AFRL-RY-WP-TR-2021-0014

FEATURE EXTRACTION BASED ITERATIVE CLOSEST POINT REGISTRATION FOR LARGE SCALE AERIAL LIDAR POINT CLOUDS

Quinn R. Graehling
University of Dayton

JANUARY 2021
Final Report

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
January 2021	Thesis	16 December 2020 –16 December 2020

4. TITLE AND SUBTITLE
FEATURE EXTRACTION BASED ITERATIVE CLOSEST POINT REGISTRATION FOR LARGE SCALE AERIAL LIDAR POINT CLOUDS

5a. CONTRACT NUMBER
FA8650-20-F-1925

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61102F/62204F/63203F

6. AUTHOR(S)
Quinn R. Graehling

5d. PROJECT NUMBER
3001/2003/665A

5e. TASK NUMBER
N/A

5f. WORK UNIT NUMBER
Y1Z3

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Dayton
300 College Park
Dayton, OH 45469

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command
United States Air Force

Air Force Office of Scientific Research
Public Affairs (AFOSR)
875 North Randolph Street, Suite 325, Room 3112
Arlington, VA 22203

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RYMM

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RY-WP-TR-2021-0014

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
PAO case number AFRL-2020-0451, Clearance Date 16 December 2020. © 2021 Quinn R. Graehling. Submitted to the College of Engineering of the University of Dayton in partial fulfillment of the requirements for the degree of Master of Science in Computer Engineering.. This work was funded in whole or in part by Department of the Air Force. The U.S. Government has for itself and others acting on its behalf an unlimited, paid-up, nonexclusive, irrevocable worldwide license to use, modify, reproduce, release, perform, display, or disclose the work by or on behalf of the U.S. Government. This report is from Task Order 0001 of parent contract FA8650-20-D-1013. Report contains color.

14. ABSTRACT

Image registration is a major field within computer vision and is often a required step in properly fulfilling other computer vision and pattern recognition tasks such as change detection, scene classification and image segmentation. Recent advances in 3D computer vision and lowered costs in Light Detection and Ranging devices, better known as LiDAR, have given way to an increase in readily available 3D image datasets. These 3D captures give an extra dimension to computer vision data and allow for improvements in a multitude of tasks when compared to their 2D counterparts. However, due to the large scale and complex nature of 3D point cloud data, classical methods for registration often require increased hardware usage and time and can fail to proper register data with a low degree of error. The strategy presented in this paper aims to reduce the number of points representing a point cloud in order to reduce time and hardware overhead needed to perform registration while allowing the algorithm to improve registration accuracy and reduce error between registered clouds. This is done by extracting key edge features from the point clouds using eigenvector analysis to remove ground planes and large normal planes within the point cloud. The algorithm is further improved by performing set differencing on two separate edge extractions to remove large clusters of points representing natural objects that can often cause confusion for registration of outdoor LiDAR scenes. The method for key point registration is evaluated on large scale, complex LiDAR point clouds obtained from aerial sensors. Tests are performed on both fully overlapping and partially overlapping clouds to ensure that the method increases performance on full and partial registration tasks. The tests are also performed on clouds of varying resolution to test the algorithms ability to maintain integrity regardless of cloud resolution. Point reduction results, registration statics and visual results are presented for comparison. A brief look into possible applications of the method and future improvements to the algorithm are included.

15. SUBJECT TERMS
feature extraction, image registration, LADAR, lidar, LiDAR point cloud

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Andrew Stokes
Unclassified	Unclassified	Unclassified	SAR	56	19b. TELEPHONE NUMBER *(Include Area Code)* N/A

FEATURE EXTRACTION BASED ITERATIVE CLOSEST POINT REGISTRATION FOR LARGE SCALE AERIAL LIDAR POINT CLOUDS

Thesis

Submitted to

The College of Engineering of the

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Master of Science in Computer Engineering

By

Quinn R. Graehling

Dayton, Ohio

December 2020

FEATURE EXTRACTION BASED ITERATIVE CLOSEST POINT REGISTRATION FOR LARGE SCALE AERIAL LIDAR POINT CLOUDS

Name: Quinn Graehling

APPROVED BY:

Vijayan K. Asari, Ph.D.
Advisory Committee Chairman
Professor, Department of Electrical and Computer Engineering

