Improving Scan Registration Methods Using Secondary Point Data Channels

by

James Servos

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, 2014

Abstract

Autonomous vehicle technology has advanced significantly in recent years and these vehicles are poised to make major strides into everyday use. Autonomous vehicles have already entered military and commercial use, performing the dirty, dull, and dangerous tasks that humans do not want to, or cannot perform. With any complex autonomy task for a mobile robot, a method is required to map the environment and to localize within that environment. In unknown environments when the mapping and localization stages are performed simultaneously, this is known as Simultaneous Localization and Mapping (SLAM).

One key technology used to solve the SLAM problem involves matching sensor data in the form of point clouds. Scan registration attempts to find the transformation between two point clouds, or scans, which results in the optimal overlap of the scan information. One of the major drawbacks of existing approaches is the over-reliance on geometric features and a well structured environment in order to perform the registration. When insufficient geometric features are present to constrain the optimization, this is known as geometric degeneracy, and can be a common problem in typically environments. The reliability of these methods is of vital importance in order to improve the robustness of autonomous vehicles operating in uncontrolled environments.

This thesis presents methods to improve upon existing scan registration methods by incorporating secondary information into the registration process. In this work, three methods are presented: Ground Segmented Iterative Closest Point (GSICP), Color Clustered Normal Distribution Transform (CCNDT), and Multi Channel Generalized Iterative Closest Point (MCGICP). Each method provides a unique addition to the scan registration literature and has its own set of benefits, limitations, and uses.

GSICP segments the ground plane from a 3D scan then compresses the scan into a 2D plane. The points are then classified as either ground-adjacent, or non-ground-adjacent. Using this classification, a class constrained ICP registration is performed where only points of the same class can be corresponded. This results in the method essentially creating simulated edges for the registration to align. GSICP improves accuracy and robustness in sparse unstructured environments such as forests or rolling hills. When compared to existing methods on the Ford Vision and Lidar Dataset, GSICP shows a tighter variance in error values as well as a significant improvement in overall error. This method is also shown to be highly computationally efficient, running registrations on a low power system twice as fast as GICP, the next most accurate method. However, it does require the input scans to have specific characteristics such as a defined ground plane and spatially separated objects in the environment. This method is ideally suited for outdoor sparse environments and was used with great success by the University of Waterloo's entry in the NASA Sample Return Robot Challenge.

CCNDT provides a more adaptable method that is widely applicable to many common environments. CCNDT uses point cloud data which has been colorized either from an RGBD camera or a joint LIDAR and camera system. The method begins by clustering the points in the scan based on color and then uses the clusters to generate colored Gaussian distributions. These distributions are then used to calculate a color weighted distribution to distribution cost between all pairs of distributions. Exhaustively matching all pairs of distributions creates a smooth, continuous cost function that can be optimized efficiently. Experimental validation of the CCNDT method on the Ford and Freiburg datasets has shown that the method can perform 3D scan registrations more efficiently, three times faster on average then existing methods, and is capable of accurately registering any scans which have sufficient color variation to enable color clustering.

MCGICP is a generalized approach that is capable of performing robustly in almost any situation. MCGICP uses secondary point information, such as color, intensity, etc., to augment the GICP method. MCGICP calculates a spacial covariance at each point such that the covariance normal to the local surface is set to a small value, indicating a high confidence matching surfaces, and the covariance tangent to the surface is determined based on the secondary information distribution. Having the covariance represented in both the tangential and normal directions causes non-trivial cost terms to be present in all directions. Additionally, the correspondence of points between scans is modified to use a higher dimensional search space, which incorporates the secondary descriptor channels as well as the covariance information at each point and allows for more robust point correspondences to be determined. The registration process can therefore converge more quickly due to the incorporation of additional information. The MCGICP method is capable of performing highly accurate scan registrations in almost any environmental situation. The method is validated using a diverse set of data including the Ford and Freiburg datasets, as well as a challenging degenerate dataset. MCGICP is shown to improve accuracy and reliability on all three datasets. MCGICP is robust to most common degeneracies as it incorporates multiple channels of information in an integrated approach that is reliable even in the most challenging cases.

The results presented in this work demonstrate clear improvements over the existing scan registration methods. This work shows that by incorporating secondary information into the scan registration problem, more robust and accurate solutions can be obtained. Each method presented has its own unique benefits, which are valuable for a specific set of applications and environments.

Acknowledgements

First and foremost I would like to express my deepest gratitude to my supervisor Dr. Steven Waslander, without whom this thesis would not have been possible. His guidance and advice have been invaluable to me and have enabled me to become the capable, competent robotiscist and researcher I am today.

I would also like to thank everyone in the Waterloo Autonomous Vehicles Lab for all of their assistance and support through out this experience. It was always a pleasure to come into the WAVELab and discuss current and future projects with everyone. I'd like to a give special thanks to Arun Das who helped mentor me when I was just starting out and continued his support throughout.

The University of Waterloo Robotics Team has been a huge part of my life since my early undergraduate career and has help keep my passion for robotics alive as well as generously allowing me to play with all of the fun toys that they have. I will miss the team greatly but I know it will continue to shine on long into the future.

Finally, I would like to thank my family for always being there to support me, my friends who have stuck by me thoughtout this adventure and listened patiently to all of my grumbling over the years, and Amy for her continuous support through sleepless nights and stress filled days, I could not have done it without you.

Dedication

This thesis is dedicated to my family and friends, for all their love, friendship and support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Autonomous mobile robots are fast becoming a staple of everyday life. This technology can make a major impact on the world today in a vast range of fields and industries. Mobile robots are ideally suited for performing tasks which humans find dirty, dull or dangerous, such as driving, repetitive industrial tasks, search and rescue tasks and operations in hazardous environments. Today there already exist robots performing these tasks to some degree, blazing the trail for autonomous robotics in to move into everyday life.

Driverless cars have been making great strides in recent years as they move towards mass availability in the near future. Google has had their fleet of autonomous cars driving smoothly throughout Californian streets along side urban traffic and are becoming more robust by the day [24]. The VisLab deployed four vehicles to autonomously drive from Parma, Italy to Shanghai, China, a route coving over 13,000 km where previous maps where not available [8]. Most OEMs now have functional prototype autonomous vehicles, and new features which extend self-driving capabilities reach the consumer market every year. Many experts speculate that we will see the commercial availability of these autonomous vehicles by 2020 and that their introduction could have significant benefits on the safety and efficiently of our roads [73].

Another major area where autonomous robotics can have a great impact is in search and rescue operations in hazardous environment. Currently, many search and rescue operations put additional humans at risk in highly uncertain, dangerous environments in order to perform rescue operations. By deploying autonomous robots, not only is the risk to human life reduced significantly but these robots can perform these tasks more easily and effectively than many of their human counterparts. An excellent example of robots working in an environment hazardous to humans presented itself during the Fukushima nuclear reactor disaster. The radioactive disaster area was extremely hazardous, so robots were sent in to survey, assess, and even begin decontaminating the disaster zone [46]. However, their limited ability to map and navigate reliably meant that humans still had to go into the contaminated reactors, and even had to rescue one of the robots. This is not the first time robots have been used in disaster areas but it does demonstrate the new wave of disaster response methods utilizing the assistance of robotic systems.

Industrial applications of autonomous systems are leading to what has been called the automation revolution [57], where many menial industry jobs are being replaced with intelligent robotic systems. Robots have been taking over menial assembly line work for the past several decades, however with recent developments these robots are no long constrained to be bolted to the floor and are free to move around facilities as needed. This new freedom has allowed robots to take on an ever increasing range of tasks from automated warehousing solutions [23], to janitorial duties [35]. Robots are no longer limited to just the factory or the warehouse and have begun to see significant use in harsher environments such as mines [43] where they survey and transport materials, or often the more chaotic environment, the home, where personal robots can do anything from vacuum the floor to clean the gutters [31].

It is clear that with advancements in autonomous technology we will continue to see robots proliferate into all aspects of our everyday life, however there are still many challenges to be faced before this day can come. Any autonomous system is extremely complex and needs to solve many difficult problems before it can operate effectively. A typical basic autonomous system must include some form of task planning, object recognition, control computation, localization, and environmental mapping, while most require even more specialized systems. All of these systems are very active fields of research, however, localization and mapping continues to be the focus of many robotics researchers, being a cornerstone requirement for any autonomous system. Localization and mapping are highly interconnected and dependent tasks, one trying to determine where the robot is relative to a given environment and the other trying to map an environment given the robot's current location. This tightly coupled problem, when solved together, is known as simultaneous localization and mapping (SLAM).

In order to perform high level mission autonomy tasks such as vehicle path planning, obstacle avoidance and exploration, a strategy to perform Simultaneous Localization and Mapping (SLAM) is required. Unmanned vehicles are being deployed to perform a wide variety of missions in a diverse range of settings with many of these environments having never been mapped before, whether it be the bottom of the ocean [75] or a disaster zone [45]. Additionally, in a GPS and magnetometer denied environment, the error in the pose of the vehicle can grow unbounded due to imprecise odometry and sensor noise. Thus, specific measures should be taken to re-localize against previously visited features. SLAM in a GPS and magnetometer denied environment commonly arises in indoor environments, which are generally well structured and allow for accurate localization and mapping using camera or laser based approaches [19, 21].

Simultaneous localization and mapping is performed when the robot does not have a preexisting map of the environment and must build a map and localize against it as it traverses through the space. This operation is a keystone component of any mobile robotic system. SLAM is a particularly difficult problem because it is cyclically dependent, with mapping dependent on localization and vice versa. Many researchers have developed methods of solving the SLAM problem, however, there are still many issues which are unresolved and significant work is left to be done before a definitive solution is proposed.

1.1 SLAM Algorithms

The goal of the original research developing the first SLAM algorithms was to establish a statistical relationship between landmark locations and vehicle positions [17, 63]. The work showed the evolution of the estimates of the location of different landmarks as additional observations were recorded, and there correlation to the estimate of the vehicle pose, as well as the positions of other landmarks. Following this original formulation of the SLAM problem, a new solution was proposed using the Bayesian estimation framework [67]. A common implementation of the Bayesian SLAM framework uses an Extended Kalman Filter (EKF) to estimate the states of the landmarks and vehicle positions and is therefore known as EKF-SLAM. EKF-SLAM is a relatively intuitive concept but suffers from many drawbacks, such as the rapid growth in computational complexity as feature count increases, sparse feature-based maps which are difficult to use for mission planning, challenging feature correspondence which can lead to poor estimates, and limited robustness due to the linearization at each update. EKF-SLAM is an online SLAM method, this means that it only stores the current estimates of the vehicle location as well as the landmark locations. In contrast more recent SLAM algorithms, known as full or graph SLAM solutions, maintain the feature locations and vehicle positions over the entire trajectory.

Graph SLAM methods are currently among the most common methods for maintaining a globally consistent and accurate map for localization [39, 26, 36, 51, 32, 49]. Graph SLAM methods store all of the current and previous information in the form a pose graph which is capable of correcting even the oldest information as new information becomes available. The optimal graph is maintained by performing a non-linear graph optimization to identify a locally optimal solution over the entire history of measurements and vehicle motions.

Almost all SLAM algorithms conform to the same basic structure. Each has two main components, the front-end and the back-end. The front-end in responsible for generating correspondences between sets of measurements typically representing the motion of the vehicle relative to some set of landmarks. The front-end often represents these correspondences in the form of a pose graph where the nodes of the graph represent the poses of the vehicle or landmark and the edges represent measurements between these nodes. The back-end of the SLAM structure is responsible for using the information provided by the front-end and optimizing over the whole set of information to generate a globally consistent solution. Back-end graph optimization solutions are a relatively solved problem with open source solvers such as TORO [22] and g2o [38] freely available. Current research on the back-end of SLAM focuses on making more robust solvers to account for bad measurements from the front-end. Conversely, this thesis focuses on making front-end solutions which are extremely robust, produce minimal divergent measurements, and provide the most accurate information possible for the back-end optimization.

In recent years SLAM algorithms have progressed significantly, enabling faster more robust localization and more refined mapping. The most common state-of-the-art methods either use images from cameras or 2D/3D points typically from a LIDAR or depth camera. These methods can be further classified as either dense or sparse methods based on whether they use a subset of features or incorporate all of the sensor information. Each type of method has positive and negative aspects which make them suited for different sets of applications. Dense SLAM methods are ideal for inspection tasks in close-quarters environments. Dense methods are capable of creating highly detailed maps of small to medium sized areas with great accuracy. Sparse methods often are significantly more computationally efficient and can be performed on platforms with minimal computational capacity or when solutions must be obtained at high frequency such as if the vehicle is moving at high speed. Sparse methods are also more memory efficient and scale more smoothly into larger scale mapping applications.

With many modern sensors, such as LIDAR, RGBD cameras, and stereo cameras, providing robots with reliable 3D point information of their environment, scan registration techniques have become a prevailing solution to the front-end SLAM problem. Scan registration aligns consecutive scans to obtain the rotation and translation of the system relative to its environment and allows for the aggregation of point cloud data. These aggregate maps provide detailed environmental information which can be used for path planning and obstacle avoidance. In this thesis improvements on current state-of-the-art scan registration algorithms are presented which result in more robust and accurate registrations. Methods presented in this work use a combination of secondary information, often from images, with point information from LIDARs and depth cameras in order to obtain more reliable and robust results than existing methods.

1.1.1 Image-Based SLAM

Cameras are a very common sensor for many autonomous vehicles, as they provide a dense set of measurements in a relatively narrow field of view, however, by networking many cameras together a larger field of view can be obtained while maintaining a high resolution. Multiple cameras can also be used in a stereo pair to extract range information from the environment. Stereo matching can be highly computationally expensive and is highly dependent on the environmental texture present. Cameras have the additional benefit of having relatively high frame rates allowing for high speed SLAM applications. A relatively new addition to robot sensing technology is the range camera or RGBD camera. These cameras use active IR illumination to more accurately determine range to physical objects. Although these sensors can provide dense range information they still have limited range and do not function reliably outdoors.

Sparse SLAM methods using cameras are often performed by extracting image features from either stereo or monocular images. Two common image features used for this purpose are Speeded-up Robust Features (SURF) [3] and Scale Invariant Feature Transform (SIFT) [42] features. SURF and SIFT are very similar feature extraction techniques which provide scale invariant image features which can be matches between consecutive frames. Many other image features have been developed, each performing better or worse in different situations, however from a SLAM perspective all provide the same correspondence information necessary to calculate the SLAM solution.

A well known sparse camera method, known as RGBD-SLAM [18], has been developed by Endres et al. and uses RGBD images collected from a Kinect or stereo camera to track SIFT or SURF image features in 3D space and optimize the motion transform using a graph SLAM back-end. This approach can produce accurate maps but can slow down significantly as the graph grows and can struggle is non-feature-rich environments. Stereo vision has also been successfully used for real-time SLAM by Konolige et al. [37], who augmented stereo feature matching with sparse bundle adjustments and are able to provide accurate pose information in rough outdoor environments over large trajectory lengths with relatively low position drift. Using a single monocular camera can be difficult for SLAM applications as it gives no range information, however, Klein and Murray [34] developed the Parallel Tracking and Mapping (PTAM) algorithm which matches image features from a monocular camera using the ego-motion of the camera itself to generate 3D landmark and position estimates. PTAM was also extended by Tribou [70] to use multiple non-overlapping cameras and is additionally capable of independently determining scale, something the original PTAM algorithm could not accomplish.

Recently, with the development of more powerful computers and parallel processing techniques such as CUDA [52], it has become feasible to perform dense SLAM using all the information captured in an image frame. These methods attempt to use every pixel in an image to estimate vehicle pose. Therefore they can be extremely computationally expensive, however, with parallelization are often capable of running in real time.

Several dense camera SLAM methods have been proposed including Dense Tracking and Mapping (DTAM) [48], created by Newcombe et al., which uses a single monocular camera to generate 3D reconstructions of environments. This method creates accurate reconstructions but can not determine scale without an outside source and is ideal for close up applications. Newcombe et al. also proposed the dense SLAM method KinectFusion [47], which uses a Kinect sensor to generate dense 3D maps of small areas. The method was further extended by Whelan et al. [74] to allow larger working volumes and include color. These methods rely on GPU parallelization in order to be able to perform in real time, and can be extremely memory intensive as the mapped region grows.

While camera based techniques have the advantage of providing long distance bearing measurements, they suffer from poor range and field of view when compared to the laser based approaches. Camera based methods typically require detailed close quarters information in order to perform effectively and struggle to handle larger, more homogeneous environments. They generally have significant difficulty in making large quick movements, due to motion blur and tracking, and using long range measurements, due to poor triangulation, such as are expected in outdoor operations. Networking many cameras together can improve the field of view of the system but often at the cost of computational performance.

1.1.2 Point Based SLAM

Laser scanners (LIDAR) have become a key tool for many robotics applications. Laser scanners provide highly accurate 3D information at long ranges which is invaluable for building high quality maps for use with planning or obstacle avoidance algorithms. The use of LIDAR has been proposed to overcome the field of view and point cloud density limitations of stereo vision [20, 7], with recent extensions and experimental results demonstrating consistent mapping results over large areas [68]. Laser scanners can come in many forms, some providing dense high accuracy information, while others provide sparse 360 degree information at higher rates. The ILRIS high-accuracy laser scanner can produce highly detailed and accurate 3D point cloud maps of the environment however only at significant computational cost. Conversely, the Velodyne HDL-64E [29], is a popular choice for autonomous vehicles and gives a relatively sparse 3D 360 degree scan at 10 Hz.

Sparse LIDAR methods use 3D features such as points, planes, lines, or subsections of the scans to perform high speed accurate SLAM. Examples of 3D features can include Point Feature Histograms (PFH) [58], Viewpoint Feature Histograms (VFH) [59], or Normal Aligned Radial Features (NARF) [64]. Laser features can be used in a very similar manner to image features in that they provide a correspondence between consecutive scans. The inherent sparsity of some laser scanning technologies can make it very difficult to obtain accurate and descriptive features and thus the sparse methods can fail.

Feature based methods can have several drawbacks, both for camera and laser based methods, as demonstrated in [28]. Since feature based methods rely on matching a small number of points, noise in the 3D location can cause significant errors. Additionally, a small number or even a single false correspondence can cause catastrophic failure. This can make feature based methods unreliable particularly in dynamic environments where a single feature could move and distort the entire map.

An alternate approach which uses aggregated data is the Multi-Level Surface (MLS) maps [56, 39, 71], which models a point cloud with a collection of patches. The patches are generated by binning the point cloud data into fixed size columns, parallel to the height axis of the vehicle. The height information from the point cloud is used to create patches within each column, where each patch models surfaces at differing heights. The orientation of the patches can then be used to classify points as traversable or non-traversable. This classification system allows scan registration algorithms to constrain point correspondences between scans. Although conceptually attractive, the MLS mapping approach has several draw back including added computational complexity in the aggregation stage as well as the inability to accurately represent some features, particularly those which are not vertical projections.

Dense laser SLAM methods attempt to align entire scans with one another. Most often this is performed using a form of scan registration algorithm such as the Iterative Closest Point (ICP) method [4]. Dense laser based methods have been shown to be very successful as demonstrated by the numerous autonomous driving platforms which use these methods, such as the DARPA Urban Challenge vehicles or the Google driverless car [72, 33, 41, 24].

1.2 Scan Registration as a SLAM Front-End

One of the first methods proposed for solving the scan registration problem was the Iterative Closest Point (ICP) method introduced by Besl and McKay [4]. The ICP algorithm minimizes the Euclidean distance between nearest neighbour points in the two scans to find the relative transform. Taking advantage of the locally planar nature of most environments, Chen and Medioni [9], proposed a point to plane based variant of ICP which penalizes the cost only normal to the surface of the environment. This approach mitigates the sampling error seen in point to point ICP which assumes that points correspond exactly between scans. Recently, Segal et al. developed Generalized-ICP (GICP) [60] which, using a probabilistic framework, generalized the ICP method and introduced a plane to plane approach with improved performance over the previous versions.

A parallel school of scan registration techniques, first suggested by Biber and Strasser in [5] in 2D and extended to 3D by Magnusson et al. [43], is the Normal Distribution Transform (NDT). The NDT method divides the base scan into a grid and calculates a Gaussian distribution from the points in each grid cell. Scans are then aligned by minimizing the point to distribution cost of each point in the input scan to the distribution in the target scan within the corresponding cell. Further expansion by Stoyanov et al. [65] demonstrated that using a distribution to distribution cost function could improve scan registration results over ICP and point to distribution NDT.

