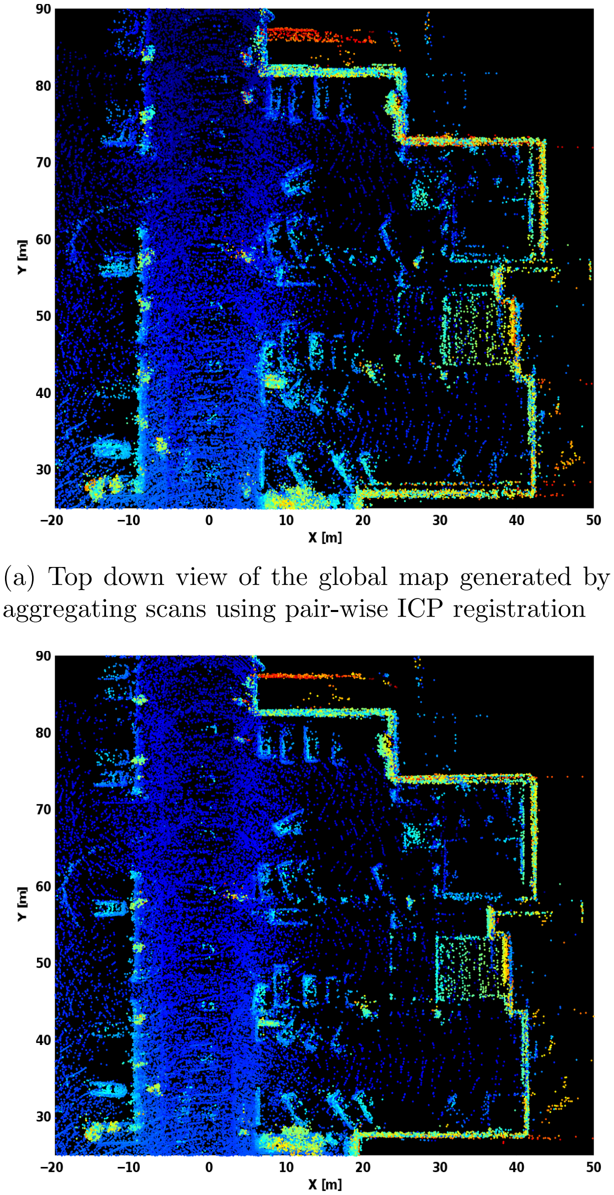
(a) Absolute translation error

(b) Absolute rotation error

Figure 4.9: Box-plots of the error distributions for each algorithm for the experiment from the Ford campus data set. (a) Absolute translation error. (b) Absolute rotation error

the WPI data set. In the experiment, only the two best performing algorithms, SGC-NDT and G-ICP, from the previous urban scenario experiment are presented, as it was determined that the ICP and NDT algorithms were unable to converge consistently and did not provide any meaningful results for comparison. The experiment is performed in a similar manner to the urban scenario experiment, where scans are pair-wise registered using the competing algorithms and then aggregated into a global map. Each algorithm performs pair-wise scan registration using every 50th scan produced by the LIDAR, as the robot moved slowly through the environment. The registration is initialized with a parameter estimate of zero translation and zero rotation. The experiment demonstrates performance in situations where odometry estimates are highly inaccurate, such as in GPS denied environments, or on vehicles where wheel slip or other factors make accurate state estimation difficult. Since accurate ground truth is not available for the WPI data, the global maps are evaluated based solely on the crispness measure. The crispiness measures for the WPI data set are presented in Table 4.1.