Theus H. Aspiras, Ph.D.
Committee Member
Professor, Department of Electrical and Computer Engineering

Eric J. Balster, Ph.D.
Committee Member
Department Chair, Professor, Department of Electrical and Computer Engineering

Robert J. Wilkens, Ph.D., P.E.
Associate Dean for Research and Innovation
Professor
School of Engineering

Eddy M. Rojas, Ph.D., M.A., P.E.
Dean, School of Engineering

ABSTRACT

FEATURE EXTRACTION BASED ITERATIVE CLOSEST POINT REGISTRATION FOR LARGE SCALE AERIAL LIDAR POINT CLOUDS

Name: Quinn R. Graehling
University of Dayton

Advisor: Dr. Vijayan K. Asari, Ph.D.

Image registration is a major field within computer vision and is often a required step in properly fulfilling other computer vision and pattern recognition tasks such as change detection, scene classification and image segmentation. Recent advances in 3D computer vision and lowered costs in Light Detection and Ranging devices, better known as LiDAR, have given way to an increase in readily available 3D image datasets. These 3D captures give an extra dimension to computer vision data and allow for improvements in a multitude of tasks when compared to their 2D counterparts. However, due to the large scale and complex nature of 3D point cloud data, classical methods for registration often require increased hardware usage and time and can fail to proper register data with a low degree of error.

The strategy presented in this paper aims to reduce the number of points representing a point cloud in order to reduce time and hardware overhead needed to perform registration while allowing the algorithm to improve registration accuracy and reduce error between registered clouds. This is done by extracting key edge features from the point clouds using eigenvector analysis to remove ground planes and large normal

planes within the point cloud. The algorithm is further improved by performing set differencing on two separate edge extractions to remove large clusters of points representing natural objects that can often cause confusion for registration of outdoor LiDAR scenes.

The method for key point registration is evaluated on large scale, complex LiDAR point clouds obtained from aerial sensors. Tests are performed on both fully overlapping and partially overlapping clouds to ensure that the method increases performance on full and partial registration tasks. The tests are also performed on clouds of varying resolution to test the algorithms ability to maintain integrity regardless of cloud resolution. Point reduction results, registration statics and visual results are presented for comparison. A brief look into possible applications of the method and future improvements to the algorithm are included.

Dedicated to my parents, whose character and virtue have been a constant encouragement and inspiration to my own life.

ACKNOWLEDGMENTS

I would like to that Dr. Asari, Dr. Aspiras and Nina Varney as well as the rest of the team members in the University of Dayton's Vision Lab for there support and assistance in my research. The knowledge and skills I have gained from my time with them has been fundamental in my growth at the University and in my personal and professional life.

This work is supported by the United States Air Force Research Labs. I would Like to thank AFRL for providing the means without which this research would not be possible. I would also like to that the City of Surrey Engineering Department for providing the data used for this research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Image registration is seen as one of the pillars of computer vision and object recognition in the modern era. The ability to register large and complex datasets into a single aligned coordinated system is often one of the main tasks for image processing tasks and is required for many high-level applications such as change detection and object tracking. Early methods for achieving image registration between sets of 2-dimensional images relied on robust feature extraction techniques such as Harris Corners [1] and Sobel filters [2] to create a set of features that could be matched between two sets of images in order to perform image alignment. Though more descriptive and advanced feature extraction methods would be developed and implemented for 2D image registration, the general method of extracting key features from a set of images and aligning those features remained the dominant technique for 2D image registration. In the early 2000's, 3D computer vision data began to appear in the form of voxel grids and computer-aided design (CAD) structures. Though techniques for object recognition and segmentation began to be designed for these early datasets, such as VoxNet [3] in Figure 1, and multimodal RGB-D learning [4], 3D registration was often not required as the majority of these datasets contained artificially generated data that had no need to be registered or that could be easily registered using classical 2D methods.