Many robots use scan registration in some form as the front-end method of their SLAM system. Scan registration has becoming a staple for autonomous vehicles due to its robustness, accuracy, simplicity, and reliability. It can be performed in real time and provides a relatively dense reconstruction of the environment when necessary for planning on control.Since the typical scan registration method provide the relative transform between two scans it makes it very easy to integrate into a back-end system. The graph can easily be build by setting each scan location as a node and adding edges as the relative scan registration transforms between each pair of scans. The integration of a scan registration front-end into a graph optimization back end and its use in a complete robotic system is discussed in detail in Chapter 6. However, one of the major issues with all geometric based scan registration algorithms is geometric degeneracy.

Geometric degeneracy occurs when the features in an environment are positioned in such a way that the scan registration optimization is degenerate in one or more of the optimization parameters. This occurs when there are not sufficient objects in the environment to fully constrain the optimization. Two common examples of this situation are long corridors or large open spaces. In large corridors the scan registration can localize laterally within the corridor but the longitudinal distance travelled can not be determined. Similarly in a large open space the vertical position from the ground plane can be determined but there are no features to constrain the registration in the x-y plane. This type of degeneracy can be a major problem for many robots compounded by the fact that these two types of situations are relatively common in everyday environments.

1.2.1 Colorized Scan Registration

The classic scan registration formulations use only the 3D point information to calculate point correspondences, distributions, and to perform the registration. However, many sensors, or combinations of sensors, can provide additional information for each point such as intensity or color. The additional information can help to improve registration accuracy, convergence rate, and solve many structural ambiguities.

With laser scanners becoming common place several authors have proposed not only using the 3D points returned by the scanner but also the intensity values which many scanners also produce. Levinson and Thrun [40] use 2D probabilistic intensity maps and a histogram filter to localize in an urban environment. This method, although successful in certain environments, completely ignores geometric information and assumes sufficient intensity information is always available.

Color information has also become commonly applied to 3D scans. Cameras and depth sensors can be extrinsically calibrated such that points can be associated with corresponding image pixels and colorized. This sensor combination has been used by many authors to attempt to improve the performance various SLAM algorithms.

Color ICP [30] attempts to improve ICP performance by using the additional color channels to perform the nearest neighbour search in a higher dimension. This method showed improved point correspondence results but did not change the underlying scan registration method. A colorized version of NDT has also been proposed by Huhle et al. [28], which uses color based kernel functions to generate a Gaussian mixture model such that each voxel then contains a color based mixture of Gaussians. Other methods include [55], which initializes GICP using image features, and [27], which augments NDT using a small set of image feature correspondences as a secondary error function.

The addition of color information into the scan registration formulation is able to mitigate some of the major issues present in the classic scan registration algorithms as well as improve accuracy and runtime in some cases. The work presented in this thesis builds upon these colorized approaches and generalizes the methods into a robust framework for reliable scan registration.

1.3 Contributions

The goal of this research is to improve the accuracy, robustness, computation time and convergence of scan registration methods used for the front-end of SLAM by incorporating secondary information into the registration formulation. The work focuses particularly on feature poor, challenging environments, such as sparse forests and large open spaces, where current state-of-the-art methods tend to fail. This thesis presents three novel scan registration methods which improve upon the basic scan registration methods to provide significant improvement in overall reliability. These methods include:

- Ground Segmented Iterative Closest Point (GSICP) [13]
- Color Clustered Normal Distribution Transform (CCNDT) [62]
- Multi Channel Generalized Iterative Closest Point (MCGICP) [61]

Ground Segmented ICP was developed to solve the problem of scan registration in sparse unstructured environments. Typical scan registration methods function very well in structured environments with solid surfaces and edges which can be registered against, however, in unstructured environments with little to no flat surfaces modern scan registration methods have significant problems. Good example of unstructured environments are sparse forests or dunes which have no sharp edges. Foliage tends to give very noisy measurements due to the inconsistent measurements from the various leaves that make up the structure and similarly an environment made up of a collection of dunes, or hills can give vastly different measurements depending on the cross section which is scanned. The GSICP algorithm combats these problems by first taking a 3D laser scan and segmenting the ground plane from the scan. Next the scan is compressed into a 2D representation. From this 2D representation points are classified as either ground-adjacent or non-ground-adjacent. Using this classification information, points are corresponded only within the same class. This results in generating simulated edge feature in the environment which the scan registration algorithm can more accurately match. Because this method is compressing the scans into 2D and classifying the data before correspondences are calculated, the method is very computationally efficient and can be performed robustly at high speed. Tests performed on the Ford dataset show that GSICP has 60% less deviation in error, and a 44% improvement in mean error over the next best performing method, GICP. Tests also show that GSICP is capable of performing a scan registration over twice as fast as GICP and NDT, and consistently faster then ICP. Although the method performs very well, it requires a specific structure for the scans in order to function correctly. Specifically, it requires spatially separated geometric objects with a segmentable ground plane and minimal variation outside of the horizontal plane. Although most outdoor robotics applications using a 360 degree scanner will meet these requirements it does limit the algorithms applicability. This algorithm also falls victim to geometric degeneracy, a common issue with all geometric based scan registration algorithms.

Color Clustered NDT addresses several of the issues faced by GSICP and other geometric scan registration techniques. CCNDT calculates a full 3D scan registration solution but unlike typical NDT solutions is able to maintain excellent computational performance and robustness. The CCNDT algorithm uses combined color and geometric information, such as would be generated by an RGBD camera or joint calibrated LIDAR and camera system, to perform the registration. The algorithm first clusters points based on color and Euclidean distance and uses these clusters to generate a set of Gaussian distributions. The colored distributions are then used to perform a color weighted distribution to distribution optimization to extract the optimal transform. Unlike standard NDT, by using the clustering approach a minimal set of distributions is needed to represent the scan and therefore the optimization can be performed significantly more efficiently. Additional because the optimization is color weighted and there are a small number of distributions, the cost function can be calculated between every pair of distributions resulting in a smooth and continuous cost function. This smooth and continue cost function is less likely to become trapped in local minima and requires only a single iteration to optimize to the global minima. CCNDT is much more generally applicable than GSICP, but comes at the cost of additional computational complexity. The accuracy of CCNDT method is shown to be comparably, or marginally better then the existing methods on both the Ford and Freiburg datasets, however demonstrates significant improvements in computation time. Compared to GSICP, CCNDT takes approximately 50% longer to perform a registration, however, compared to other 3D registration methods, such as GICP and NDT, CCNDT runs more then three times faster on average. Furthermore, by incorporating color information into the scan registration process, CCNDT is also able to avoid many instances of geometric degeneracy by using matching colors. CCNDT can still fail in some cases as it relies exclusively on the existence of non-trivial variations in the coloring of the scene in order for the clustering to be effective. For scenes which do not have significant variations in color the optimization can once again become degenerate.

Multi-Channel GICP generalizes the incorporation of secondary information as well as geometric information in a seamless structure that utilizes both sources of information to their full extent but does not rely explicitly on either. This allows for the method to be generally applicable in any environment and able to perform reliably even in cases normally degenerate for other registration methods.. The MCGICP method generates a point covariance at every point in the scan which represents the confidence of the position of that point in each direction. Similar to GICP, the covariance normal to the geometric surface is set to a small value to represent a high confidence normal to the surface. Then the covariance tangent to the surface is determined using the secondary descriptor channels. In the case of MCGICP the descriptor channels do not need to be limited to just color and can represent any type of information. By having variations in the covariance in both the normal and tangential directions each point can contribute error terms in all directions according to the information it provides. It is also vital to ensure that points are correctly corresponded. Therefore the correspondence calculation is performed in a higher dimensional space which incorporates both geometric, descriptor, and covariance information. By using all three types of information for correspondence and registration optimization, the algorithm is able to converge reliably in a wide range of environments. MCGICP is highly robust against degeneracy, has increased accuracy compared to the other scan registration methods, and can provide reliable registrations in almost any environment. Experiments performed on the Ford and Freiburg datasets show accuracy improvements of up to 31% over existing methods. Experiments also showed significantly fewer outliers and a decrease in the standard deviation of error of 49%. Qualitative results also demonstrate notable improvements in map reconstruction in challenging and degenerate environments. The cost of this reliability is an increase in computational complexity. Although the method is shown to converge faster then standard GICP, method such as CCNDT and GSICP easily outperform MCGICP in terms of computation speed.

All of the proposed methods demonstrate improvements over the existing scan registration methods and each shows a specific case in which that method would be preferable. Whether it be the high speed 2D performance of GSICP, the 3D efficiency of CCNDT, or the high accuracy, robust, reliability of MCGICP, each method contributes a piece to complete the landscape of scan registration algorithms, and is suited for different subsets of the many robotics applications and environments in which a robot could be called on to operate.

Chapter 2

Scan Registration

Given a set of scans, typically in the form of point cloud data, scan registration is used to find the transformation which best transforms a scene scan to overlap with a reference scan. Scan registration algorithms define nonlinear, often discontinuous cost functions which are used in an optimization framework to determine the ideal transform. The assumption of these methods is that the global minimum of the defined cost function yields the optimal transformation that aligns the two scans. In this chapter the scan registration problem is defined in a general sense and the formulations of three modern scan registration algorithms, Iterative Closest Point (ICP), Generalized Iterative Closest Point (GICP), and Normal Distribution Transform (NDT), are presented. These three methods form the basis of the works presented in the following chapters.

2.1 Problem Formulation

Scan registration algorithms attempt to find the optimal transform between an input scan and a target scan. The optimal transform is qualitatively defined as the transform which best aligns the scans into a common coordinate frame. Given scan $A = \{a_i\}$, where $a_i \in \mathbb{R}^3$ for $i = 1, ..., N_A$, and scan $B = \{b_j\}$, where $b_j \in \mathbb{R}^3$ for $j = 1, ..., N_B$, the optimal transformation matrix, T, can be be defined as,

$$T = \begin{bmatrix} R_T & t_T \\ 0 & 1 \end{bmatrix}$$

where $R_T \in \mathbb{SO}(3)$ is the rotation matrix and $t_T \in \mathbb{R}^3$ is the translation vector.

The solution is typically computed by optimizing a specified score function, $\Gamma : \mathbb{SE}(3) \rightarrow \mathbb{R}$, given the point clouds A and B, to find the optimal transformation T. The optimization can be generally defined in the form:

$$T^* = \arg\min_T \Gamma(T) \tag{2.1}$$

The score function is specific to each algorithm and can take many different forms and in many cases is optimized in an iterative method before settling on the final solution. The following sections of this chapter will introduce the score function definitions for the three most commonly used scan registration methods: ICP, GICP, and NDT.

2.2 The Iterative Closest Point Algorithm

The Iterative Closest Point (ICP) algorithm is one of the most common methods for scan registration and is arguably the most straight forward of the existing algorithms. The ICP method uses the squared Euclidean distance between corresponding sets of points as the cost function used to evaluate the transformation.

Point correspondences are evaluated by assuming that the nearest neighbour points correspond to the same location in space. In practice, a KD-tree [44] can be used to determine nearest neighbours efficiently in both 2D and 3D cases. Additionally, a threshold, κ_d, is often defined such that only corresponding points with a Euclidean distance less than the threshold, $||a_i - b_i|| < \kappa_d$, are considered a correct correspondence. This threshold allows the exclusion of correspondences which are assumed to be from non-overlapping sections of the scans.

Given point clouds, A, and B, and assuming that these point clouds are organized such that corresponding points are given the same indices and include only points with correspondences within the threshold, the ICP scan registration optimization can be defined as:

$$T^* = \arg\min_T \sum_{i=1}^{N_C} ||a_i - b_i||^2 \tag{2.2}$$

This optimization is performed iteratively such that after each iteration, the point correspondences are updated and the optimization is repeated. After several iterations the

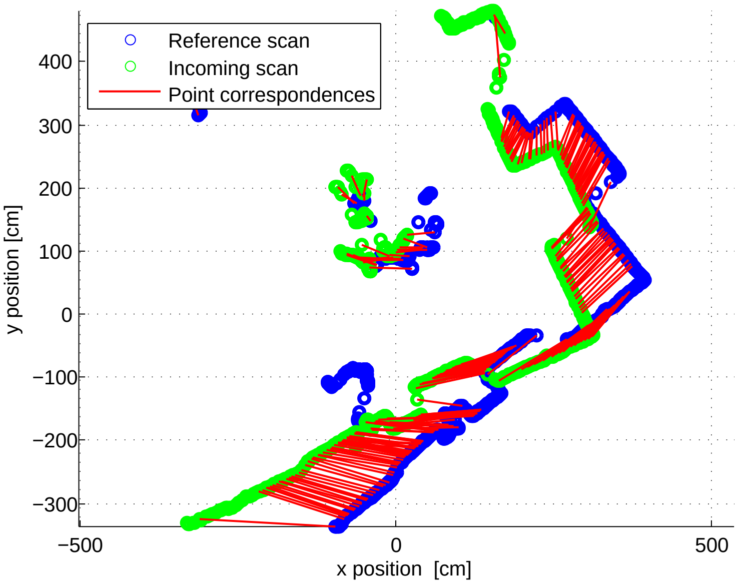


Figure 2.1: Example of the correspondences of the ICP algorithm in a 2D case. The red lines indicate the score which is being minimized. When registered the blue and green points would ideally overlap perfectly. [12]

optimization will ideally converge to the global minima, giving the optimal transformation between the two scans. An example of the ICP correspondences and optimization score minimization is shown in Figure 2.1.

The ICP optimization has been shown to have a closed form solution by Arun et al. [1] and as a result can be perform very quickly. However, due to the algorithm often requiring many iterations of the optimization the computation time of the algorithm can still be substantial. Additionally, Chen and Medioni [9], introduced point-to-plane ICP which penalizes the cost only normal to the local surface. This helped improve the convergence of ICP at the cost of computational complexity.

2.3 The Generalized Iterative Closest Point Algorithm

The Generalized Iterative Closest Point (GICP) method was developed by Segal et al. in [60] as a unifying framework of the previously proposed ICP methods. The GICP formulation uses a probabilistic framework to determine the error function and proceeds as follows.

First, it is assumed that the nearest neighbour correspondences have been calculated and scan A and scan B are indexed with corresponding points having the same indices and non corresponding points being removed. Using the probabilistic model it is assumed that the point clouds A and B are generated from an underlying set of distributions, where $a_i \sim \mathcal{N}(\hat{a_i}, C_i^A)$ and $b_i \sim \mathcal{N}(\hat{b_i}, C_i^B)$. Therefore given perfect correspondences and the correct transform, T^*,

$$\hat{b_i} = T^* \hat{a_i} \tag{2.3}$$

The difference between samples a_i and b_i is then defined as $d_i = b_i - Ta_i$. Given that a_i and b_i are drawn from independent Gaussian distributions, and given the correct transformation, d_i can be written as:

$$d_i \sim \mathcal{N}(0, C_i^B + T^* C_i^A (T^*)^T) \tag{2.4}$$

The transform is then solved for using maximum likelihood estimation (MLE) and simplified to the form

$$T = \arg\min_T \sum_i d_i (C_i^B + T C_i^A T^T) d_i^T \tag{2.5}$$

This formulation can be used to represent any of the standard forms of ICP including basic point to point as well as point to plane ICP. However GICP proposes a plane to plane model in which it is assumed that points are sampled from surfaces which are locally planar. In this model the covariance of a point is assumed to be small in the direction of the normal at that point and large in all other directions. This assumes that the points have little information to offer in the directions tangent to the plane. The covariance at every point, $q_i \in A \cup B$, in both A and B is calculated using an archetype covariance, C^G, define as

$$C^G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \epsilon \end{bmatrix}$$

where ϵ is a constant representing the covariance along the normal. The covariance at a point is then calculated as

$$C_i^W = (R_i^W) C^G (R_i^W)^T \tag{2.6}$$

where R^W is the rotation matrix which rotates ϵ to align with the surface normal, at point q_i.

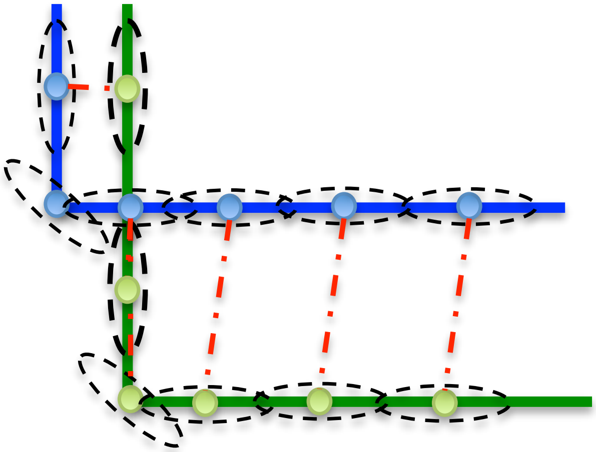


Figure 2.2: Example of the alignment of two scans (green and blue) using the GICP algorithm. As can be seen each point has a corresponding covariance which aligns with the surface of the scan (black). Corresponding points (red lines) with covariances which are aligned will result in a higher weighted cost in the minimal direction compared to those which are not aligned.

The local covariance, C_i^L, is calculated using the k nearest points to the query point q_i found using [44]. The local covariance approximates the model covariance in the region around the query point.The surface normal information for this method is then computed using principal component analysis (PCA) on the local covariance, C_i^L. The component with the lowest eigenvalue corresponds to the surface normal. In practice, the model covariance at a given point can be calculated using the singular value decomposition (SVD) of the local covariance,

$$C^L = USV^T \tag{2.7}$$

where the singular values are the diagonal elements of $S \in R^{n \times n}$ sorted in descending order and U and V are orthonormal matrices. In the singular value decomposition decomposition U is equivalent to the rotation matrix R_i^W and therefore S can be replaced by C^G, to compute, C_i^W.

Figure 2.2 shows an example of the alignment of two planar scans using the GICP algorithm. The plane to plane nature of GICP allows the scans to be align correctly without falling into a local minima which would most likely occur in the case of standard ICP.

2.4 The Normal Distributions Transform Algorithm

The Normal Distributions Transform is a common form of scan registration used in modern robotics applications. Many authors have proposed improvements to the algorithm but the underlying principles have remained constant. The NDT method attempts to model sets of points as Gaussian distributions and match these distributions either to points or corresponding distributions in the input scan.

The original NDT method begins by first dividing the point in the scan into a set of equally sized cells, c_i. For each cell a Gaussian distribution, $g_i = \mathcal{N}(\mu_{c_i}, \Sigma_{c_i})$, is calculated using the points from the scan which are within the boundaries of the cell. At the end of this process the entire scan is converted into a set of Gaussian distributions, $G = \{g_1, \ldots, g_{N_G}\}$. In the case of distribution to distribution NDT this step must be performed on both the target and input scan to obtain sets of distributions for the input, G^A, and target, G^B. The total number of distributions, N_G, is largely dependent on the size of the grid cells used to divide up the scan. Larger grid cells will result in fewer total Gaussians but small details can be lost in the Gaussian model. Small grid cells are capable of maintaining finer detail, however, this will result in a significant increase in the number of Gaussians and a corresponding increase in computation time. The size of the grid cells often has to be set and reset given different environmental conditions.