The crispness measure illustrates that G-ICP was not able to provide accurate scan registration results as compared to the SGC-NDT algorithms. The inaccuracy is most likely because G-ICP relies on accurate surface normals to determine the correct point correspondences, which is difficult to achieve in a forested environment. The SGC-NDT algorithm models environmental features such as trees and shrubs as a mixture of Gaussian distributions, thus good registration results are achievable in sparse environments with poor local surface structure. Visually, the result of aggregating the registered scans is presented

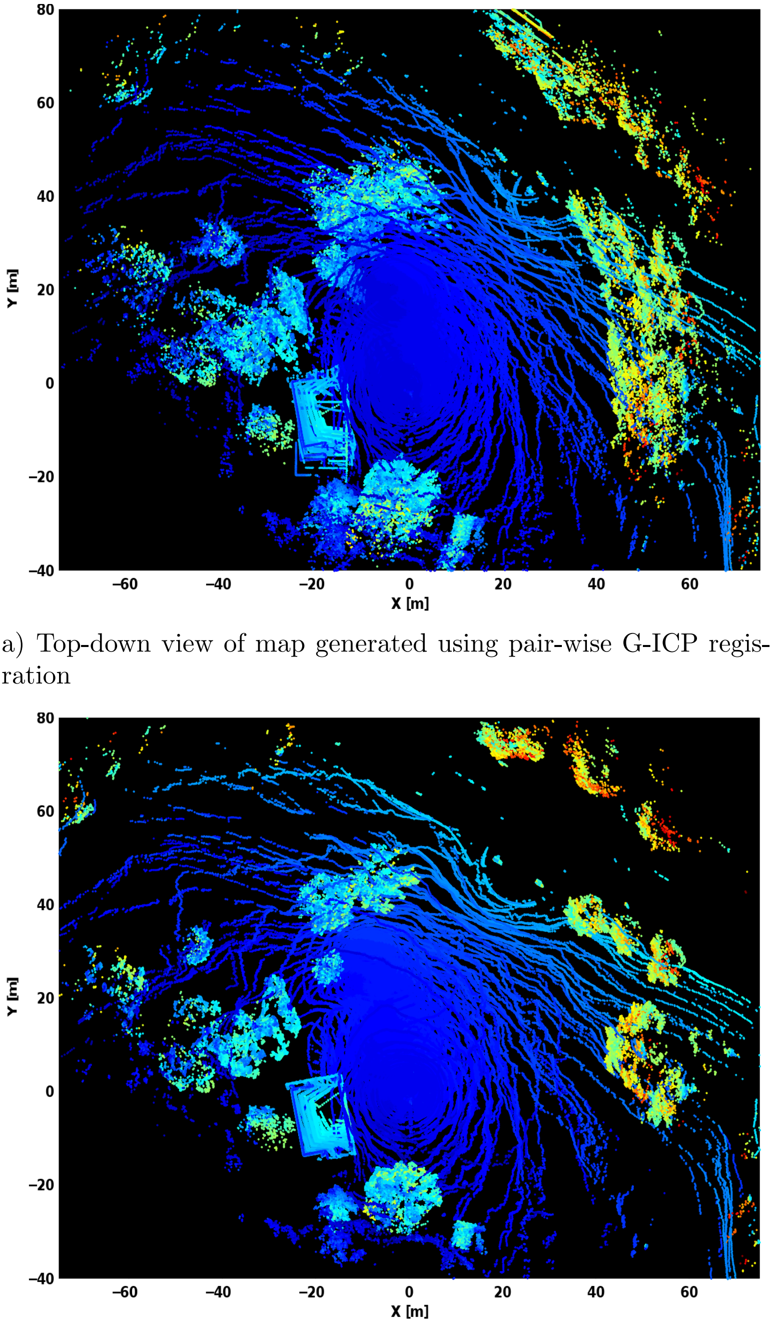


(a) Top down view of the global map generated by aggregating scans using pair-wise ICP registration

(b) Top down view of the global map generated by aggregating scans using pair-wise SGC-NDT-P2D registration

Figure 4.10: Resulting maps generated by aggregating scans using pair-wise (a) ICP and (b) SGC-NDT-P2D algorithms. Note the sharper corners, walls, cars and other features from the SGC-NDT map compared to the ICP generated map. The map is coloured based on height from the ground.