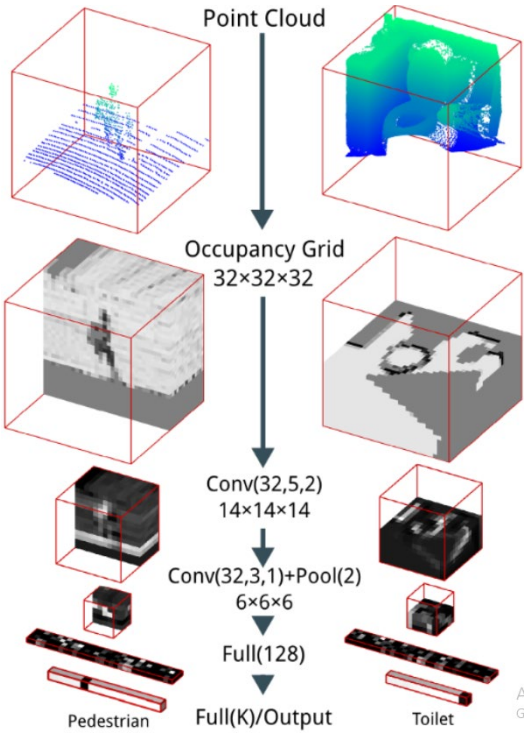

Figure 1: Network architecture for VoxNet. VoxNet is a deep learning architecture aimed at real-time recognition of 3D objects within point clouds. The network first preprocesses a point set into a rigid structure comprised of volumetric pixels known as voxels. These voxels can then be extracted into a feature vector that the network is capable of learning from [3].

Soon after these early computer-generated datasets were introduced, image capture devices were created that could capture an accurate 3D representation of an environment. One of the more prominent devices for capturing 3D data, LiDAR, represented its captured data in the form of data points with listed coordinate values known as point clouds. Recent advances in LiDAR data, along with the reduced costs in hardware, has led to a massive increase in the use of LiDAR devices to capture large and complex scenes. This has subsequently led to an increase in the amount of present point cloud data and a demand for modern methods for performing registration on 3D point cloud datasets.

Registrations methods for 2D images often relied on feature extraction and alignment to perform their accurate registration. Though the methods for feature extraction have changed and advanced, this overarching theme has remained constant. Due to the success and commonality of this technique for 2D registration, early methods of 3D registration were often based upon this method. These feature extraction methods usually consisted of hand-crafted geometric or deep learning-based feature extraction and pairing. Other methods for coarse registration aimed to normalize point distributions between two clouds [5]. Though this often worked for coarse registration, the increased complexity of point cloud data often meant that the computing resources and time required to perform registration this way were often very high. Furthermore, large-scale point cloud data of point clouds captured from aerial devices of large landscapes, such as the DALES dataset [6], were often too complex for these methods and still have significant point alignment error after coarse registration was applied. Other methods, such as Iterative Closest Point [7] and Robust Point Matching [8], aimed to align point clouds by iteratively reducing the distance of nearest neighboring points until a stabilized position was reached. Though these methods proved useful for fine registration, they tended to fail to perform proper registration when given large scale point clouds with various regular planes that often interfered with the stabilization error. This paper presents Feature Extraction Aerial Registration, a method that combines a robust feature extraction method for identifying and extracting key features within a large scale point cloud and a fine pairwise registration method to allow for highly accurate registration of large scale aerial LiDAR data without the need for coarse registration or deep learning. Figure 2 below illustrates the proposed method.