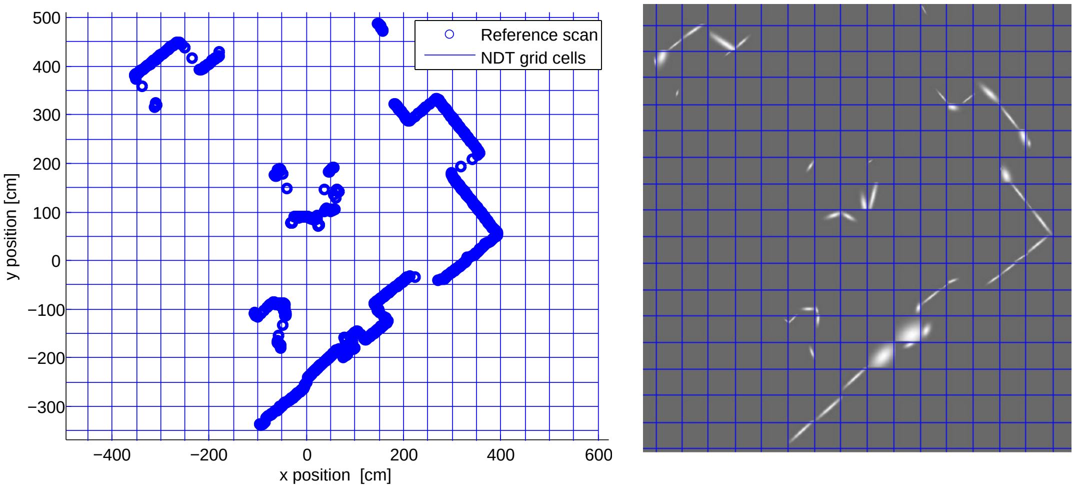
Next, the NDT cost function used to optimize the registration transform is calculated using the set(s) of Gaussians computed in the previous step. Both point to distribution (P2D) and distribution to distribution (D2D) have similar cost functions. The point to distribution cost function contribution, $J_{P2D}(x) : \mathbb{R}^3 \rightarrow \mathbb{R}$, of an individual point, x, is defined as:

$$J_{P2D}(x) = \exp(-(x - \mu_{c_i})^T {\Sigma_{c_i}}^{-1} (x - \mu_{c_i})) \tag{2.8}$$

where x is the point being evaluated and μ_{c_i} and Σ_{c_i} are the mean and covariance of the cell c_i to which x is corresponded. The sum of the individual point costs is then minimized:

$$T^* = \arg\min_T \sum_{i=1}^{N_A} J_{P2D}(Ta_i) \tag{2.9}$$

The resulting transform is the optimal relative movement of the sensor frame between the target and input scans that minimizes the cost function. Figure 2.3 shows an example of the NDT transformation of a 2D point cloud. Unfortunately, the overall cost function



(a) Original point cloud (b) NDT distributions

Figure 2.3: Example of the resulting distributions when applying the NDT transform on an example 2D point cloud. [12]

will be discontinuous. The correspondence of a point x to a specific distribution, c_i, is dependent on the transformation, T. As the transformation changes the points of the input scan change cells relative to the target scan and therefore correspond to a different distribution and cause a discontinuity when this change occurs. These discontinuities can be problematic for optimization as the derivative is not well defined and convergence can not be guaranteed. The cost function also demonstrates that in the case of geometric degeneracy the solution will not converge as there will be no sensitivity of the cost function to changes in the degenerate transform direction.

The NDT method can be tuned for accuracy or speed based on the size of the grid cells used, but the grid representation can be not ideal for some environments, and the cell size must be changed as the environment changes. To remove the need to tune the grid cell size, Magnusson et al. suggest a multi-scale approach [43] which runs multiple NDT optimizations at multiple scales. This approach significantly improves the convergence of the registrations but also significantly increases computation time in the process.

Chapter 3

Ground Segmented ICP

A large, unstructured environment, such as a forest, makes laser scan registration based approaches especially difficult since a typical point cloud from a laser scanner such as the Velodyne HDL-32E sensor is sparse and relatively noisy. Current scan registration algorithms generally make assumptions about point cloud data which are not valid in all cases. For example, 3D iterative closest point (ICP) methods assume every point is sampled in both scans, which is impossible, but in practice requires a high point density in order to provide accurate correspondences from nearest neighbour search [50]. Generalized ICP (GICP) improves ICP by using the underlying surface structure of the point cloud to relax the perfect point correspondence assumption [60]. The use of G-ICP requires the computation of surface normal information, which is difficult to perform accurately with noisy, unstructured point cloud data such as that generated from grass, trees, shrubs or rubble. The Normal Distributions Transform models the point cloud as a set of Gaussian distributions aggregating individual points and avoiding the limiting assumptions of both ICP and GICP, but since lasers scanners only detect the surface of objects, the Gaussian distribution is often an inaccurate representation of the points within a cell. As a result the algorithm suffers from poor convergence [11] and is not suitable for real-time implementation on a system with limited computation power.

The proposed method, Ground Segmented ICP (GSICP), is able to perform 2D registrations in large, unstructured environments using sparse point clouds. It should be noted that although the presented solution is well suited to operate in open, unstructured environments, the approach also works well on any navigable terrain where there are sufficient environmental features for scan registration. Since only a 2D drivability map is required for most ground vehicle applications this method performs only 2D registrations. Scans are first rotated based on vehicle pitch and roll estimates which is assumed to be available from an extended Kalman filter (EKF) or IMU, and then ground points are segmented and removed using a Gaussian process regression. Ground segmentation is justified since the ground points contribute little to the localization accuracy, compared to the majority of the natural features in the environment such as trees and buildings [10]. The remaining non-ground points are then used to generate a 2D top-down map of the environment.

A classification system is used to constrain the scan registration algorithm to compute point correspondences only between similarly classified points. Classification based scan registration approaches have been successfully implemented such as in the MLS mapping technique. More general point cloud segmentation and classifications are also possible, and allow for the application of ICP correspondences to be constrained between segments of similar proximity, shape and relative position between scans [15]. Class-constrained ICP (CC-ICP) methods have been shown to improve point correspondences and convergence rates, however general 3D point cloud segmentation is computationally expensive. In order to improve the robustness of our approach the remaining obstacle points, after the ground points have been removed, are classified as *ground-adjacent* or *non-ground-adjacent*.

Ground adjacency classification is well suited for this application, as it ensures edge features are maintained. Since the ground segmentation is already required to compute drivability, the determination of the non ground points requires no additional computation, and to further classify the points based on ground adjacency is computationally inexpensive. To localize the vehicle, a 2D CC-ICP registration is performed using the classified points.

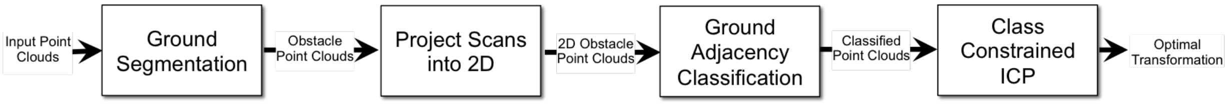


Figure 3.1: Flowchart showing the stages of the Ground Segmented ICP algorithm.

3.1 Ground segmentation and drivability analysis

The segmentation of the ground plane is an important aspect of this method. In the 2D scan registration case the ground points do not provide useful information and in fact hinder the convergence of registration algorithms. Segmenting the ground points also removes a significant number of points from the scan which helps improve the computational efficiency of the method. The ground segmentation algorithm used in this work is based off of the work first shown by Douillard et al [16] which uses a 2D Gaussian process to model the terrain and incremental sample consensus (INSAC) to identify ground points. However, a modified version of this method was introduced by Tongtong et al. [69] which is able to run faster than the original method and produce comparable results. This version improves runtime performance by first binning the scan into a polar grid then formulating a 1D Gaussian process using the points contains in each radial sector.

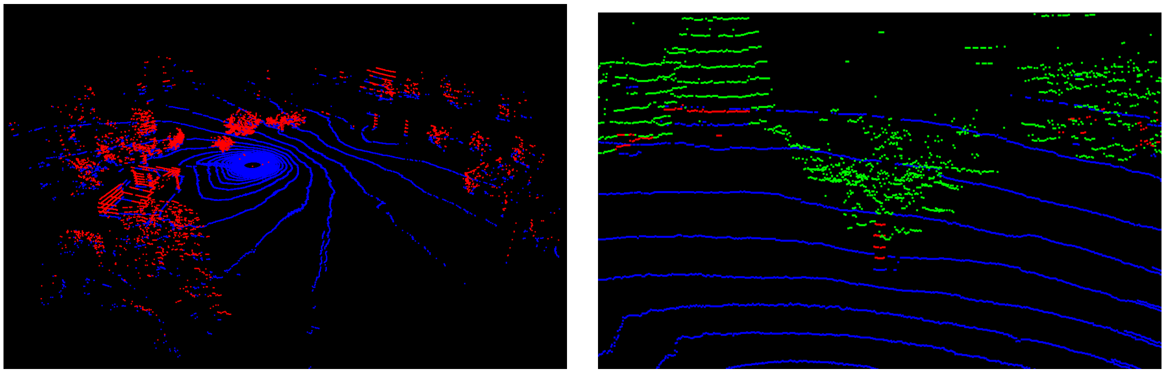
In order to create the 1D Gaussian process the point cloud, P, is first binned into a polar grid containing N_s angular sectors, and N_d linear range divisions. The set of points, γ_{ij}, contained within the bin s_i^j, are then given as

$$\gamma_{ij} = \{p \in P : (p^x, p^y) \in s_j^i\} \tag{3.1}$$

where s_i^j is the area of the j^{th} angular sector and i^{th} range bin. To generate the 1D signal from the points collected within each sector a set of prototype points must be selected from each bin that is assumed to be a reasonable representation of the ground. The simplest way to select these prototype points is to simply take the point, p, in each bin with the lowest z component. The 1D signal is then constructed using tuples of the xy range to each prototype point, $r = \|p^x + p^y\|_2$, and the corresponding height value p^z.

The Gaussian process for each sector can then be generated using these 1D signals. The process is defined using a mean function of zero and a squared-exponential function to model the covariance function. Using the Gaussian process the inliers of the model can be determined. The INSAC algorithm is used by incrementally adding inliers to the Gaussian process and regenerating the model until no more inliers can be added. Then the remaining points are classified as ground or object points by comparing their z components against the model of the corresponding sector. If the difference is less then a user defined threshold, κ_g, then the points are classified as ground. Figure 3.2-a presents the results of the Gaussian Process based ground segmentation approach on a typical point cloud from the Velodyne HDL-32E in a sparse forested environment containing a few additional manmade structures.

Once the points have been classified as obstacle or ground, they can be further classified as drivable or not-drivable for the vehicle. For each sector, obstacle points where the difference in the z-components between the obstacle point and the ground prototype point are less than the height of the vehicle, are classified as non-drivable points. The drivability classification is later used to generate a drivability map for the vehicle. Figure 3.2-b illustrates the classification results for a typical point cloud in a forested area. As is visible in Figure 3.2-b, it is important to classify obstacle points as drivable versus non-drivable. Many features can include overhanging sections which the vehicle is capable of



(a) Ground segmentation (b) Drivability segmentation

Figure 3.2: Point cloud segmentation results using Gaussian process method, which includes drivable obstacle points (green), non-drivable obstacle points (red) and ground points (blue). (a) Ground segmentation only. (b) Ground and drivability segmentation.

driving underneath. The drivability classification allows for the traversable sections to be accurately modelled, which is imperative for path planning purposes.

3.2 Ground adjacency classification

Obstacle points are classified as either *ground-adjacent* or *non-ground-adjacent*. In order to classify the points, a temporary local map is generated. The GSICP local map partitions a plane in $\mathbb{R}^2$ that is orthogonal to the gravity vector for the vehicle orientation at the current time-step. Assume that the obstacle point cloud has been transformed to compensate for the roll and pitch angle of the vehicle. Denote the local map as a set of fixed sized partitions, $\Psi = \{\varphi_1, \dots, \varphi_{N_\Psi}\}$, where $\varphi_i \subset \mathbb{R}^2$ is a cell in the local map, which contains N_Ψ total cells. The cells are non-overlapping, or $\Psi = \cup_{i=1}^{N_\Psi} \varphi_i$ and $\varphi_i \cap \varphi_j = \emptyset$, $\forall i, j$ where $i \neq j$. With the cell partitions, define the points from point cloud P whose projection falls within cell φ_j as

$$\gamma_j = \{p \in P : (p^x, p^y) \in \varphi_j\} \tag{3.2}$$

In order to classify the points within a cell, a collection of nearest neighbour cells are determined and evaluated. For the local map, $\chi_i \in \mathbb{R}^2$ and $\chi_j \in \mathbb{R}^2$ denotes the geometric mean, or centroid of the cells φ_i and φ_j, respectively. Define the distance between two cells φ_i and φ_j, using the distance function $d : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ between their respective centroids, as

$$d(\chi_i, \chi_j) = ||\chi_i - \chi_j||_q \tag{3.3}$$

When $q = 2$, the function d is the Euclidean distance between the centroids of the cells, however a distance based on the Manhattan norm, $q = 1$, or infinity norm, $q = \infty$, can also be used. For implementation, the infinity norm is used in order to extract a square sub-grid of cells as the nearest neighbours. Using the definition for the distance between grid cells given in Equation 3.3, the nearest neighbours, $\Phi \subseteq \Psi$, for a cell φ_i can be given as

$$\Phi = \{\varphi_j \in \Psi : d(\chi_i, \chi_j) < \kappa_\Psi\} \backslash \varphi_i \tag{3.4}$$

where κ_Ψ is a user defined threshold which determines the size of the neighbourhood of φ_j to consider. In order to classify the points within a cell as ground-adjacent or non-ground-adjacent, the nearest neighbour cells for a target cell are evaluated. A counter variable I_Ψ is used to count the number of ground-adjacent cells in the neighbourhood. If at least β_Ψ of the cells are ground-cells (i.e. they contain no points), then the points in the target cell are classified as ground-adjacent, otherwise the cell is labeled as non-ground-adjacent. The point classification method is summarized in Algorithm 1. Figure 3.3-a demonstrates the results of the ground adjacency classification algorithm for a typical example point cloud in a forested area. The classification illustrates that points which are generally contained within the boundaries of an object are classified as non-ground-adjacent while those on the boundary of the object are classified as ground-adjacent.

3.3 Class Constrained ICP

In GSICP, the classified obstacle points from the temporary local maps are registered into a consistent frame using a CC-ICP scan registration technique. The CC-ICP scan registration algorithm seeks to find the transform, T, such that the Euclidean distance between corresponding points of a model point cloud and a scene point cloud which has been transformed by T, is minimized. The ICP method treats nearest neighbour points as correspondences for each iteration of the minimization. For CC-ICP, the points in the classified scene sets P^S_{GA} and P^S_{NGA} may only correspond with the classified points in the model set P^M_{GA} and P^M_{NGA}, respectively. In order to further improve correspondences, the height values maintained in the cells are used to reject correspondences which do not agree with the modelled height distributions. If the z-component of a point from an obstacle point cloud which has been transformed by estimate T is not within one standard deviation

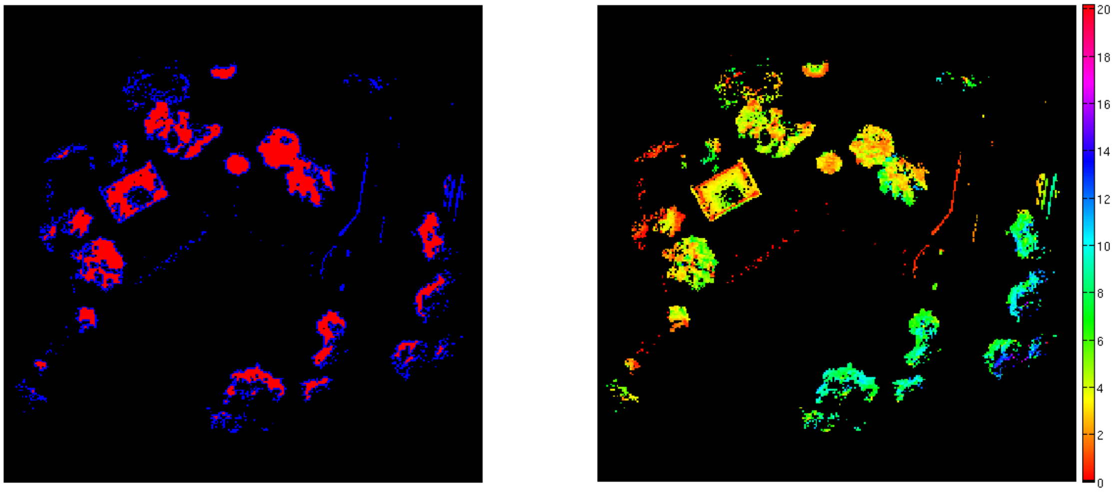
Algorithm 1 Using an input obstacle point cloud P_{obs}, return the scene point clouds for the ground-adjacent points, P^S_{GA} and the non-ground-adjacent points P^S_{NGA}

```
$P^S_{GA} \leftarrow \emptyset$
$P^S_{NGA} \leftarrow \emptyset$
$I_\Psi \leftarrow 0$
for all $\varphi_i \in \Psi$ do
    $\Phi \leftarrow \{\varphi_j \in \Psi : d(\chi_i, \chi_j) < \kappa_\Psi\} \backslash \varphi_i$
    for all $\phi \in \Phi$ do
        $\gamma \leftarrow \{p \in P_{obs} : (p^x, p^y) \in \phi\}$
        if $\gamma = \emptyset$ then
            $I_\Psi \leftarrow I_\Psi + 1$
        end if
    end for
    $\gamma \leftarrow \{p \in P_{obs} : (p^x, p^y) \in \varphi_i\}$
    if $I_\Psi \geq \beta_\Psi$ then
        $P^S_{GA} \leftarrow P^S_{GA} \cup \gamma$
    else
        $P^S_{NGA} \leftarrow P^S_{NGA} \cup \gamma$
    end if
end for
```

of the height distribution for the cell it is located in, the point correspondence is rejected. The strategy of using constrained nearest neighbour correspondences with height based outlier rejection results in the contributions for the CC-ICP minimization to come from the most likely point-pair correspondences. Figure 3.3 illustrates an example of the ground adjacency classification and provides a visualization of the aggregated height distributions within each cell.

3.4 Experimental Results

The Ground Segmented ICP algorithm was evaluated using the Ford Vision and LIDAR Dataset [53]. The dataset uses a Velodyne 64E laser scanner mounted on a truck as it traverses a typical urban roadway and includes ground truth information collected from an Applanix POS-LV 420 INS with Trimble GPS. The accuracy of the method is compared to the 2D versions of ICP, GICP and NDT, using the ground truth information from the dataset for reference. In order to perform a fair comparison the ground is segmented before



(a) Ground adjacency classification for the global map

(b) Mean cell height for the global map

Figure 3.3: Classification of the temporary local map (a) Classification of ground-adjacent (blue) versus non-ground-adjacent (red) cells. (b) Visualization of the cell mean height values (in meters)

passing the point cloud to all of the methods. The computation time of the algorithm is also analysed.

The registration accuracy is evaluated by performing scan to scan matching between each consecutive pair of scans over the entire dataset. The results are shown in the box plots in Figure 3.4, and demonstrate that the GSICP method is notably more accurate than the other methods. This increase in accuracy is particularly noticeable in areas of the dataset which are more unstructured, including mostly only trees and foliage to match against. The robustness of the algorithm can be seen by the marked decrease in the variance in the errors, meaning more consistent results are produced.

Over local areas of the dataset the GSICP method is very accurate at estimating the pose of the vehicle. Figure 3.5, shows a local stretch of the dataset in which the GSICP method is performing very well. However, due to the fact that only scan-to-scan matching is being performed the results will tend to drift over time. This is an artifact of using the scan-to-scan matching as no globally consistent frame is available at each iteration. This drift is present in all methods when using scan-to-scan matching. Figure 3.6 shows the complete map generated over the entire dataset. The drift of the map is obvious, however, it can be seen that locally the map is very good. In Chapter 6 the incorporation of this method into a full system including back-end graph optimization corrects this map for drift and can be seen in the Section 6.4.

Finally, computation time is compared. The comparison is performed on a computer with an Intel i7 Quadcore processor and 16GB of ram with each algorithm run in a single thread with no parallelization. Figure 3.7 shows a plot of the computation time for a series of scans for the existing algorithms as well as for GSICP. It can be seen from this plot that GSICP is capable of performing registrations faster and at more consistent speeds than the other 2D methods, all of which are significantly faster than the corresponding 3D methods. This increase in speed is due to the classification stage resulting in correspondences being faster to compute on a smaller subset of points and the consistency comes from the ability of GSICP to obtain correct correspondences more consistently than other methods.

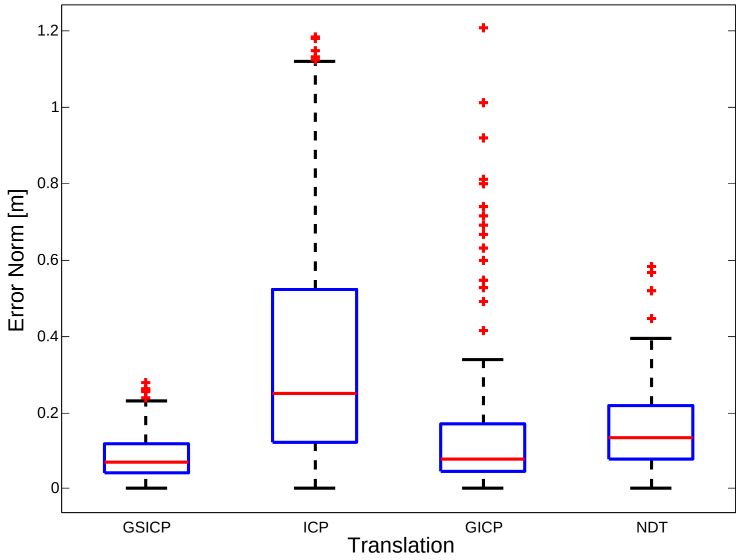
3.5 Conclusions

The GSICP method is ideally suited for 2D scan registration in sparse unstructured environments. The experimental results demonstrate that the method maintains high accuracy when moving through areas with both structured and unstructured sections and shows minimal failed convergences. The GSICP method is also significantly lighter than many of the other state of the art algorithms and is capable of running in real time on platforms with minimal computational power.