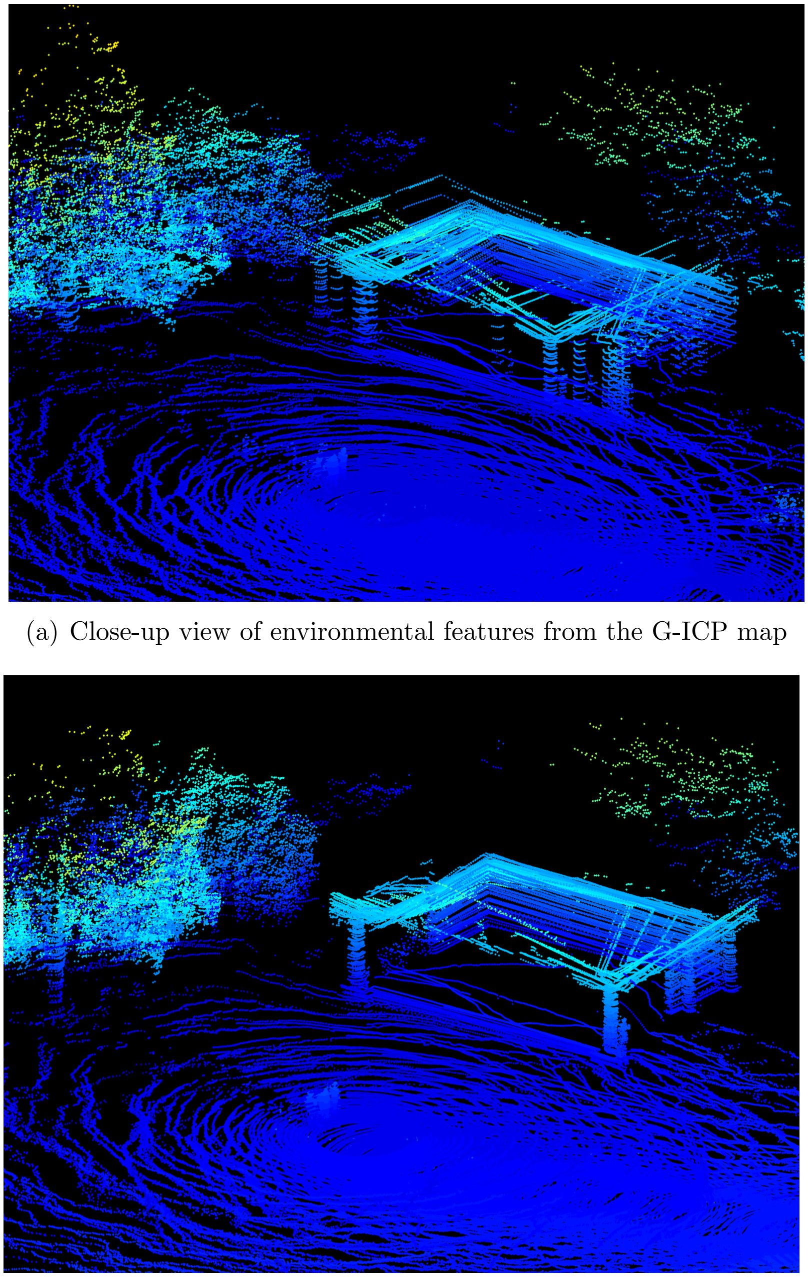
in Figures 4.11 and 4.12, which show the maps from top down view of the traversed area, as well as a close up view of the environmental features. The G-ICP map shows significant drift and occasionally poor convergence to local minima, while the SGC-NDT map maintains very crisp features. The crispness of the features can be clearly identified by inspecting the alignment of the structural columns seen in the close-up view.



(a) Top-down view of map generated using pair-wise G-ICP registration

(b) Top-down view of map generated using pair-wise SGC-NDT-P2D registration

Figure 4.11: Top-down view for the resulting global maps using the SGC-NDT-P2D and G-ICP algorithms. Note the crispness in features and reduced map drift for SGC-NDT, compared against the G-ICP map.



(a) Close-up view of environmental features from the G-ICP map

(b) Close-up view of environmental features from the SGC-NDT-P2D map

Figure 4.12: Close up views for the resulting global maps using the SGC-NDT-P2D and G-ICP algorithms. Note the crispness in features and reduced map drift for SGC-NDT, compared against the G-ICP map.