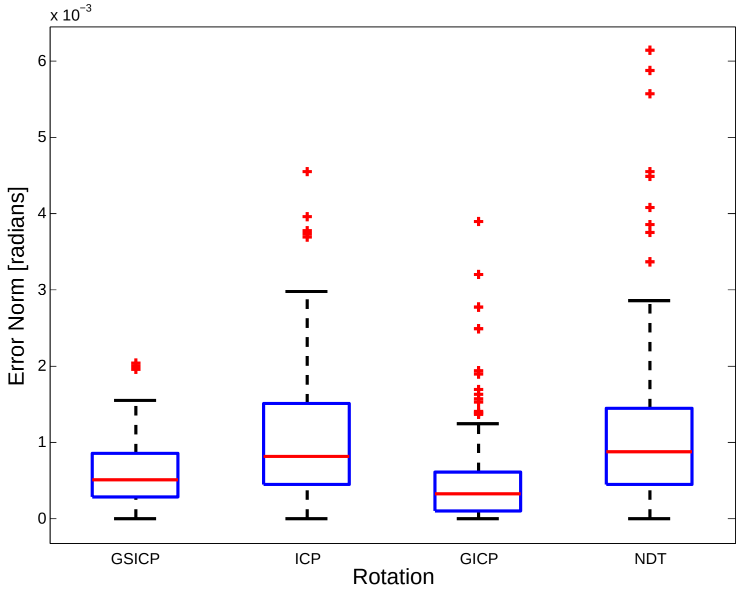
The GSICP method does however have several drawbacks. Foremost, the method is only capable of performing 2D scan registrations. Extension of the method into 3D is impractical as the ground adjacency classification becomes significantly more complex and the benefits of minimal computation time are lost. However, 2D registration is adequate for many ground vehicles traversing on relatively flat roadways or paths with minimal variations in height, roll and pitch.

The proposed method also requires distinct geometric features to exist in the horizontal plane. By compressing the scans into a 2D representation a significant performance boost is obtained in computation time, however, important features can be lost which can lead to a decrease in robustness in some environments. In order for the GSICP method to function correctly the scan must have several properties. First the scan must have a ground plane. This is not an issue for most outdoor vehicles with 360 degree scanners, however, indoors with a directional scanner this assumption my not be valid. Second, the scans must have spatially separated features in order for the ground adjacency classification to perform optimally. Finally, the scans must not have geometric degeneracies. Geometric degeneracies result when the scan does not constrain the optimization in one or more of the parameters. The above requirements limit the applicability of the method somewhat, although, in many cases robots operating outdoors do in practice meet all of these requirements. By using these assumptions the method is able to be optimized to perform quickly and accurately and is well suited for outdoor sparse environments.

The GSICP method is designed to meet a specific need; fast accurate 2D SLAM performed in outdoor environments with a sparse 3D sensor. The method is able to meet this need very effectively and was implemented and used on the University of Waterloo's entry in the NASA Sample Return Challenge in both 2013 and 2014.



(a) Translational Error



(b) Rotational Error

Figure 3.4: Accuracy comparison of 2D scan-to-scan matching results on the Ford dataset. Relative translational (top) and rotational (bottom) errors are shown. The red line represents the median, the blue box shows the 25th to 75th percentile, the whiskers extend to the max and min points within 1.5 times the inter quartile range, and the red crosses show outliers.

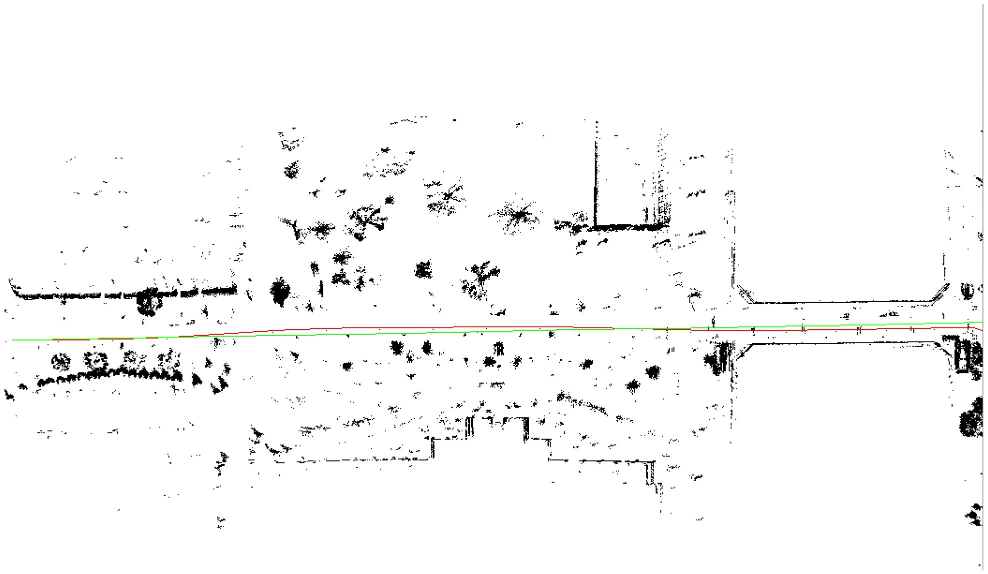


Figure 3.5: Local map and path for a subset of the Ford dataset. Black points represent the map, red line shows the GSICP calculated path, while the green line is the ground truth path.

Figure 3.6: Full map generated from the Ford dataset using only scan-to-scan matching with the GSICP method. This map shows obvious drift but the local structure is very accurate.

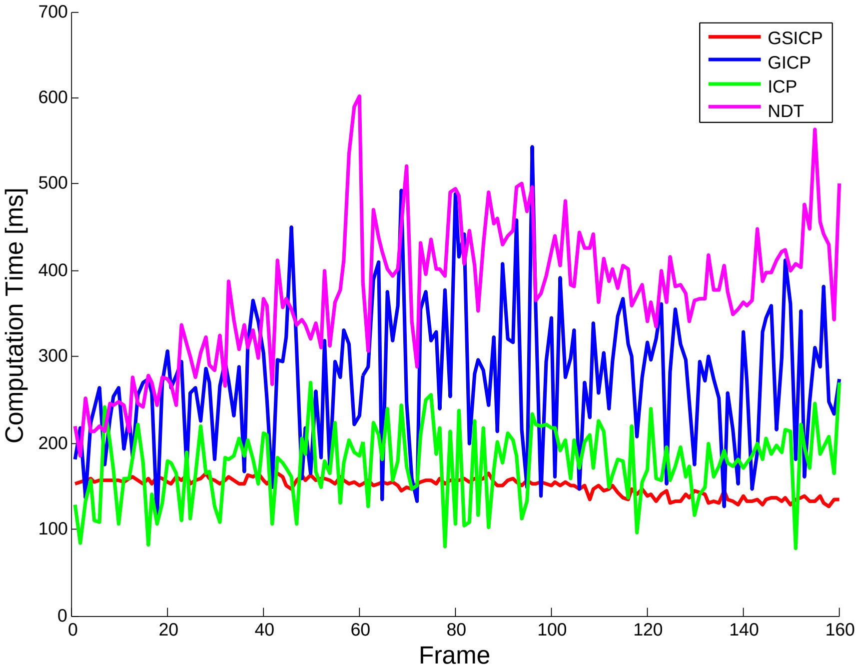


Figure 3.7: Computational performance of the GSICP algorithm as well as existing methods.

Chapter 4

Color Clustered Normal Distribution Transform

In general, scan registration methods can be quite time consuming as they require the cost function to be evaluated at every point in a scan or at least a large subset of points or derived distributions in order to obtain accurate results. This can be a problem for vehicles traveling at high speeds or in dynamic environments as the localization and mapping results may be delayed. 2D methods, such as GSICP, can perform registration very efficiently but require scans which provide enough information using only the horizontal plane. Vehicles experiencing significant changes in height, roll and pitch, such as an off-road, all-terrain vehicle, require the complete 3D registration information. Also, GSICP as well as many other existing SLAM methods, can fall victim to geometric degeneracies in spaces with minimal structure in the environment.

Recently, Das et al. [10] showed that by using a greedy clustering approach to generate surface Gaussians, a smooth and continuous NDT cost function could be created and could be evaluated significantly faster than previous methods. This approach however is not ideal in all environments as it requires object to be disjoint in 3D space in order to be accurately clustered. Conversely, Color NDT by Huhle et al. [28], uses color information to generate multiple distributions per cell and match distributions and points based on color weightings. This method helps reduce the effects of geometric degeneracies but can be slow if the grid is made small, or can be inaccurate if the grid is large.

In this chapter an improved NDT algorithm is introduced, which is capable of performing robust, accurate localization and mapping in a broad spectrum of possible environments and with a multitude of different sensors. The proposed method uses color and 3D information in a greedy clustering approach to cluster points and generate the requisite Gaussian distributions used for the optimization phase. The optimization function uses the calculated Gaussians to perform a weighted distribution to distribution cost between all pairs of distributions. The weightings are determined based on a normal distribution of the expected variance in color channels for the given sensor configuration. This approach gives a smooth, continuous cost function which can be optimized efficiently, reduces local minima, and avoids issues with geometric degeneracies when varied color information is present. This method also performs well in both dense, close range situations, as well as large scale sparse outdoor environments. With the addition of color information to the NDT framework, the method is capable of providing accurate results with minimal computation time.

The proposed method is evaluated using the Freiburg RGBD Dataset [66] as well as the Ford Vision and Lidar Dataset [53]. The Freiburg dataset includes RGBD information collected from a Kinect Sensor in several indoor environments as well as corresponding ground truth information. The indoor environment demonstrates an example of dense SLAM data of a small area with moderate sensor motions. The Ford dataset is generated from a vehicle mounted Velodyne laser scanner and Ladybug omnidirectional camera and demonstrates outdoor, road speed traversal of a typical unmanned vehicle. The Ford dataset evaluates the algorithm in a relatively sparse scenario where the sensors are moving at high speed.

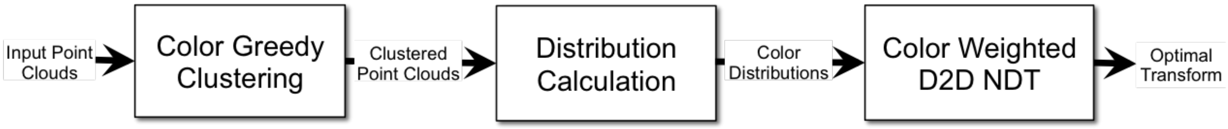


Figure 4.1: Flowchart showing the stages of the Color Clustered Normal Distribution Transform algorithm.

4.1 Greedy Color Clustering

The clustering method used in this work is based on the Color-Based Segmentation of Point Clouds method proposed by Zhan et al. [76]. The algorithm proceeds in two main stages. First a greedy region growing process is performed to make rough color consistent regions. Second a region merging and refinement stage is performed in order to refine the rough regions into the final segmentation result.

The greedy region growing stage follows a standard greedy clustering approach and uses color information to determine region inclusion. The set of clusters, $\zeta = \{c_1, \ldots, c_{N_c}\}$, where each cluster is a set of points, $c_i = \{p_1, \ldots, p_{N_{c_i}}\}$, is calculated by adding all points to the unlabeled set, $\mathcal{U}$, and incrementally removing a point from $\mathcal{U}$ to begin a new cluster and adding the point to the open set K. The function $\mathcal{K}(p, k)$ uses the kd-tree nearest neighbour algorithm defined in [44] to find the k nearest neighbours of a point p. The algorithm then checks the nearest neighbours of each point in K to determine inclusion in the current cluster, c_i, based on color difference to the initial cluster point. The function, $\delta(x, y) : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$, defines the color difference between point x and y as the Euclidean norm of the difference of the color vectors. If the nearest neighbour point is sufficiently similar, $\delta(x, y) < \kappa_c$ where κ_c is a threshold parameter, then the point is added to K. Once K is empty, the current cluster is added to the cluster set ζ, and a new cluster is started by selecting a new point from $\mathcal{U}$. The process is repeated until all points are labeled. The flow of the algorithms can be seen in Algorithm 1.

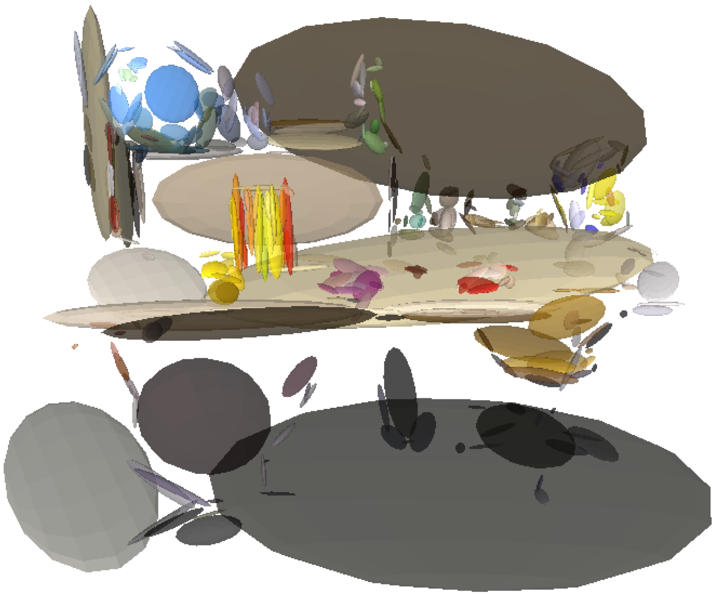
The second stage, region merging and refinement, uses the set of clusters, ζ, found in the previous stage and attempts to merge each cluster to its nearest neighbour clusters based on the adjacent cluster being of the same color. If two nearest neighbour clusters are of similar color, they are merged. Unlike the original method proposed in [76], for this work we do not explicitly merge regions below a certain threshold of points with its nearest neighbour. These regions are assumed to be outliers and are simply discarded from the set of clusters. It is important to note that since the computational complexity of calculating the cost function in Equation 4.3 scales quadratically with the number of Gaussians generated, N_c, it is helpful to limit the maximum number of distributions considered. A simple trimming of the smallest and largest clusters can be performed to stay within a defined maximum. It can be assumed that very large clusters provide minimal information in terms of overall alignment that the smaller clusters don't already capture and that the smallest clusters are most likely noise or areas that will be hard to match between scans.

The final stage is to convert the clustered points into a corresponding set of Gaussians, $G = \{g_1, \ldots, g_{N_G}\}$, where each Gaussian maintains the geometric mean, μ^g, as well as a color mean, μ^c, along with the geometric covariance, Σ. Figure 4.2 shows the clustering results on an indoor office scene. Note that the Gaussians are of various sizes and shapes and accurately represent the underlying structure of the scene with a minimal set of distributions.

Algorithm 2 Color Region Growing

```
$i \leftarrow 0$
$\zeta \leftarrow \emptyset$
$K \leftarrow \emptyset$
$\mathcal{U} \leftarrow P$
while $\mathcal{U}$ != $\emptyset$ do
  $K \leftarrow p \in \mathcal{U}$
  while $K$ != $\emptyset$ do
    $i \leftarrow i + 1$
    $p \in K$
    for $q \in \mathcal{K}(p, k)$ do
      if $\delta(p, q) < \alpha$ and $q \notin K$ then
        $K \leftarrow K \cup \{q\}$
        $c_i \leftarrow c_i \cup \{q\}$
        $\mathcal{U} \leftarrow \mathcal{U} \backslash q$
      end if
    end for
    $K \leftarrow K \backslash p$
  end while
  $\zeta \leftarrow \zeta \cup \{c_i\}$
end while
```

(a) Pointcloud of example scene before clustering



(b) Clustering results of the example scene

Figure 4.2: Clustering results of an example scene using the Color-Based Segmentation of Point Clouds method and converting point clusters to Gaussians

4.2 Color Weighted Clustered NDT

Using the sets of Gaussian distributions calculated in Section 4.1, for both the input scan, G_i, and the target scan, G_j, the transformation optimization proceeds using a modified cost function. The cost function uses a fully connected approach, where each distribution is evaluated against all other distributions, and incorporates a weighting based on the color difference between the two Gaussians, g_i and g_j, being evaluated. The color weights, λ_{ij}, are calculated given the covariance of the color channels, Λ, defined by the accuracy of the sensor used, and is defined as:

$$\lambda_{ij} = \exp(-\frac{1}{2}(\mu_i^c - \mu_j^c)^T \Lambda^{-1} (\mu_i^c - \mu_j^c)) \tag{4.1}$$

where $\mu_i^c \in \mathbb{R}^3$ is the mean color of Gaussian g_i. The Gaussians are therefore weighted such that distribution pairs with significant differences in color have a minimal impact on the cost. The final cost evaluation is simply the original distribution to distribution NDT cost weighted by λ_{ij} and calculated between all combinations of distributions. The cost for a single distribution to distribution registration, with a transformed difference of means, $d_{xy} = T\mu_x^g - \mu_y^g$ would be:

$$J_{D2D}(g_x, g_y) = \lambda_{xy} \exp(-\frac{1}{2} d_{xy}^T \left[T^T \Sigma_x T + \Sigma_y\right]^{-1} d_{xy}) \tag{4.2}$$

The minimization of this cost,

$$T^* = \arg\min_T \sum_{i=0}^{N_{G_i}} \sum_{j=0}^{N_{G_j}} J(g_i, g_j) \tag{4.3}$$

optimizes the alignment of the scans such that distributions of similar color should converge together and result in a more accurate, more robust registration result.

4.3 Experimental Results

The method proposed in this chapter was evaluated based on two datasets, the Freiburg RGBD Dataset and the Ford Vision and Lidar Dataset. Both of these dataset contain color and 3D information as well as ground truth position information. For both datasets the method is evaluated based on the accuracy of the transform compared to the ground truth data as well as qualitatively based on the aggregation of the scans into a consistent final map. The method is evaluated against standard NDT and GICP for both datasets. Finally a comparison of computation time for registering a set of scans is performed.

The evaluation of the method accuracy is performed in a frame to frame arrangement and the computed transformation is compared to the relative motion of the sensor in the ground truth data. The translational and rotational errors are calculated as the Euclidean norms of the respective translational and rotational parameters. The NDT and GICP evaluations are performed with default parameters unless otherwise specified.

4.3.1 Freiburg RGBD Dataset Evaluation

The Freiburg Dataset was collected using a Kinect RGBD sensor to capture both depth and color images. For this work the depth and color images are combined to create colorized point clouds which are used with the proposed method. Ground truth information is collected using a MotionAnalysis motion capture system. The Freiburg Dataset consists of dozens of different scenes of indoor environments in a wide range of conditions. For this work the *Long Office Household Scene* scene is used as the benchmark. This scene was selected because it moves through a range of motions and consists of a typical close-quarters indoor scene. This dataset consists of 162 scans from the sensor traversing an approximately 21m path.

Figure 4.3 shows a box plot of the translational and rotational errors for each frame to frame match. In this case the grid size for NDT was set to 0.1m. This plot shows that Color Clustered NDT has very high accuracy results. The median error in translation for CCNDT is 1.7cm compared to 2.9cm for GICP and 2.7cm for NDT. The rotational errors are only marginally different with CCNDT having 0.019rad and GICP and NDT both with 0.021rad median error. The error plot also shows that the CCNDT has a tighter bounds on its error distribution compared to the other methods leading to more consistent results. Of particular note is that although GICP and NDT are able to perform admirably on this dataset due to a high degree of geometric diversity, in situations of geometric degeneracy, such as a flat wall, both GICP and NDT can fail catastrophically. Overall Color Clustered NDT in the dense indoor environment has highly accurate, robust registration results.

The aggregate map generated using Color Clustered NDT can be seen in Figure 4.4. The map is seen to be very consistent over the entire dataset.

Since the method proposed in this work is a SLAM front end all of the results represent relative error in frame-to-frame matching. This makes comparison to methods with back-end implementations, such as RGBDSLAM, challenging as they will have a clear advantage

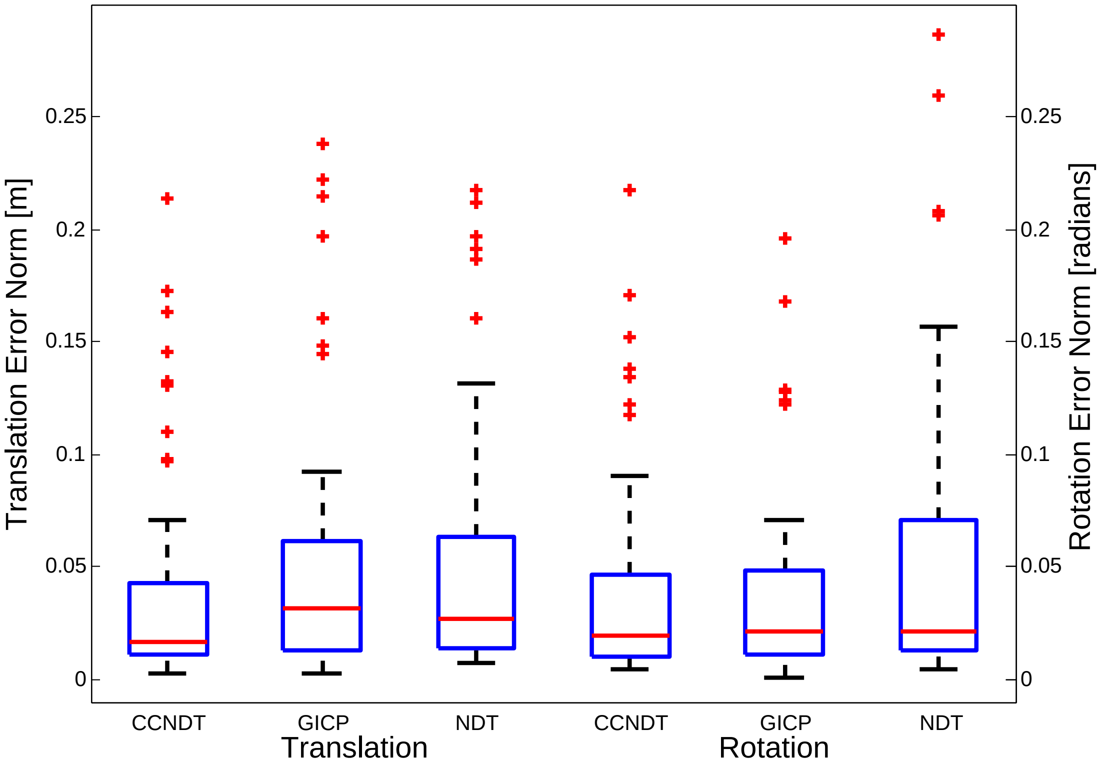


Figure 4.3: Freiburg dataset registration transformation accuracy for frame to frame matching for GICP, NDT, and CCNDT, compared to ground truth data.

Figure 4.4: Aggregate point cloud map of Freiburg dataset using Color Clustered NDT

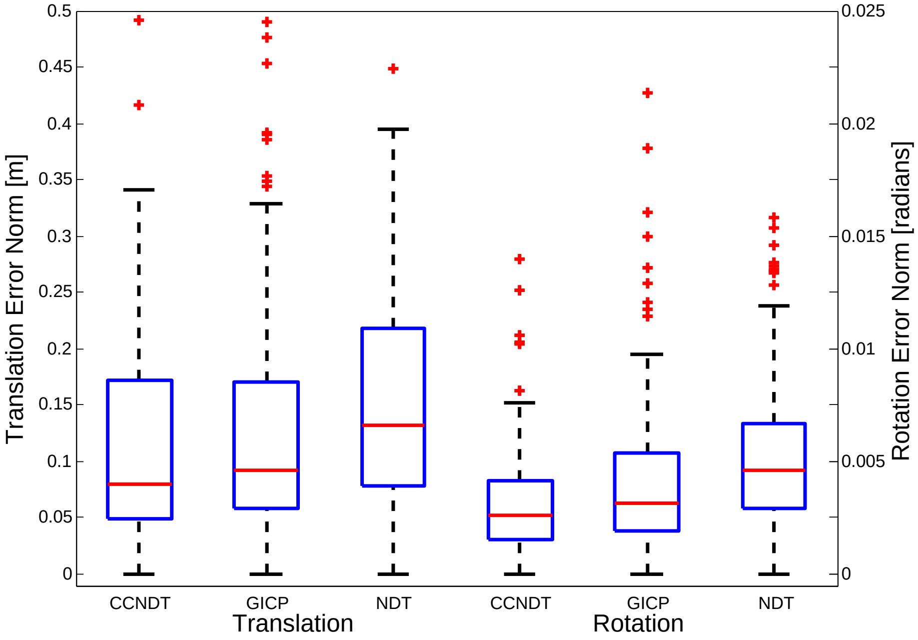


Figure 4.5: Ford dataset registration transformation accuracy for frame to frame matching for GICP, NDT, and CCNDT, compared to ground truth data.

by using a global optimization. Implementation of this method with a back-end graph optimizer and comparison to a more extensive set of RGBD algorithms is left to future work.

4.3.2 Ford Vision and Lidar Dataset Evaluation

The Ford dataset was collected using a Velodyne laser scanner combined with a Ladybug omnidirectional camera. With the accurate extrinsic calibration, the Ladybug camera is used to color the Velodyne point cloud by projecting each 3D point into the camera image. Ground truth information is collected using an Applanix Position and Orientation System. The sensors are mounted to a Ford F-150 truck and data was collected as the truck traversed a stretch of road around the Ford Research Campus. A 200m section of the dataset is used for evaluation purposes. This dataset demonstrates typical conditions of an unmanned vehicle on a roadway.

The box plot shown in Figure 4.5 shows the overall accuracy of each method in the outdoor environment. For this dataset the grid size for NDT is set to 2m. From this plot

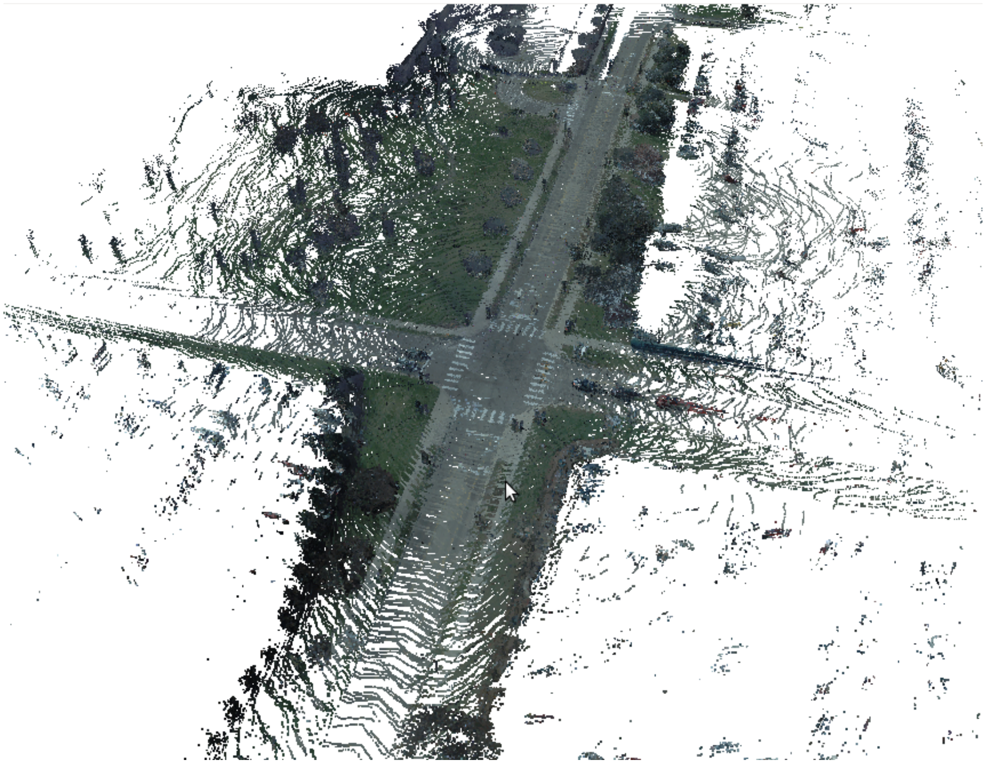


Figure 4.6: Aggregate point cloud map of Ford dataset segment using Color Clustered NDT

it can be seen that GICP and CCNDT have very similar error distributions with CCNDT having only a marginally smaller median value, while both have much lower errors than NDT. The rotational errors in these cases are disproportionately small due to the lack of rotation in the data used. Figure 4.6 displays a map of aggregate point clouds created using Color Clustered NDT on the Ford dataset. This map shows that in challenging outdoor, sparse environments Color Clustered NDT is still able to generate relatively accurate maps. This is particularly noticeable by the accurate alignment of the road markings visible in the scene.

4.3.3 Computation Time

The computation time comparison is performed using the Freiburg dataset, where each scan consists of on average 200,000 points, and using default parameters for each algo-

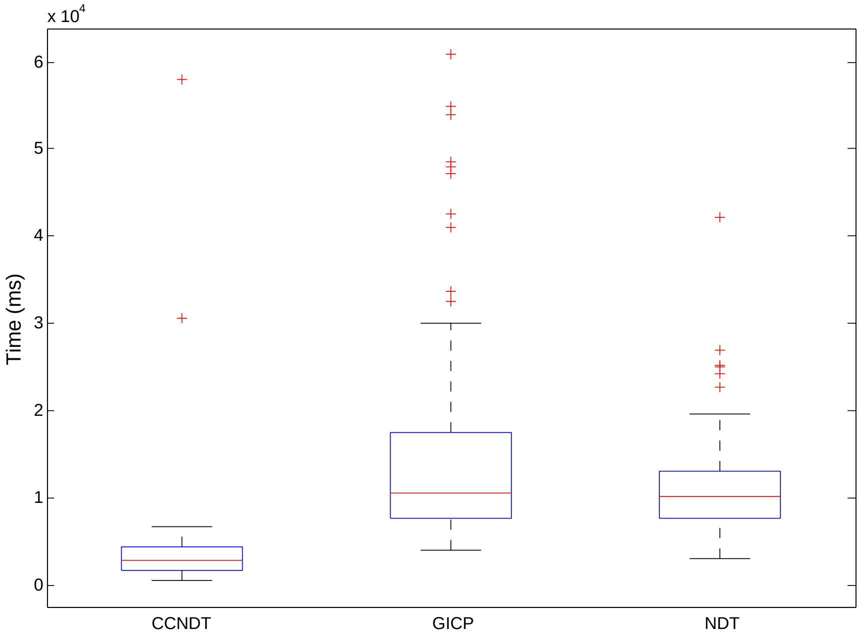


Figure 4.7: Computation time comparison between GICP, NDT, and CCNDT, for a single scan registration

rithm. The scans are not pre-filtered and are left at full point density for all algorithms. The comparison is performed on a computer with an Intel i7 Quadcore processor and 16GB of RAM with each algorithm run in a single thread with no parallelization. Overall computation time for each registration is computed and the results are plotted in Figure 4.7.

Figure 4.7 shows that Color Clustered NDT has significantly faster computation times than both standard NDT and GICP. This is expected because Color Clustered NDT uses significantly fewer clusters to perform the registrations and requires only a single optimization iteration, whereas NDT must iterate though each filled cell and GICP through every point.

The boost in computational efficiency of the Color Clustered NDT algorithm would allow it to run on a robot with only a moderately powerful computing platform. This makes it very useful for systems with limited payloads, such as aerial vehicles, or for systems with limited power available, such as rovers.

4.4 Conclusions

The Color Clustered NDT method presented incorporates the use of color based greedy clustering to generate the Gaussian distributions and uses a color weighted continuous cost function for optimization. This method is shown to have improved accuracy over existing methods in both indoor and outdoor environments and is highly robust in a wide spectrum of environments and conditions. The method proposed is highly versatile and maintains minimal computational complexity.

This method has minimal requirements on the structure and data of the input scans. CCNDT can function in a much wider range of environments and on a larger range of platforms than existing methods. The use of color allows the method to avoid many of the pitfalls faced by other registration methods, such as geometric degeneracy, but can cause its own set of issues related to the coloring itself. This method is highly sensitive to the coloring of the point cloud. Since the scan is clustered based on color the thresholds used to determine the separation of clusters can be very sensitive to illumination changes and saturation of the colors in the scene. In scenes with vibrant coloring the method performs exceptionally well and is able to cluster and register the scene without issue. In scenes which have smaller variations in color the thresholding parameters must be tuned more precisely and in scenes with zero color differences the method will not out perform NDT. In future work these parameters could ideally be learned online as the environments changes, however, this would most likely slow down the overall computation time.

Chapter 5

Multi-Channel Generalized ICP

Current state of the art scan registration algorithms which use only position information often fall victim to correspondence ambiguity and degeneracy in the optimization solutions. Other methods which use additional channels, such as color or intensity, often use only a small fraction of the available information and ignore the underlying structural information of the added channels. Some methods, such as Color Clustered NDT, rely almost completely on having sufficient secondary information in order to perform the registration. This can be an equally large problem when trying to develop an autonomous system to function in any possible environment. The proposed method reduces instances of degenerate transformation estimates and improves registration accuracy, reliability, and convergence rate.

This work proposes the Multi-Channel Generalized-ICP method. The Multi-Channel GICP method is an extension of the GICP method which incorporates additional channels of information for each point. The Multi-Channel GICP algorithm uses additional channels such as color, intensity, or any other point information, to introduce additional information to the problem. The additional channels are used to generate a spacial covariance in the plane of the surface which is used to compliment the existing plane to plane matching and allow the planes to be aligned not only in the normal direction, but also perpendicular to it. The in-plane covariance is calculated using a kernel weighted covariance based on the additional channels and is normalized by the unweighted population covariance of the points. The in-plane covariance is added to the GICP archetypical covariance in the planar directions and is then rotated into 3D space as done in GICP. Point correspondences are also changed to use a modified Color ICP method, which leverages a higher dimensional weighted kD-tree. The error function is unchanged from the original GICP, however the modified covariance will induce non-trivial error terms in the planar directions. The increased problem space not only reduces the risk of degeneracy but also improves accuracy, convergence, and robustness of the scan registration results.

The proposed method was evaluated using the Ford Campus Vision and Lidar Dataset [53], the Freiburg RGBD dataset [66], as well as Microsoft Kinect data taken on the University of Waterloo campus, in order to demonstrate a wide range of possible environmental conditions as well as a degenerate case.

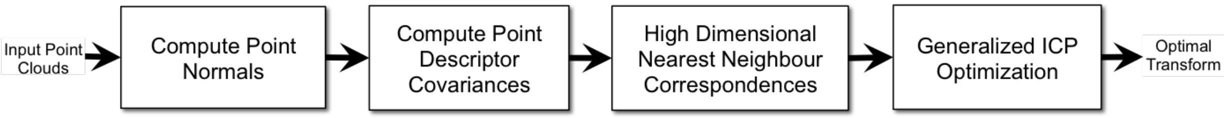


Figure 5.1: Flowchart showing the stages of the Multi-Channel Generalized Iterative Closest Point algorithm.

5.1 Covariance Calculation

Multi-Channel GICP assumes, as GICP did previously, that the environment is locally planar and that the 3D points only contain useful information in the direction normal to the surface. However, since the points have at least one additional channel of information the additional channel(s) can be used to define the covariance of a point along the surface plane as well. The added channels will have no effect normal to the plane as the sample must lie on the surface and therefore will complement the positional information well.

First, let all points, $p_i = \{p_i^p, p_i^d\}^T$, have both positional information, $p_i^p \in \mathbb{R}^3$, and n descriptor channels, $p_i^d \in \mathbb{R}^n$, which can include, for example, intensity or RGB colour information. Then for each point in point clouds A and B, the model covariance sets, C^A and C^B, are calculated using both position and descriptor information.

Let $q \in A \cup B$ be the current query point for which the model covariance is to be calculated. The local covariance, C_i^L, of the query point position is calculated using the k nearest neighbour points to q^p using [44]. Let the nearest neighbours be defined as the set of points $L = \{l_j\}$ for $j \in \{1, ..., k\}$, such that $\|l_j^p - q^p\| \leq \|r - q^p\|$, for all $r \in Q \cap \bar{L}$, where Q is the point cloud associated with the current query point, q. Singular value decomposition (SVD) is performed to extract the principal components. The normal of the surface is found as the component with the smallest singular value in S. The neighbourhood points are then projected onto the plane perpendicular to the normal and reduced to $\mathbb{R}^2$. Let $z_j^p \in \mathbb{R}^2$ be the projected point and $Z = \{z_j^p, z_j^d\}^T$ be the new set of points in $\mathbb{R}^{2+n}$. The projection is then given as

$$\begin{aligned} z_j^p &= \begin{bmatrix} {U_1}^T \\ {U_2}^T \end{bmatrix} l_j^p \\ z_j^d &= l_j^d \end{aligned} \tag{5.1}$$

where U_1 and U_2 are the first and second columns of the SVD matrix U. Note that after the transform, the new population covariance, $\Sigma_w \in \mathbb{R}^{2\times 2}$, of the points, Z^p, is the diagonal matrix of the largest two singular values of S.

Now that the 3D points have been projected onto the local surface approximation, the reduction in uncertainty due to the incorporation of descriptor information in the plane can be calculated. To this end a descriptor kernel weighted covariance is calculated using weightings based on a probabilistic model similar to that used in [28]. The descriptor kernel calculates the probability that an arbitrary point corresponds to the query point in descriptor space. The kernel can be defined as a Gaussian distribution, $\mathcal{N}(q^d, \Lambda)$, centered at the query point descriptor, q^d, and with $\Lambda \in \mathbb{R}^{n\times n}$ being the measurement covariance of the descriptor sensor. The kernel weights are then calculated for each point in Z as:

$$\lambda_j = \exp(-\frac{1}{2}(z_j^t - q^t)^T \Lambda^{-1}(z_j^t - q^t)) \tag{5.2}$$

Using the kernel weights the descriptor kernel weighted covariance and mean, Σ_t and μ^p, can be calculated as

$$\mu^p = \frac{1}{\sum\limits_j \lambda} \sum_j \lambda_j z_j^p \tag{5.3}$$

$$\Sigma_t = \frac{1}{\sum\limits_j \lambda} \sum_j \lambda_j (z_j^p - \mu^p)(z_j^p - \mu^p)^T \tag{5.4}$$

This gives the spacial distribution of points based on their similarity to the query point. This distribution models the uncertainty of the descriptor information along the wall locally, however it can be biased if the original sample population was itself already biased. Figure 5.2 shows an example of a population and descriptor covariance with a biased initial population. To compensate for this potential bias the distribution is normalized by the population covariance such that

$$\Omega = \Sigma_w^{-\frac{1}{2}} \Sigma_t \Sigma_w^{-\frac{1}{2}} \tag{5.5}$$

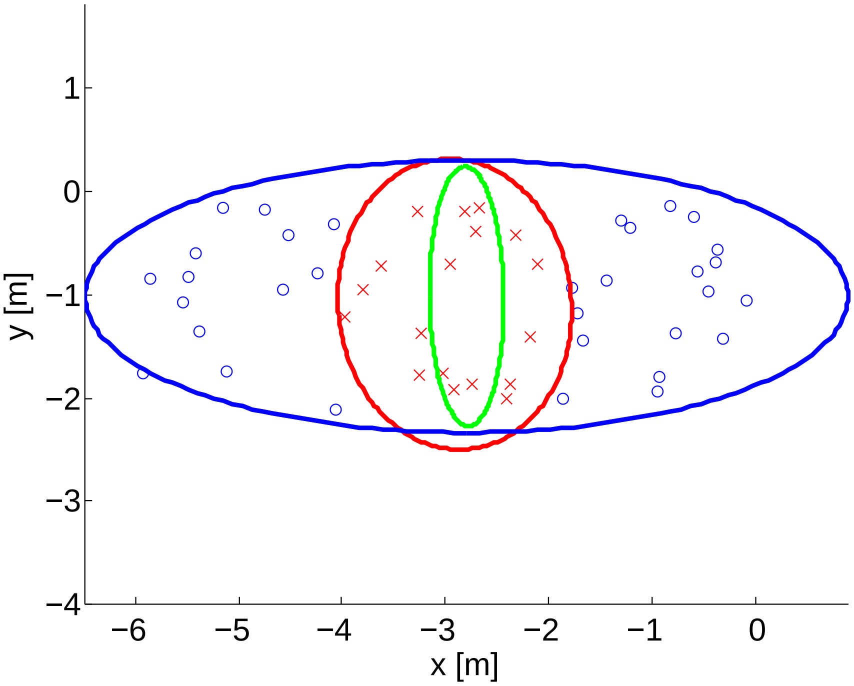
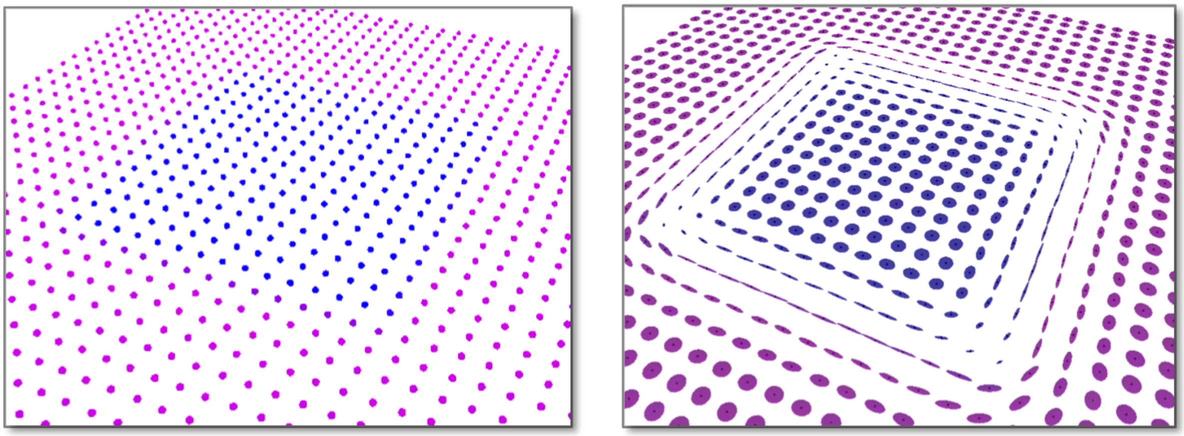


Figure 5.2: An example of the population (blue), and descriptor (red), covariances given a biased initial population and the resulting normalized correlation (green).