Chapter 5

Conclusion

Advances in autonomous vehicle technologies are improving the ability for unmanned systems to accomplish tasks without human intervention. Autonomous vehicles are valuable in many applications, ranging from deployment in hazardous and hostile environments for search and rescue, to exploration, mapping, and sample collection on Martian surfaces. The effectiveness of these unmanned vehicles is limited by their ability to perceive and interact with the environment, thus SLAM for autonomous systems remains an active and important research topic. High level mission tasks, such as drivability analysis and obstacle avoidance, require SLAM algorithms which generate dense map representations. Typically, these dense maps are constructed using point clouds, and are incorporated into the SLAM algorithm using scan registration techniques. Although a great deal of work has been done in the area of scan registration, there are many issues with the existing approaches.

The well known ICP algorithm solves the scan registration problem by minimizing the Euclidean distance of corresponding nearest neighbour points between two overlapping point clouds. However, since ICP operates at the point level, the algorithm does not take into account the underlying surface of the environment which is sampled to generate the point cloud. Instead of performing matching using individual points, G-ICP approximates the surface normal at each point from its local neighbourhood. The G-ICP cost function assumes that the point location has low uncertainty along the surface normal direction and high uncertainty along the surface direction, thus penalizing correspondences for points with inconsistent surface normals. Finally, the G-ICP approach explicitly takes sensor noise and sampling characteristic into account, thus improving the convergence basin and convergence rate when compared to ICP. The major shortcomings of both the ICP and G-ICP approach is the generation of the nearest neighbour correspondences, as the step is generally computationally expensive. The NDT approach is desirable because it does not require the computation of explicit point correspondences, but rather represents the scan as a set of Gaussian distributions and solves a nonlinear optimization to determine the transformation parameters. Although promising, the NDT approach suffers from major shortcomings: the convergence basin of the NDT algorithm is poor, the optimization of the NDT cost function is not guaranteed to converge, and the registration performance in sparse, outdoor environments is weak.

To overcome the limited convergence basin of NDT with 2D data, this work presents a modified algorithm known as the Multi-Scale K-Means NDT (MSKM-NDT). In contrast to the standard NDT algorithm, the proposed method divides the scan into clusters at multiple scales using k-means clustering, instead of discrete grid-based divisions. The optimization step is then performed for each scale, and the solution of the optimization from the i^{th} scale is used to initialize the optimization for the $i^{\text{th}+1}$ scale. The k-means clustering of the scan generates fewer Gaussian distributions when compared to the standard voxel based grid division method, which allows for the evaluation of the cost function across all distributions in a computationally efficient manner. The evaluation of the cost across all clusters removes the discontinuities of the cost function found in the standard NDT algorithm caused by the cell boundaries, resulting in a smooth and differentiable cost function which guarantees convergence of the optimization at each scale. The coarse-to-fine optimization approach is experimentally shown to avoid the local minima associated with the standard NDT algorithm and improve the convergence basin. Experiments also demonstrate that the MSKM-NDT algorithm is able to generate high quality SLAM solutions using laser data with large initial transformations between scan pairs.

As the k-means clustering does not scale well for use with higher dimensional point clouds, a different approach is required for use with 3D data. This work presents the Segmented Greedy Cluster NDT (SGC-NDT), which specifically addresses the 3D scan registration problem for sparse outdoor environments, where objects such as trees, shrubs and bushes make accurate surface normal computation difficult. The proposed method segments the ground points from the point cloud and greedily clusters the non-ground points to generate Gaussian distributions representing natural features within the environment. Similar to the MSKM-NDT approach, the NDT scan registration algorithm is modified to evaluate the cost function across all of the Gaussians, resulting in a smooth cost function that guarantees convergence of the optimization. The SGC-NDT approach is applied to both point to distribution and distribution to distribution NDT, and is shown to produce accurate results in both urban and sparse, forested environments.