The normalized descriptor covariance, $\Omega \in \mathbb{R}^{2\times 2}$, shows the correlation of the descriptor weighted data compared to that of the population. The normalization in this case is a whitening transform on the population covariance. The population covariance would be transformed into an identity matrix in the normalized space and the descriptor covariance would be transformed relative to the population to give an unbiased descriptor covariance as if the population were identity. In practice a value less than one indicates that the descriptor data increased the data certainty in that direction, while a value less than one indicates an increase in uncertainty. Directions which have a low normalized covariance are more likely to capture correct correspondences in the descriptor space in that direction. Cases which have normalized covariances equal to or great than one indicate areas of low descriptor correspondence certainty, such as a wall of a single continuous color.

To use this information in the GICP framework, Ω is used along the planar directions. Therefore, the resulting covariance used in the MC-GICP algorithms is

$$C^D = \begin{bmatrix} \Omega & 0 \\ 0 & \epsilon \end{bmatrix}$$



(a) Example point cloud

(b) MCGICP Covariances

Figure 5.3: Example point cloud (left) with each point covariance calculated using the MCGICP method (right)

Figure 5.3 shows an example color point cloud surface with a distinctive color feature in the center and the corresponding covariances calculated by the MCGICP algorithm. It can be seen in the figure that points along the edges of the color boundary have covariances that align with the color edge. This allows the method to align the surface, even without geometric features, by aligning the edge features in the color channels.

The error function and minimization of the Multi-Channel GICP method remain unchanged from the original GICP, presented in Equation (2.5), which means that all current methods for solving the GICP optimization are still valid and no changes are necessary to the optimization.

5.2 Correspondence Calculation

In addition to the covariance changes, the calculation of corresponding points is also changed to reflect the higher dimensionality of the information. An $n + 3$ dimensional weighted kD tree is used to incorporate all of the information into the search as first shown by Johnson and Kang [30]. A weighting vector, $\alpha = \{\alpha_1, ..., \alpha_{n+3}\}$ where $\alpha_1 = \alpha_2 = \alpha_3 = 1$ are the weights of the position data, is used to weight the descriptor information relative to the positional information.

The correspondence information does not need to come directly from the raw sensor information. Processed information can be used to improve the correspondences. By using the eigenvalues of the covariance matrix calculated at each point as well as spacial and descriptor information, better correspondence matching has been shown to be possible. Using the eigenvalues of the covariance matrix, weights points with similarly shaped covariances to be corresponded. This results in points along the edges of both geometric and descriptor structures to be corresponded and results in much more robust overall registration results. The use of other processed information to improve the correspondences further is possible, however this is left as a point of future work.

5.3 Extrema Cases

Unlike many other methods the MCGICP method is capable of gracefully handling the extreme cases of input scans. This includes the two practical instances which are a scan with a consistent descriptor value for every point and a scan with unique descriptor values for every point. Most other methods fail catastrophically in these cases where as MCGICP falls back to a logical state that is still capable of performing the registration with the information provided.

In the first case, where all points have identical descriptor information, the MCGICP method will mathematically become equivalent to the base GICP algorithm. This happens because the normalized descriptor covariance will become the identity when there is no change in the weighted descriptor covariance, Σ_t, compared to the population covariance, Σ_w. This would mean that,

$$\Sigma_w = \Sigma_t \tag{5.6}$$

$$\Omega = \Sigma_w^{-\frac{1}{2}} \Sigma_t \Sigma_w^{-\frac{1}{2}} = I \tag{5.7}$$

$$C^D = C^G \tag{5.8}$$

Therefore the GICP archetype covariance, C^G, is assigned at every point. Since each point has the same descriptor and covariance, the correspondence search will only be dependent on the geometric distances. Thus the algorithm will be equivalent to the GICP algorithm.

In the second case, where all points have a unique descriptor, the MCGICP method will become equivalent to the color ICP algorithm. In this case, because all of the descriptors are unique, the weighted descriptor covariance will be very small. Thus it follows that,

$$\Omega = \epsilon I \tag{5.9}$$

$$C^D = \epsilon I \tag{5.10}$$

Since C^D in this case is simply a scaled identity matrix and all points will have approximately the same matrix, each point will have the same weighting relative to others as well as the same weighting in all directions. The correspondence calculation will however still take into account the descriptors with the result being points corresponded by descriptor and geometry but with equal weightings. This is equivalent to how the color ICP algorithm would perform and in this case is the ideal case for this type of environment.

In both extreme cases the MCGICP algorithm seamlessly handles using the information available in an intelligent manner, allowing the method to be robust to the widest range of possible environments.

5.4 Implementation

The proposed method is a generalized framework which can be used with a variety of different sensor combinations. The number of additional channels which can be added to the points is not limited by the algorithm but in practice is only limited by diminishing returns on the usefulness of the information. Three possible sensor configurations are discussed below. First, a system which uses only a LIDAR sensor with intensity information, second, a typical colourized point cloud generated from a camera and range sensor combination, and finally a configuration incorporating a LIDAR sensor with intensity information with a camera setup to provide four additional information channels.

5.4.1 Laser Intensity Descriptor

A single LIDAR, such as the Velodyne HDL-64E, can add an additional channel in the form of laser intensity, which provides useful information to the registration. In the single channel case, the descriptor covariance reduces to a single value, and is found by computing the variance, $\Lambda \in R$, of the laser scan intensity values. Finally the weighting, α_4, of the intensity channel in the nearest neighbour search to scale the influence the intensity channel will have on finding nearest neighbours.

5.4.2 Color Descriptor

The combination of a camera and a range sensor is a common setup on many robotic systems. Color provides three channels to incorporate into the model. In this case $\Lambda \in \mathbb{R}^{3\times3}$ is a covariance matrix of three variables. However, it can be assumed that the color channels are independent and therefore Λ will be a diagonal matrix consisting only of the intra-channel variances. The exact values of Λ will depend not only on the sensor used but also on the color space which is chosen.

In this work the RGB space is used. In the RGB space, the color variances and weightings can be set equal to each other to represent equal uncertainty in each of the colors. The option of alternative color spaces is also possible as each space provides different benefits and considerations. For an HSV or YUV space, the variance of the value and illumination channels can be larger and the nearest neighbour weighting smaller to distinguish a higher uncertainty in illumination which is common in real world scenes. The choice of color space does not have a direct impact on the algorithm but does change the desired values of Λ and $\{\alpha_4, \alpha_5, \alpha_6\}$.

5.4.3 Combined Color and Intensity

The combination of both color and laser intensity information presents and interesting configuration which is not typically leveraged in current algorithms. Although the channels of the combined descriptor could be considered to be independent it has been shown in [54] that laser intensity and color intensity are in fact positively correlated. This can be incorporated into the algorithm by setting the inter-channel covariance terms of Λ to non zero values. The covariance matrix can be determined experimentally using a set of known training data. The weighting values for the nearest neighbour search are typically inversely proportional to the variance of that particular channel and are therefore dependent on the specific sensors being used.

5.5 Experimental Results

The Multi-Channel GICP method is evaluated using three sets of data. The first set is the Ford Campus Vision and Lidar Dataset [53]. The Ford dataset contains LIDAR and omnidirectional image data as well as ground truth and is used to evaluate the quantitative accuracy of the scan registration results in outdoor evnironments. The method is evaluated using the laser intensity (I), color (C) and combined (C+I) descriptors as described in Section 5.4. The second dataset used for evaluation is the Freiburg RGBD dataset. The Freiburg dataset is generated in staged indoor environments and is collected using an RGBD camera. The Freiburg data is used to evaluate the accuracy and robustness of the method in indoor environments. The final set of data was obtained using a Microsoft Kinect sensor on the University of Waterloo campus. This dataset is used to evaluate the method on data with limited geometric structure. The method is evaluated qualitatively based on the reconstruction of a flat textured surface. Finally the convergence rate of each algorithm is compared. In all cases the method is compared to both the original GICP and Color ICP algorithms as implemented in the Point Cloud Library (PCL) [6] as well as the CCNDT algorithm. In all evaluations the RGB color space is used.

5.5.1 Ford Dataset Absolute Error

The Ford Campus Vision and Lidar Dataset was generated using a Ford F-250 pickup truck equipped with a Velodyne HDL-64E laser scanner, a Point Grey Ladybug3 omnidirectional camera, and a Applanix POS-LV 420 INS with Trimble GPS used for ground truth data. A series of 200 frames from a challenging subset of the dataset were used to perform pairwise registration on each consecutive pair of scans. The resulting transforms were then compared to the ground truth data to calculate the mean and standard deviation of the translational and rotational errors. The results are summarized in Table 5.1.

Table 5.1: Summary of translation errors of the MCGICP on the Ford dataset

	Color ICP	GICP	CCNDT	MC-GICP Intensity	MC-GICP Color	MC-GICP Combined
mean error [m]	0.3532	0.2344	0.2090	0.2370	0.2045	**0.1606**
std. dev.	0.2935	0.2965	0.1952	0.2419	0.1943	**0.1492**

The error distributions from Figure 5.4 as well as the error summary provided in Table 5.1 show that Multi-Channel GICP using either color or combined descriptors has increased accuracy and reduced uncertainty over that of CCNDT, GICP, and Color ICP. Of the Multi-Channel GICP variants, the combined descriptor produced the best results followed by the color descriptor while intensity alone produce poor results. This is expected as the combined descriptor provides the most robust information while the intensity alone has minimal distinctive variation in value. The combined descriptor increases the information in each point while maintaining consistency. This results in increased correlation and correspondence certainty. This shows that the Multi-Channel GICP method is dependent on the use of an accurate and discriminative descriptor space.

The aggregated maps of a challenging section of the Ford dataset generated by Color ICP, GICP and Multi-Channel GICP using the combined descriptor are shown in Figure 5.5. In the aggregate maps, it can be seen that Multi-Channel GICP creates more accurate results. This is evident by the blurring which can be seen in the Color ICP and GICP maps but is reduced in the Multi-Channel GICP results.

An aggregate map of a large section of the Ford dataset generated using scan-to-map matching with the MCGICP algorithm is shown in Figure 5.6. Only a subsection of the map, aggregating every 20th scan, is shown for visual clarity. The map shows a highly consistent path, with only minimal drift throughout the loop. The use of scan-to-map, where scans are matched to an aggregation of previous scans, improves the global consistency of the results but in order to generate an optimally globally consistent map a beck-end optimization would be used.

5.5.2 Freiburg RGBD Indoor Dataset

The Freiburg RGBD dataset used for these experiments is the *Long Office Household Scene*. In this case, since the dataset only contains points with color information, only color is used as the descriptor channel. This dataset consists of 162 scans with ground truth information from a motion capture system. Scan-to-scan matching is performed and compared to the ground truth information to evaluate accuracy.

Figure 5.7 shows the accuracy comparison of MCGICP with CCNDT and existing methods. The graph shows that both MCGICP and CCNDT have improved accuracy over the existing methods, however in this case MCGICP and CCNDT have approximately equivalent results. This is because this scene is ideally suited for CCNDT as it has high variation in color. MCGICP is also able to use this color information effectively but not to the extent that it improves in accuracy significantly over CCNDT. It should be noted however that MCGICP does show fewer outliers in the registration results than CCNDT demonstrating that MCGICP is more robust even in this case.

5.5.3 Kinect Sparse Geometry Data

The Kinect dataset was obtained using the Microsoft Kinect RGB-D sensor mounted to a mobile robotics test platform.

The sequence is of a flat wall which is covered in posters. This sequence contains no distinct geometric surfaces other than the flat wall and therefore produces degenerate solutions in the x and y directions when only geometry is considered. The sequence consists of 200 frames traversing from right to left over a $10m^2$ area. The aggregated results of the pairwise registration of the three methods are presented in Figure 5.8.

Figure 5.8 demonstrates the ability of the Multi-Channel GICP algorithm to use the additional channels to compensate for the lack of geometric information. CCNDT performs comparably to MCGICP in this case and is not shown. Color ICP is also able to partially compensate but has much lower accuracy observed by the increased blurring of the posters. Color ICP has trouble correctly aligning the scans due to noise in the images causing incorrect correspondences and skewing the results. The GICP results fail catastrophically in this case due to the lack of geometric information along the wall and therefore all the scans are incorrectly aligned.

5.5.4 Convergence Rate

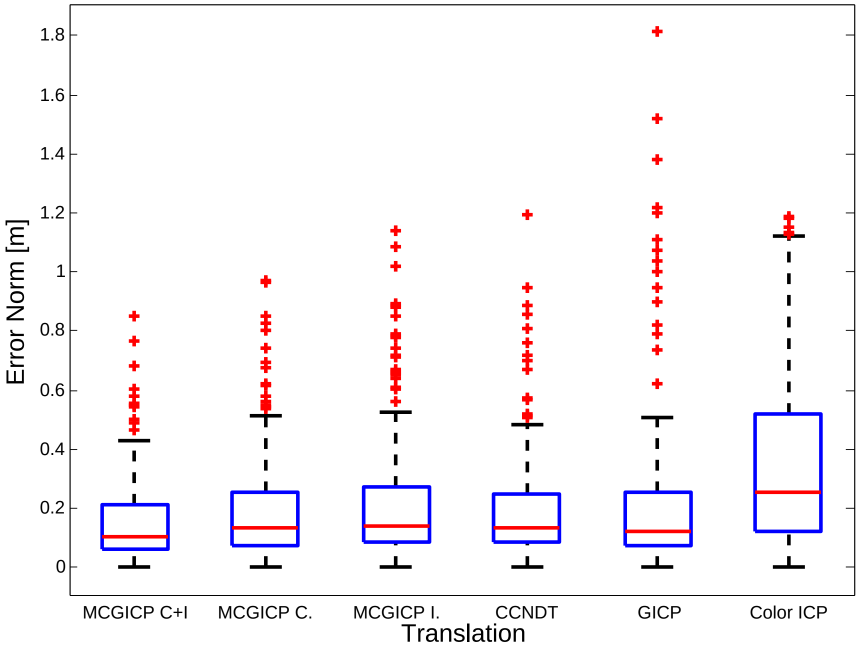
The final evaluation compares the convergence rates of the three algorithms. Given an example frame from the Ford Data Set, the error residual is plotted versus the iteration in Figure 5.9. As all three algorithms use computationally similar cost functions, iterations are proportional to convergence time.

Figure 5.9 clearly shows that MCGICP converges significantly faster than the original GICP algorithm. Color ICP has been shown to converge faster than standard ICP due to the fact that is acquires the correct correspondences faster using the higher dimensional search space. However, GICP, is shown to converge faster than Color ICP. This is because GICP does not rely as heavily on correct point correspondences to converge and only needs corresponding points to lie on the same surface. MC-GICP combines the beneficial properties of both to acquire the correct surface correspondences and converges most rapidly.

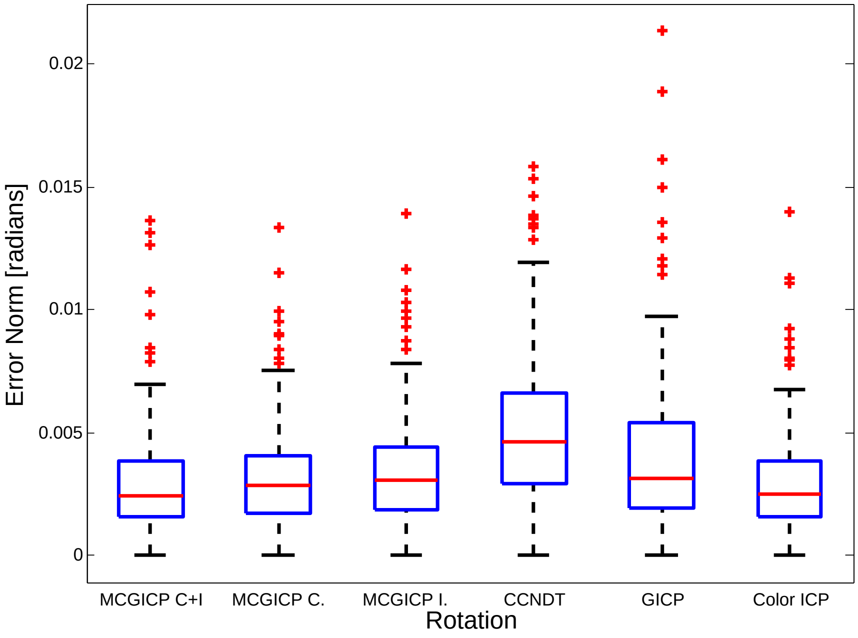
5.6 Conclusions

This work presents the Multi-Channel Generalized-ICP method for robust scan matching. The proposed method incorporates the additional sensor channels directly into the GICP formulation to provide additional information in the plane parallel to the local surface. The GICP algorithm relies solely on surface normal information at each point, and requires surface normals from the point set to span all of $\mathbb{R}^3$ to properly determine the transformation and avoid degeneracy of the solution. However, many real world environments do not provide sufficient information using surface normals alone, such as hallways, or flat open spaces. The Multi-Channel GICP method modifies the model covariance planar to the surface normal and calculates nearest neighbour correspondences in a higher dimensional space, thereby exploiting the additional information available in the point cloud from secondary sensor channels to avoid this shortcoming.

The proposed method demonstrates improved registration accuracy and convergence rate over existing methods as well as robustness to degenerate geometric cases. The MCG-ICP method is capable of performing effectively in a wider range of possible environments then any other scan registration technique and is applicable to a vast range of possible sensor configurations. Although MCGICP may be slower computationally then some other methods, such as Color Clustered NDT, its reliability and robustness can easily make up for any such short coming when implemented in a full SLAM system in most cases.

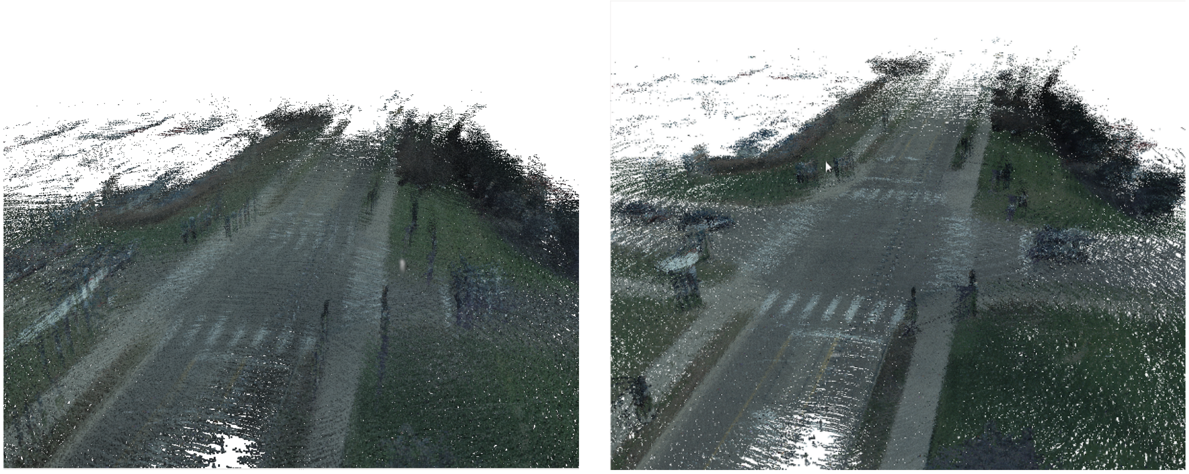


(a) Translational Error

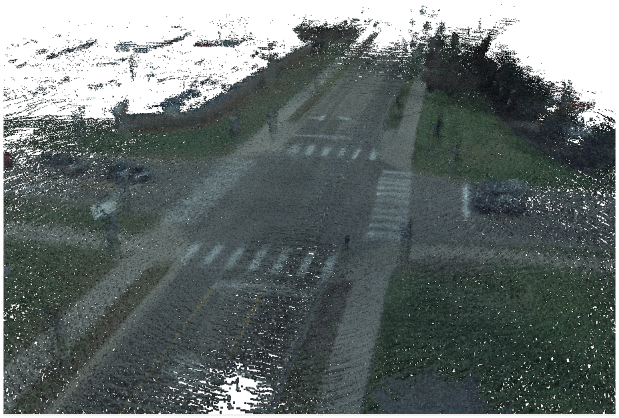


(b) Rotational Error

Figure 5.4: Comparison of accuracy results for MCGICP against existing algorithms performed on the Ford dataset.Translational (top) and rotational (bottom) errors are shown.