Although this work demonstrates that intelligent clustering methods can be used effectively to improve the accuracy and convergence properties of the NDT algorithm for 2D and 3D point cloud data, open issues remain with both MSKM-NDT and SGC-NDT that require further investigation:

MSKM-NDT: A thorough comparison against G-ICP is required to determine the relative accuracy and convergence performance between the two algorithms. A full analysis of computational complexity is also required to fully understand the trade-offs between the computational overheard imposed by k-means clustering, and the computational savings achieved by evaluation of the cost function using fewer Gaussian distributions. A real-time implementation of MSKM-NDT is desirable, as it will enable larger and more numerous data sets to be evaluated efficiently, which should lead to a better understanding of the types of environmental characteristics to which the method is best suited. Finally, as the test data presented in this thesis is somewhat limited, evaluation on a wide range of environments and determination of specific failure cases remain areas of future work.

SGC-NDT: An investigation of the computational complexity of SGC-NDT is also required, as the GP regression and greedy clustering add overhead to the scan registration process. It is advantageous to determine how the GP regression and greedy clustering scale with the number of points in the point cloud and under what circumstances the overhead of segmentation and clustering justifies the computational savings from evaluation of the NDT cost function using fewer Guassians. The SGC-NDT algorithm assumes that the ground points contribute little to the scan registration process, however, it would be useful to determine a method to quantify the contribution of the ground points and determine a threshold on this value. The ability to quantify the contribution of the grounds point should lead to a better understanding of under what conditions the ground segmentation approach is valid, and the environments to which the SCG-NDT algorithm is suitable. Analogous to MSKM-NDT, a real time implementation will enable larger data sets to be evaluated effectively, and should present a better understanding for the conditions under with the SGC-NDT approach is viable. Finally, as the NDT algorithm fundamentally assumes that the clusters are well modelled by Gaussian distributions, the development of a clustering method which actively seeks to generate clusters based on the similarity of the clustered points to a Gaussian distribution is left as future work.

In addition to the open issues associated with the presented work, known limitations to the algorithms also exist, mainly due to the assumptions made in each of the clustering processes. The MSKM-NDT algorithm clusters the point cloud using k-means, which assumes that the number of partitions are known before performing clustering. The SGC-NDT algorithm requires that the natural features are separated by the ground, and assumes that the ground points do not contribute significant information to the scan registration process, which is valid only when the ground has very few distinguishing features. The limitations associated with making assumptions about the clustering methods motivate future work in the development of a generalized clustering algorithm for NDT, which is able to partition and cluster the scan without having to make any assumptions about the environment. One possible method to achieve generalized clustering would be through the use of an *information theory* based approach, which could provide a well established framework with which to evaluate point cloud data and systematically extract the subset of points which contribute the most information to the scan registration process. By performing scan registration using only the points that provide significant information, accurate registration results could be achieved with significantly less computational burden.

Scan registration remains a challenging problem for the SLAM community. The presented approaches perform scan registration with superior convergence and accuracy characteristics compared to standard NDT, and greatly improve upon the performance of ICP and G-ICP methods in situations and environments for which surface normals are difficult to calculate or are unreliable (outdoors, cluttered indoors, forested, sparse point cloud). The improvements to scan registration methods presented in this work are a necessity for the success of graph SLAM algorithms, and through advancements in robotic mapping, will allow for numerous future applications with large scale societal impact.

References

[1] M. Agrawal, K. Konolige, and M. Blas. Censure: Center surround extremas for realtime feature detection and matching. *Computer Vision–ECCV 2008*, pages 102–115, 2008.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3):346–359, 2008.

[3] D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1999.

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

[5] P. Biber and W. Strasser. The normal distributions transform: a new approach to laser scan matching. In *Intelligent Robots and Systems, Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2743 – 2748, Las Vegas, NV, USA, Oct 2003.

[6] R. Bogdan and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–4, Shanghai, China, May 2011.

[7] J. Bohren, R.B. Rusu, E.G. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mosenlechner, W. Meeussen, and S. Holzer. Towards autonomous robotic butlers: Lessons learned with the PR2. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5568–5575, Shanghai, China, May 2011.

[8] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg. Globally consistent 3-D mapping with scan matching. *Robotics and Autonomous Systems*, 56(2):130–142, 2008.

[9] A. Broggi, P. Medici, P. Zani, A. Coati, and M. Panciroli. Autonomous vehicles control in the VisLab intercontinental autonomous challenge. *Annual Reviews in Control*, 36(1):161 – 171, 2012.

[10] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2724 –2729, Sacramento, CA , USA, Apr 1991.

[11] A. Das and S. Waslander. Scan registration with multi-scale K-means normal distributions transform. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 2705–2710, Vilamoura, Algarve, Portugal, Oct 2012.

[12] B. Douillard, A. Quadros, P. Morton, J. Underwood, M. De Deuge, S. Hugosson, M. Hallstrom, and T. Bailey. Scan segments matching for pairwise 3D alignment. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3033–3040, St. Paul, MN, USA, May 2012.

[13] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. On the segmentation of 3D lidar point clouds. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2798–2805, Shanghai, China, May 2011.

[14] M. Dunbabin and L. Marques. Robots for environmental monitoring: Significant advancements and applications. *Robotics & Automation Magazine, IEEE*, 19(1):24–39, 2012.

[15] H.F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal of Robotics and Automation*, 4(1):23–31, 1988.

[16] A. Golovinskiy, V.G. Kim, and T. Funkhouser. Shape-based recognition of 3D point clouds in urban environments. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2154–2161, Oct 2009.

[17] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[18] G. Grisetti, C. Stachniss, and W. Burgard. Nonlinear constraint network optimization for efficient map learning. *Intelligent Transportation Systems, IEEE Transactions on*, 10(3):428–439, 2009.

[19] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using depth cameras for dense 3-D modeling of indoor environments. In *International Symposium on Experimental Robotics*, 2010.

[20] T.M. Howard, A. Morfopoulos, J. Morrison, Y. Kuwata, C. Villalpando, L. Matthies, and M. McHenry. Enabling continuous planetary rover navigation through FPGA stereo and visual odometry. In *IEEE Aerospace Conference*, pages 1–9, Big Sky, MT, USA, 2012.

[21] M. Inaba, N. Katoh, and H. Imai. Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. In *Annual Symposium on Computational Geometry*, pages 332–339. ACM, 1994.

[22] S. Julier and J. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997.

[23] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *Robotics, IEEE Transactions on*, 24(6):1365–1378, 2008.

[24] K. Klasing, D. Wollherr, and M. Buss. A clustering method for efficient segmentation of 3D laser data. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4043 –4048, May 2008.

[25] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234. IEEE, 2007.

[26] K. Konolige, J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. *The International Journal of Robotics Research*, 29(8):941–957, 2010.

[27] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[28] R. Kümmerle, D. Hahnel, D. Dolgov, S. Thrun, and W. Burgard. Autonomous driving in a multi-level parking structure. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3395–3400, Kobe, Japan, May 2009.

[29] M. Li, W. Li, J. Wang, Q. Li, and A. Nüchter. Dynamic veloSLAM–preliminary report on 3D mapping of dynamic environments. In *IEEE Intelligent Vehicles Symposium*

(IV), Workshop on Navigation, Perception, Accurate Positioning and Mapping for Intelligent Vehicles, Alcala de Henares, Spain, 2012.

[30] R.A. Lindemann, D.B. Bickler, B.D. Harrington, G.M. Ortiz, and C.J. Voothees. Mars exploration rover mobility development. *Robotics Automation Magazine, IEEE*, 13(2):19 –26, June 2006.

[31] S. Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[32] D.G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

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

[34] M. Magnusson, T. Duckett, and A. Lilienthal. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827, Oct 24 2007.

[35] M. Magnusson, A. Nüchter, C. Lörken, A.J. Lilienthal, and J. Hertzberg. 3D mapping the Kvarntorp mine–a field experiment for evaluation of 3D scan matching algorithms. In *IEEE Conference on Robotics and Automation (ICRA)*, 2009.

[36] M. Magnusson, A. Nüchter, C. Lörken, A.J. Lilienthal, and J. Hertzberg. Evaluation of 3D registration reliability and speed – a comparison of ICP and NDT. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3907–3912, Kobe, Japan, 2009.

[37] M.W. Maimone, P.C. Leger, and J.J. Biesiadecki. Overview of the Mars Exploration Rover autonomous mobility and vision capabilities. In *IEEE International Conference on Robotics and Automation (ICRA) Space Robotics Workshop*, Rome, Italy, 2007.