(a) Color ICP (b) GICP



(c) MC-GICP Combined

Figure 5.5: Aggregated point cloud maps generated from a subsection the Ford Vision and Lidar dataset.

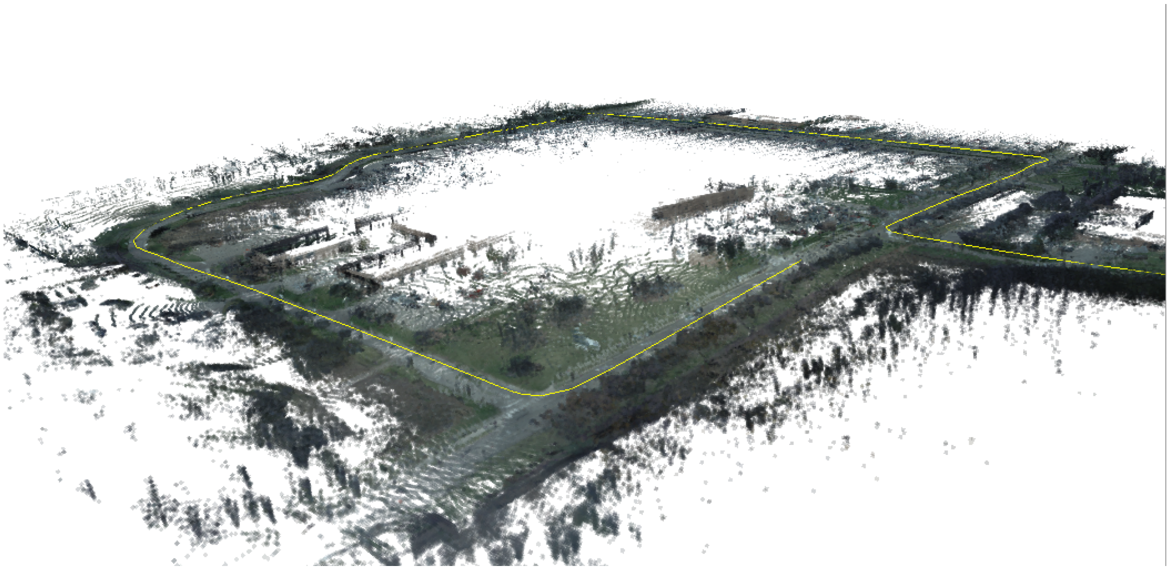
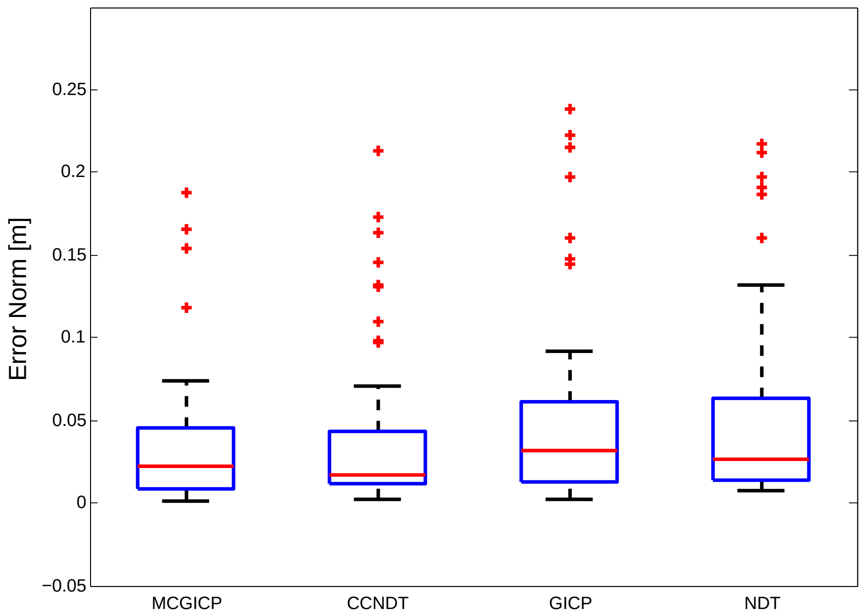
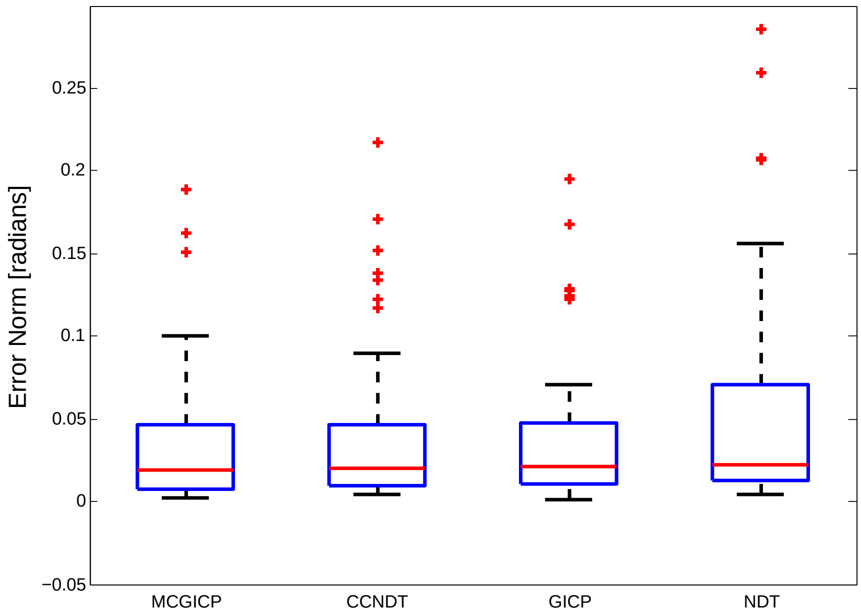


Figure 5.6: Aggregate map of a subsection of the Ford dataset generated using MCGICP with scan-to-map matching. The yellow line indicated the calculated path of the vehicle.



(a) Translational Error



(b) Rotational Error

Figure 5.7: Comparison of accuracy results for MCGICP against CCNDT and existing algorithms performed on the Freiburg dataset.

(a) GICP

(b) Color ICP

(c) MC-GICP

Figure 5.8: Aggregated point cloud maps generated from the geometrically degenerate poster wall dataset using the Kinect sensor.

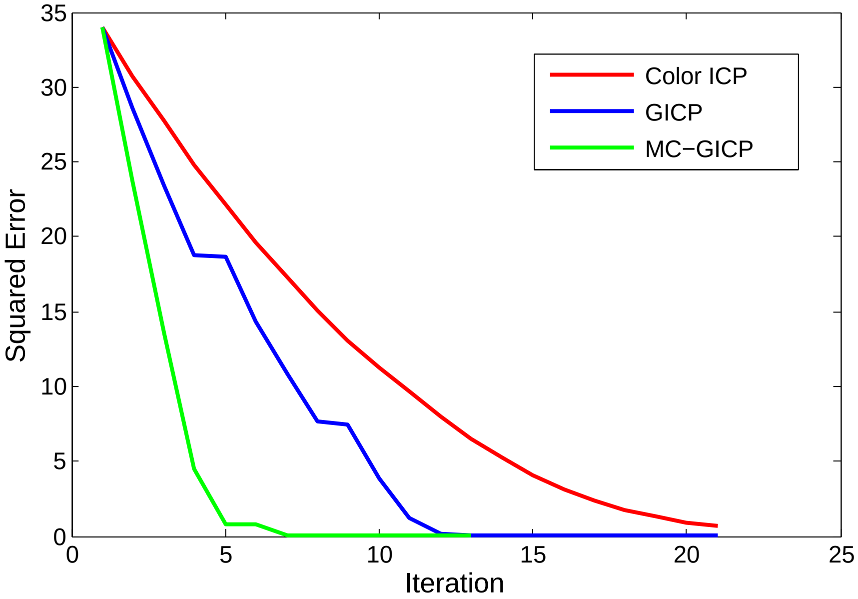


Figure 5.9: Comparison of convergence of the Color ICP (red), GICP(blue) and MC-GICP (green) algorithms.

Chapter 6

Full System Integration

In the previous Chapters, methods for effective scan registration were discussed. However, scan registration is only a piece of a complete SLAM solution for a given robotics application. In order to effectively be able to navigate, the SLAM system needs both a front and back end solution as well as a method for storing and processing the overall map in such a way that the other components, such as path planning, can effectively access the required information. In this chapter the components necessary for completing the overall system are discussed and results of the overall SLAM implementation are presented. For this work the GSICP algorithm is used as the scan registration method to perform 2D SLAM, however, either of the 3D methods would be equally applicable with only minor changes to the setup.

The SLAM system presented here consists of four main nodes, as shown in Figure 6.1. The local mapping node is responsible for transforming, segmenting and classifying the point cloud data from the Velodyne LIDAR. The EKF maintains an accurate estimate of the current robot pose, fusing information from all sources. The global mapping node aggregates keyframes into a global map based on the discrepancy score and generates the pose graph through scan registration. Finally, the graph optimization node is used to update the global pose graph and regenerate a drivability map, which is then passed to the global planner in order to perform planning operations.

In the global map, each grid cell stores an average position of the planar projection of non-ground points located within that bin in the global frame. In implementation, this point average is selected to be the mean of the point components. Using the mean points is a more robust approach than a naive occupancy grid, since the mean of the points provides a better measure of location of the true model points within the cell. Accurate modelling

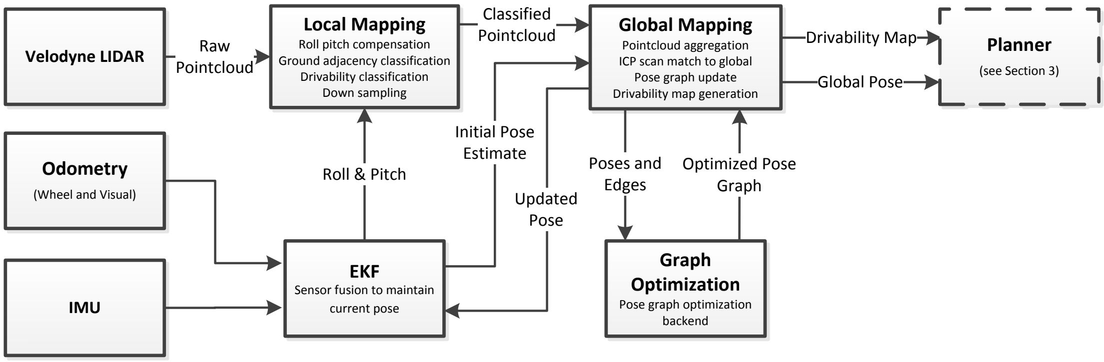


Figure 6.1: Block diagram of the SLAM mapping algorithm build for GSICP. The local mapping node uses the point clouds from the Velodyne LIDAR and the attitude estimate from the EKF to compensate the scan for the roll and pitch of the vehicle. The global mapping node then aggregates the information from the local maps, which is used to generate a drivability map for the path planner. The global mapping node also generates a pose graph which is updated by the graph optimization node.

of the points is especially important when performing registration, as the registration algorithm typically attempts to minimize the Euclidean distance between corresponding points. The height, or z-components for the points located within each bin are used to generate a Gaussian distribution which models the height of the obstacles within the bin. This allows the global map to maintain a sense of the obstacle elevation within each bin, which is used to perform outlier rejection for the algorithm.

In order to integrate new information into the global map, keyframes are employed [25, 14, 2], which allow for the generation of a pose graph that is used to improve the global consistency of the map. A confidence score for the map is computed based on the amount of discrepancy between the global map and the current obstacle point cloud, which allows for the integration of new information into the map only when required, mitigating drift accumulation in the global map. Vertices for the pose graph are added when the map uncertainty score grows above a user defined threshold and new information is integrated into the global map. Additional edges for the pose graph are constructed by performing registration between the newly added vertex and its k nearest neighbour vertices. Once a loop closure is detected, the network graph optimizer g2o is used to optimize the pose graph represented by the stored keyframes [38]. Finally, the global map is regenerated using the updated pose graph.

6.1 Global map integration

In order to define when to include a new key frame in the global map, a map confidence score is proposed. Denote the point cloud $\bar{P}$ as the 3D point cloud which has been transformed into the global frame, using the transform parameters T obtained through scan registration. Denote the Gaussian height distribution of a cell φ_i in the global map as $\mathcal{N}_i^{\Theta}(\mu_i^z, (\sigma_i^z)^2)$, and the height distribution for the set of transformed scene points, γ_i, contained within cell φ_i, as $\mathcal{N}_i^{\gamma}(\bar{\mu}_i^z, (\bar{\sigma}_i^z)^2)$. Using the height distribution contained within a cell and the height distribution generated using the transformed points, a score can be defined which quantifies the divergence between the two Gaussian distributions,

$$\psi_i = \exp\left(-\frac{1}{2}\frac{(\mu_i^z - \bar{\mu}_i^z)^2}{(\sigma_i^z)^2 + (\bar{\sigma}_i^z)^2}\right) \tag{6.1}$$

Denote the points of $\bar{P}$ that are located with a cell of the global map, φ_i, as γ_i. For each point in the set γ_i, a residual distance to the mean point value associated with the global map cell φ_i can be calculated. Denote the total residual score for the cell ω_i as

$$\omega_i = \begin{cases} 0, & \text{if } \mu_i^z = 0, \\ \sum\limits_{p \in \gamma_i} \|p^{xy} - \mu_i^{xy}\|, & \text{otherwise.} \end{cases} \tag{6.2}$$

where $p^{xy} \in \mathbb{R}^2$ denotes a vector consisting of the x and y components of the point p. Note that a score of zero is given if the cell is a ground cell, as this implies there are no obstacle points to obtain a registration residual from. Finally, denote N_o as the total number of non-ground cells in the global map which contain points from the transformed point cloud $\bar{P}$ and denote N_n as the total number of ground cells in the global map which contain points from the transformed point cloud. Note that N_n is the number of cells where there is a discrepancy in the global map, and the transformed point cloud suggests that the cells should be updated as obstacles. The larger N_n becomes, the less confidence there is in the global map. In order to integrate new information into the map, two conditions must exist. The *map uncertainty condition* is given as

$$\frac{N_n}{N_n + N_o} > \epsilon_n \tag{6.3}$$

where ϵ_n is a user defined parameter which controls the amount of discrepancy required before considering integrating new information into the map. In order to ensure that the scan registration converged to a correct solution, a score is computed based on the registration residuals and difference in height distributions for each cell. The *registration uncertainty condition* is given as

$$\sum_{i=0}^{N_o} \left(\frac{\omega_i}{\varsigma_d N_o} + \left(1 - \frac{\psi_i}{N_o} \right) \right) < \epsilon_o \tag{6.4}$$

where ς_d is the diagonal distance for a cell in the global map, and ϵ_o is a user defined parameter which controls the required quality of registration required in order to integrate information into the global map. If the map uncertainty and registration uncertainty conditions are satisfied, the information from the transformed point cloud $\bar{P}$ is integrated into the global map.

Intuitively, the map uncertainty condition defines a ratio of the number of cells in the global map where new information is suggested to be added by $\bar{P}$, to the total of cells which are occupied by $\bar{P}$. In the case where sections of the environment are revisited, N_n will tend towards zero, as the map within the sensor range has been previously explored. The registration uncertainty condition is computed using the sum of two scores based on the registration residual normalized using the diagonal cell dimension, and the normalized discrepancy in height between the cells in the global map and the transformed point cloud $\bar{P}$. The map and registration uncertainty conditions are suitable for SLAM because they provide a robust method to control key frame insertion based on the quality of the scan registration and are computationally inexpensive to compute.

The integration process simply consists of updating the means and standard deviations associated with each cell in the global map, φ_i, using the points γ_i. In order to further improve robustness, cells are not updated until a required log-odds ratio of occupancy is achieved. The occupancy update is the standard Bayesian occupancy grid map update and the method used for this work is described in detail in [67].

6.2 Graph SLAM and global consistency

In order to maintain a globally consistent map, a graph SLAM framework is used. Denote a vertex in the pose graph as $v \in \mathbb{SE}(d)$, where $d = 2$ for 2D registration or $d = 3$ for 3D, and the set of all vertices, V_p, where each vertex defines a vehicle pose at a particular time. Denote an edge between two vertices, v_i and v_j, as $e_{ij} \in V_p \times V_p$, and the full pose graph as $\mathcal{G}_p(V_p, E_p)$. For the graph SLAM problem, each edge represents a constraint between its vertices. In this method, each constraint on edge e_{ij} is imposed by performing a scan registration between the point clouds associated with the vehicle poses at vertex

v_i and vertex v_j. When edges are added such that their configuration results in an over-constrained pose graph, a graph relaxation optimization is performed. Intuitively, the optimization can be viewed as finding the values for the vertices such that the sum of errors imposed by each edge constraint is minimized.

The graph is initialized with a fixed vertex at the vehicle's start position, which establishes the SLAM co-ordinate frame. A new vertex is added to the graph when new information is added into the global map, as described in Section 6.1. In order to ensure the vehicle does not travel too far without adding a vertex, a vertex is forced to be added if it has travelled a distance of δ_{max} since the last added vertex. The forcing of a vertex is done to ensure that there is adequate overlap in the point clouds to perform a registration when constructing an edge. In order to keep the number of vertices in the graph manageable, a vertex is not added if there is another vertex that is within a distance of δ_{min}. Each time a vertex is added to the graph, edges are added by performing scan registration to the vertices which are the K-nearest neighbours to the current vehicle position. Once the edge constraints are added, the full graph optimization is performed using the g2o back-end. Finally, using the optimized pose graph, the global map is regenerated using the point clouds associated with each vertex in the updated graph.

6.3 Drivability map extraction

To obtain a drivability map for the path planner, an additional simplified global map is maintained. Every time new information is added into the global map, the drivability map is also updated using only the points classified as non-drivable, such as described in Section 3.1. The map update takes place when new information is integrated into the map due to map uncertainty and also when the map is regenerated after a pose graph optimization. An example of the drivability map is presented in Figure 6.2. The black coloured cells represent the non drivable obstacle information. The light grey cells represent overhanging features such as trees and archways which the vehicle can safely traverse under. An aerial photo of the mapped area is also provided for reference.

6.4 Global Consistency Results

Two experiments are performed to validate the SLAM system. The first experiment uses the Ford Vision and LIDAR dataset while the second uses data collected at Waterloo Park using our custom robotics platform. The experiment is performed in real-time using


(a) Aerial photo of test area

(b) Driveabilty map

Figure 6.2: An example of a global drivability map. (a) Aerial photo of the mapped area (b) Global map with non drivable cells coloured in black, and traversable obstacles which the vehicle can pass under coloured in grey.

C++/ROS on-board the vehicle. For the experiment, the global map cell size is selected to be 0.5 m, the map uncertainty score threshold, is set to $\epsilon_n = 0.3$ and the registration uncertainty is selected as $\epsilon_o = 0.1$. The distance required to force the addition of a key frame is set to $\delta_{max} = 20\,\mathrm{m}$ and the minimum distance between key frames is set to $\delta_{min} = 10\,\mathrm{m}$. The number of nearest neighbour edges to connect upon a vertex insertion is set to three.