[38] F. Moosmann, O. Pink, and C. Stiller. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion. In *2009 IEEE Intelligent Vehicles Symposium*, pages 215–220, Xi'an, China, June 2009.

[39] F. Moosmann and C. Stiller. Velodyne SLAM. In *IEEE Intelligent Vehicles Symposium*, pages 393–398, Baden-Baden, Germany, June 2011.

[40] R. Murphy, J. Kravitz, S. Stover, and R. Shoureshi. Mobile robots in mine rescue and recovery. *Robotics Automation Magazine, IEEE*, 16(2):91 –103, June 2009.

[41] F. Naderi, D.J. McCleese, and J.F. Jordan. Mars exploration. *Robotics Automation Magazine, IEEE*, 13(2):72 –82, June 2006.

[42] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, Basel, Switzerland, Oct 2011.

[43] P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schroeter, L. Murphy, W. Churchill, D. Cole, and I. Reid. Navigating, recognizing and describing urban spaces with vision and lasers. *The International Journal of Robotics Research*, 28(11-12):1406–1433, 2009.

[44] A. Nüchter. *3D Robotic Mapping - The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom*, volume 52 of *Springer Tracts in Advanced Robotics*. Springer, 2009.

[45] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM – 3D mapping outdoor environments. *Journal of Field Robotics*, 24(8-9):699–722, 2007.

[46] G. Pandey, J. McBride, and R. Eustice. Ford campus vision and lidar data set. *International Journal of Robotics Research*, 30(13):1543–1552, November 2011.

[47] J. Roberts and R. Walker. Flying robots to the rescue [competitions]. *Robotics Automation Magazine, IEEE*, 17(1):8 –10, March 2010.

[48] R.B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3212–3217, Kobe, Japan, 2009. IEEE.

[49] R.B. Rusu, Z.C. Marton, N. Blodow, and M. Beetz. Persistent point feature histograms for 3D point clouds. In *International Conference on Intelligent Autonomous Systems (IAS)*, Baden-Baden, Germany, July 2008.

[50] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Robotics: Science and Systems (RSS)*, volume 25, pages 26–27, Seattle, USA, June 2009.

[51] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. volume 1, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[52] M. Song, F. Sun, and K. Iagnemma. Natural feature based localization in forested environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3384–3390, Vilamoura, Algarve, Portugal, 2012.

[53] B. Steder, R.B. Rusu, K. Konolige, and W. Burgard. NARF: 3D range image features for object recognition. In *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 44, 2010.

[54] T. Stoyanov, M. Magnusson, H. Andreasson, and A. Lilienthal. Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. *The International Journal of Robotics Research*, 31(12):1377–1393, 2012.

[55] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 3$^{\text{rd}}$ edition, 2005.

[56] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7-8):693–716, 2004.

[57] C. Tongtong, D. Bin, L. Daxue, Z. Bo, and L. Qixu. 3D LIDAR-based ground segmentation. In *First Asian Conference on Pattern Recognition (ACPR)*, pages 446–450, Beijing, China, Nov 2011.

[58] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, M. Gittleman, S. Harbaugh, A. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.

[59] B. van Arem, B. Jansen, and M. van Noort. Smarter and better - the advantages of intelligent traffic. Technical Report 2008-D-R0996A, The Netherlands Organisation for Applied Scientific Research (TNO), 2008.

[60] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte. Gaussian process modeling of large-scale terrain. *Journal of Field Robotics*, 26(10):812–840, 2009.

[61] R. Weisbin, G. Rodriguez, S. Schenker, H. Das, S. Hayati, T. Baumgartner, M. Maimone, I. Nesnas, and A. Volpe. Autonomous rover technology for Mars sample return.

In *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, pages 1–10, Noordwijk, The Netherlands, June 1999.

[62] T. Whelan, J.B. McDonald, M. Kaess, M.F. Fallon, H. Johannsson, and J.J. Leonard. Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, July 2012.

[63] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. Technical report, INRIA-Sophia Antipolis, 1992.