The first experiment uses the SLAM solution on the Ford Vision and LIDAR dataset to demonstrate the effects of incorporating the back-end optimization. By incorporating the back-end graph optimization the drift caused by performing scan-to-scan registration can be mitigated to obtain a globally consistent map. Figure 6.3 shows the comparison between the final map of the Ford dataset generated with and without using the back-end optimization. It can be seen that the back end optimization is able to create a globally consistent map and correct for the build up of minor errors from the scan registration algorithm. This demonstrates that minor errors in the accuracy of the scan registration solution can be acceptable. However, if the scan registrations were to fail catastrophically, such as in a degenerate case, the back-end would not be able to find the loop closure and the map would not be able to be corrected. It is therefore more important for scan registration methods to be robust and reliable than accurate, as small errors can be corrected but large errors may not.

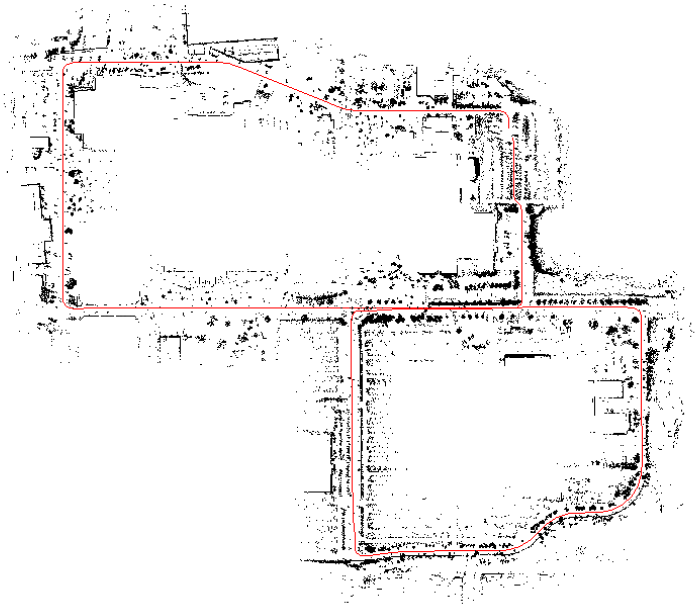
The second experiment to validate the SLAM approach is carried out in Waterloo Park, which is adjacent to the University of Waterloo campus. The test location is a field which measures approximately 60m by 60m, is mainly sparse, but contains trees and shrubs along the outer perimeter. The experiment consists of a test case where the vehicle drives a sweep path, which is typical of the operating conditions, as generally, sweep paths are required to ensure coverage. The vehicle maintains a constant velocity of 1 m/s throughout the duration of the experiment. The accuracy of the vehicle path is validated using Trimble S3 robotic total station, which is capable of collecting measurements to a tracking prism at a rate of approximately 1 Hz with millimetre level accuracy.

Figure 6.4 presents the map results of the vehicle traversing the sweep path in the open field and returning to its starting position. The high localization accuracy is illustrated by the strong overlap between the ground truth results and SLAM trajectory depicted in Figure 6.4-b. The mean error over the entire run is calculated to be 0.274 m and the maximum error is determined to be 1.264 m, demonstrating that the described SLAM approach provides accurate localization in sparse, outdoor environments. Finally, the final loop error, or the error between the SLAM solution and the ground truth once the vehicle returned to the starting point, is 0.224 m, which illustrates that the SLAM method has good global consistency over a relatively long traversal.

For the presented experiment, the average execution time for the local mapping node is 27.9 milliseconds per iteration, the average execution time for the global mapping node is 48.3 milliseconds per iteration, and the average run time for the graph optimization node is 6.3 milliseconds per iteration. In comparison, performing 3D scan registration between two point clouds generated by the Velodyne scanner using methods such as GICP or NDT requires approximately two to three seconds per iteration. The increased computation is especially problematic when generating edges for the pose graph, as multiple scan registrations are performed in succession. It is clear that the presented SLAM algorithm is able to achieve real-time performance, especially when taking into consideration that the average point cloud generation frequency for the Velodyne HDL-32E laser scanner is approximately 10 Hz, or 100 milliseconds per scan.

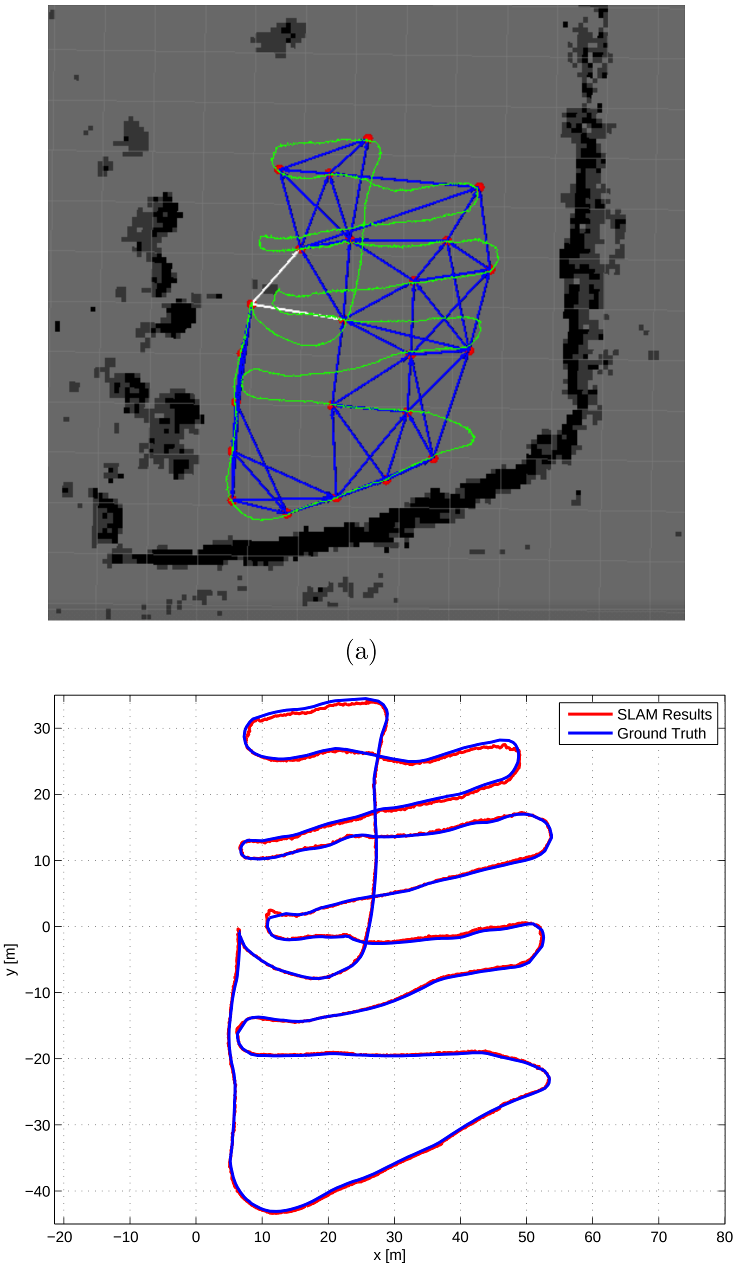
The performed tests validates that the SLAM algorithm sufficiently meets the requirements to operate in a typical autonomous mission. The approach is able to generate globally consistent maps for path planning and can localize the vehicle in with sufficient accuracy for any reasonable application. All computation is able to take place online, in real-time, using only the hardware onboard the vehicle.

(a) Map without back-end optimization



(b) Map after back-end optimization

Figure 6.3: Comparison of the maps generated with and without back-end graph optimization on the ford dataset. Black points indicate objects in the environment while the red line shows the calculated path.



(a)

(b)

Figure 6.4: Generated map and ground truth results for the sparse field experiment. Red denotes graph SLAM vertices, blue lines denote graph edges, green denotes the instantaneous vehicle path, and black denotes the map. (a) Generated Map and pose graph (b) Resulting vehicle trajectory overlaid with the vehicle ground truth determined using a Robotic Total Station.

Chapter 7

Conclusion

Autonomous vehicles are quickly becoming a reality of everyday life with these technologies having become common place in military, search and rescue, industrial, and even home applications. A robust and reliable SLAM algorithm is of vital importance to the successful operation of future and current autonomous vehicles. SLAM is a cornerstone of any robotic system, on which all other systems rely. Without accurate and robust localization and mapping, planning and task decision making would not be possible. Many modern systems use scan registration as the front-end SLAM solution of choice. Existing scan registration methods have several issues and are not reliable in all situations and environments.

Modern scan registration methods such as ICP, GICP and NDT, often struggle with unstructured environments and all fall victim to geometric degeneracies in which the registration optimization is underconstrained. These cases, such as long corridors and open spaces, are common in typical operating environments and can cause serious problems for robotics systems. Additionally, in order to maintain high accuracy solutions these method also sacrifice computational complexity and often struggle to perform in real time.

This work proposes using secondary information, either calculated from existing information or received from additional sensors, to augment the scan registration methods in order to achieve superior results with less computational expense and higher reliability. The three proposed methods, Ground Segmented ICP (GSICP), Color Clustered NDT (CCNDT), and Multi-Channel GICP (MCGICP), each demonstrate significant improvements over the basic methods with each satisfying a specific need in the scan registration spectrum.

GSICP improves upon the standard ICP algorithm for cases of sparse unstructured environments. The method segments the ground plane from a 3D laser scan, compresses the scan into a 2D representation and classifies the points into ground-adjacent and non-ground-adjacent classes. The classifications are then used to perform a class constrained ICP optimization in which only point of the same class can be corresponded. This method of classification results in creating simulated edge boundaries which improve the registration results. This method is also exceptionally computationally efficient and is capable of running at high speed on even minimally powerful platforms. This method is however limited in its applicability as it requires scans which have spatially separable objects, a defined ground plane, are geometrically constrained and do not deviate significantly from the horizontal plane. This is ideal for robots which are operating in sparse outdoor environments with minimal computational power.

CCNDT is a computationally efficient, robust method which performs full 3D scan registrations on a diverse set of possible environments. CCNDT uses a greedy color clustering method to segment the input scans and then generates a Gaussian distributions for each color cluster. The Color Gaussian distributions are then used in a color weighted distribution to distribution cost function which calculates the cost for every pair of distributions. This results in a smooth, continuous cost function which is computationally efficient to optimize and calculate. The CCNDT method shows significantly improved computational performance over the standard NDT algorithm and is applicable to any colorized laser scan which has sufficient color variation. In cases where the scans have sufficient color, the method is shown to be highly robust and accurate however when a scan is monochromatic, or close to such, the method will be unable to cluster points effectively and the algorithm will fail. This method is ideally suited for indoor applications and urban outdoor setting which typically have large variations in color. Environments such as rural or industrial setting which do not have significant variation in color would result in poor performance of this algorithm.

MCGICP is the most reliable and robust of all the presented method. It is capable of utilizing multiple channels of available information in an integrated approach, which is capable of handling many situations that result in degeneracies for other methods. MCGICP uses both geometric and descriptor channels to calculate a covariance for each point in the scans. Unlike GICP, MCGICP calculates covariance not only normal to the surface but tangent to the surface as well. This results in non-trivial error terms along the surface as well as normal to it and is therefore capable of accounting for geometric degeneracies. As well as manipulating the covariance based on descriptor channels, the correspondence calculation is also extended to incorporate both geometric, descriptor, and covariance information in order to obtain the better point to point correspondences. The MCGICP algorithm is capable of accurate, robust performance in a wider range of environments than existing approaches, including both indoor and outdoor situations. The ability to ensure reliable data in any situation is of vital importance to the proliferation of mobile robotics in the future.

As technology continues to advance and the demand for autonomous systems increases, SLAM solutions will need to continue to develop to meet the needs of future robotics applications. Future work includes incorporating the proposed methods with a full graph-SLAM back-end for loop closure and exploring other possible secondary descriptor space combinations including lighting invariant spaces and non standard imaging technologies such as IR cameras. Additionally, an investigation into profiling the proposed methods to find bottlenecks as well as possible parallelization would be beneficial in order to further improve computational performance, particularly for the MCGICP algorithm.

Moving forward, autonomous vehicles will need to focus not only on accuracy and performance but on robustness and reliability. In order to ensure the safety and productivity of the human operator and allow autonomous systems to gain wide spread adoption, these systems must be able to adapt to all possible situations they may face. SLAM continues to be a challenging and complex problem with no single clearly defined solution. The improvements to the scan registration problem presented in this work are only the next step forward in the eventual goal of developing an infallible SLAM system for the future robots that will be driving our roads and working along side us every day.

References

[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *Transactions on Pattern Analysis and Machine Intelligence*, (5):698–700, 1987.

[2] Abraham Bachrach, Samuel Prentice, Ruijie He, and Nicholas Roy. RANGERobust autonomous navigation in GPS-denied environments. *Journal of Field Robotics*, 28(5):644–666, 2011.

[3] 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.

[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 *International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2743–2748. IEEE, 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] Dorit Borrmann, Jan Elseberg, Kai Lingemann, Andreas Nüchter, and Joachim Hertzberg. Globally consistent 3D mapping with scan matching. *Robotics and Autonomous Systems*, 56(2):130 – 142, 2008.

[8] 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.

[9] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2724–2729. IEEE, Apr 1991.

[10] A. Das, J. Servos, and S.L. Waslander. 3D scan registration using the normal distributions transform with ground segmentation and point cloud clustering. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2207–2212, May 2013.

[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] Arun Das. *Scan Registation Using Normal Distributions Transform and Point Cloud Clustering Techniques.* PhD thesis, Mechanical and Mechatronics Engineering, University of Waterloo, Canada, 2013.

[13] Arun Das, Michael Diu, Neil Mathew, Christian Scharfenberger, James Servos, Andy Wong, John S Zelek, David A Clausi, and Steven L Waslander. Mapping, planning, and sample detection strategies for autonomous exploration. *Journal of Field Robotics*, 31(1):75–106, 2014.

[14] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[15] 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.

[16] 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.

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

[18] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the RGB-D SLAM system. In *International Conference on Robotics and Automation (ICRA)*, pages 1691–1696. IEEE, 2012.

[19] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard. Real-time 3D visual SLAM with a hand-held RGB-D camera. In *RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, Vasteras, Sweden, 2011.

[20] T. Fong, M. Allan, X. Bouyssounouse, M.G. Bualat, M.C. Deans, L. Edwards, L. Flückiger, L. Keely, S.Y. Lee, and D. Lees. Robotic site survey at haughton crater. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, pages 26–29, Los Angeles, USA, 2008.

[21] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.

[22] Giorgio Grisetti, Cyrill Stachniss, Slawomir Grzonka, and Wolfram Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Robotics: Science and Systems*, Atlanta, Georgia, 2007.

[23] Erico Guizzo. Three engineers, hundreds of robots, one warehouse. *Spectrum, IEEE*, 45(7):26–34, 2008.

[24] Erico Guizzo. How googles self-driving car works. *IEEE Spectrum Online*, 18, Oct 2011.

[25] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[26] 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.

[27] Benjamin Huhle, Philipp Jenke, and Wolfgang Straßer. On-the-fly scene acquisition with a handy multi-sensor system. *International Journal of Intelligent Systems Technologies and Applications*, 5(3):255–263, 2008.

[28] Benjamin Huhle, Martin Magnusson, Wolfgang Straßer, and Achim J Lilienthal. Registration of colored 3D point clouds with a kernel-based extension to the normal distributions transform. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4025–4030, 2008.

[29] Velodyne Acoustics Inc. *Velodyne HDL-64E User Manual: High Definition Lidar Sensor*. Morgan Hill, CA.

[30] Andrew Edie Johnson and Sing Bing Kang. Registration and integration of textured 3D data. *Image and vision computing*, 17(2):135–147, 1999.

[31] Joseph L Jones. Robots at the tipping point: the road to iRobot Roomba. *IEEE Robotics & Automation Magazine*, 13(1):76–78, 2006.

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

[33] Sören Kammel, Julius Ziegler, Benjamin Pitzer, Moritz Werling, Tobias Gindele, Daniel Jagzent, Joachim Schröder, Michael Thuy, Matthias Goebl, Felix von Hundelshausen, Oliver Pink, Christian Frese, and Christoph Stiller. Team AnnieWAY's autonomous system for the 2007 darpa urban challenge. *Journal of Field Robotics*, 25(9):615–639, 2008.

[34] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234, Nara, Japan, 2007.

[35] David M Knuth, Henry L Hillman, et al. Method of cleaning a surface using an automatic cleaning device, Oct 2008. US Patent App. Intellibot Robotics LLC 12/261,703.

[36] 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.

[37] Kurt Konolige, Motilal Agrawal, and Joan Sol. Large-scale visual odometry for rough terrain. In Makoto Kaneko and Yoshihiko Nakamura, editors, *Robotics Research*, volume 66 of *Springer Tracts in Advanced Robotics*, pages 201–212. Springer Berlin / Heidelberg, 2011.

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

[39] 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.

[40] Jesse Levinson and Sebastian Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4372–4378, 2010.

[41] J.S. Levinson. *Automatic Laser Calibration, Mapping, and Localization for Autonomous Vehicles*. PhD thesis, Stanford University, 2011.

[42] D.G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1150–1157, Kerkyra, Greece, September 1999.

[43] 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.

[44] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application (VISSAPP)*, pages 331–340. INSTICC Press, 2009.

[45] R.R. Murphy. Trial by fire [rescue robots]. *Robotics Automation Magazine, IEEE*, 11(3):50–61, Sept 2004.

[46] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, et al. Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63, 2013.

[47] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, Oct 2011.

[48] Richard A. Newcombe, S.J. Lovegrove, and A.J. Davison. DTAM: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2320–2327, Nov 2011.

[49] 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.

[50] 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.

[51] 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.

[52] CUDA Nvidia. *Compute unified device architecture programming guide*. Santa Clara, CA.

[53] 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.

[54] Gaurav Pandey, James R McBride, Silvio Savarese, and Ryan Eustice. Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. In *AAAI*, 2012.

[55] Gaurav Pandey, JR McBride, S Savarese, and Ryan M Eustice. Visually bootstrapped generalized ICP. In *International Conference on Robotics and Automation (ICRA)*, pages 2660–2667. IEEE, 2011.

[56] Patrick Pfaff, Rainer Kümmerle, Dominik Joho, Cyrill Stachniss, Rudolph Triebel, and Wolfram Burgard. Navigation in combined outdoor and indoor environments using multi-level surface maps. In *Workshop on Safe Navigation in Open and Dynamic Environments at the 2007 IEEE International Conference on Intelligent Robots and Systems (IROS)*, San Diego, CA, Oct 2007.

[57] Jeremy Rifkin. *The end of work*. Putnam Publishing Group, New York, USA, 1995.

[58] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[59] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

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

[61] James Servos and Steven L Waslander. Multi-channel GICP. In *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, June 2014.

[62] James Servos and Steven L Waslander. Using RGB information to improve NDT distribution generation and registration convergence. In *International Conference on Intelligent Unmanned Systems (ICIUS)*, Montreal, Canada, Oct 2014.

[63] 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.

[64] Bastian Steder, Radu Bogdan Rusu, Kurt Konolige, and Wolfram Burgard. Point feature extraction on 3D range scans taking into account object boundaries. In *IEEE International Conference on Robotics and automation (ICRA)*, pages 2601–2608, 2011.

[65] T. Stoyanov, M. Magnusson, and A.J. Lilienthal. Point set registration through minimization of the L2 distance between 3D-NDT models. In *International Conference on Robotics and Automation (ICRA)*, pages 5196–5201, St. Paul, MN, USA, May 2012. IEEE.

[66] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *International Conference on Intelligent Robot Systems (IROS)*, Vilamoura, Portugal, Oct 2012.

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

[68] C.H. Tong, T.D. Barfoot, and É. Dupuis. Three-dimensional SLAM for mapping planetary work site environments. *Journal of Field Robotics*, 29(3):381–412, 2012.

[69] 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.

[70] Michael John Tribou. *Relative Pose Estimation Using Non-overlapping Multicamera Clusters*. PhD thesis, Mechanical and Mechatronics Engineering, University of Waterloo, Canada, 2014.

[71] R. Triebel, P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 2276 –2282, Beijing, China, October 2006.

[72] 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.

[73] 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.

[74] 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.

[75] Dana R Yoerger, M Bradley Albert, B Walden Barrie, Hanumant Singh, and Ralf Bachmayer. Surveying a subsea lava flow using the autonomous benthic explorer (ABE). *International Journal of Systems Science*, 29(10):1031–1044, 1998.

[76] Qingming Zhana, Yubin Liang, and Yinghui Xiaoa. Color-based segmentation of point clouds. In *ISPRS Workshop on Laserscanning (IAPRS)*, volume XXXVIII, 2009.