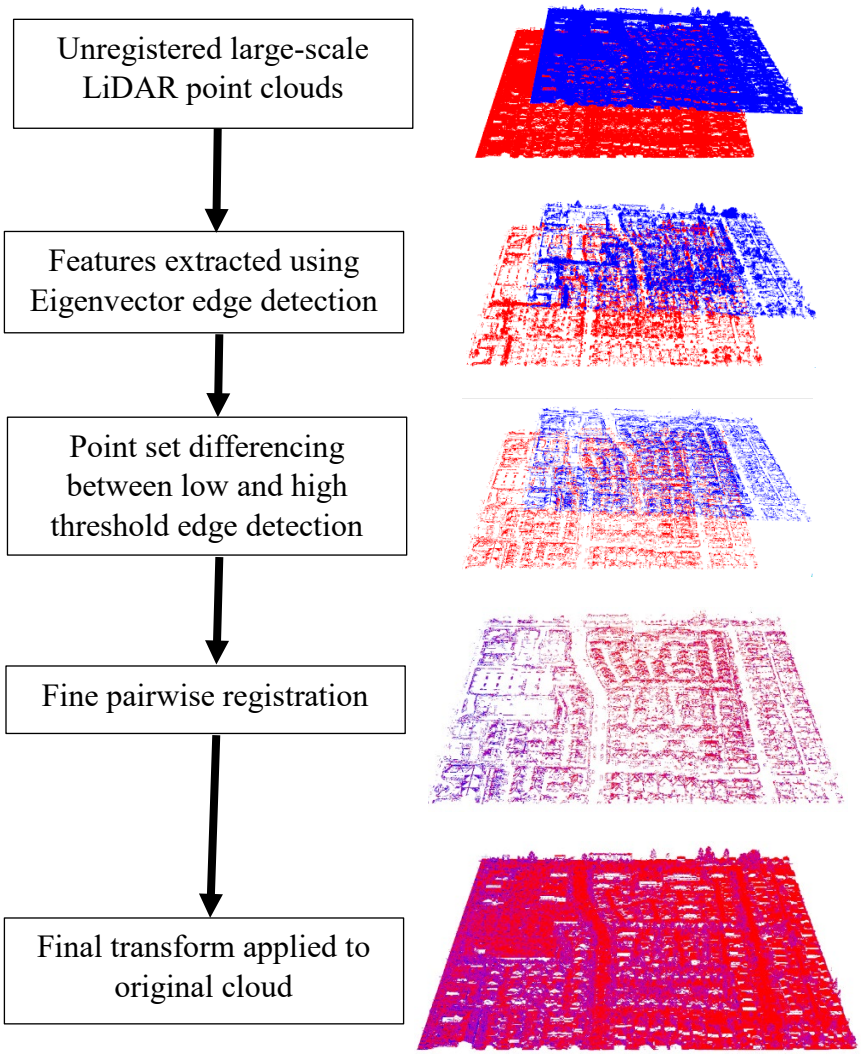

Figure 2: Illustration of proposed method. First edges are extracted from two unregistered point clouds using Eigenvector analysis for sharp feature detection. Two separate edge thresholds are detected, and the resulting low and high threshold clouds are point differenced to extract fine sharp edges and remove point clusters. These clouds are then registered using a KD-tree based ICP method for fine pairwise registration. The final transformation of the registered edge clouds is applied to the original cloud to get a highly accurate registration method that does not require coarse registration or deep learning methods.

CHAPTER 2

LITERATURE SURVEY

2.1 Pairwise Registration Methods

This paper will focus on performing pairwise registration on high resolution, dense point cloud data captured using aerial LiDAR sensors. Modern approaches to registration of point set data consists of course and fine pairwise registration. Coarse registration methods are often used to get an initial plane and spatial alignment between two clouds in order to reduce the number of iterations needed to perform fine pairwise registration, which are often more hardware and time consuming than course registration. Fine registration methods are used to further align point set data into more rigid and highly aligned structures with low distance errors between the paired points. We will not discuss some modern approaches to both coarse and fine pairwise registration.

2.1.1 Pairwise Coarse Registration

Registration through handcrafted features focuses on a two-step process. The first is the extraction of the features themselves, these are most commonly geometric features. The second step is finding the correspondences between these features. We can separate the geometric features to include point-based, line-based, and surface-based. Point-based includes methods such as Fast Point Feature Histograms (FPFH) [9], spin images, projection density, geometric shape constraint, 3D distance of point pairs, Four Point Congruent Sets (4PCS) [10], and SIFT features [11], where these features are thought of

as key points and then fed to the correspondence algorithm. These point feature methods have a high sensitivity to noise and have low computational efficiency.

Line-based and surface-based extraction methods include those based on Laplacian matrix decomposition, random sample consensus (RANSAC) [12], least squares, and Principle Component Analysis [13]. These methods have a high level of accuracy but can struggle in natural scenes, where there are not many sharp edges and the number of features is relatively low. One additional disadvantage of these methods is they are sensitive to noise and changes in the point cloud density.

In contrast to the use of handcrafted features, many deep learning-based methods provide a robust and generalized way of extracting features, without the need for excessive parameter adjustment. These methods can be separated into two different approaches, the first is a voxelization approach such as that proposed in 3DMatch [14] which can be seen in Figure 3 below. The voxelization method solves the problem of transforming an unstructured point cloud into a structured 3D grids but is also inefficient from a memory perspective and can result in an information loss which is highly dependent on the chosen size of the grid.

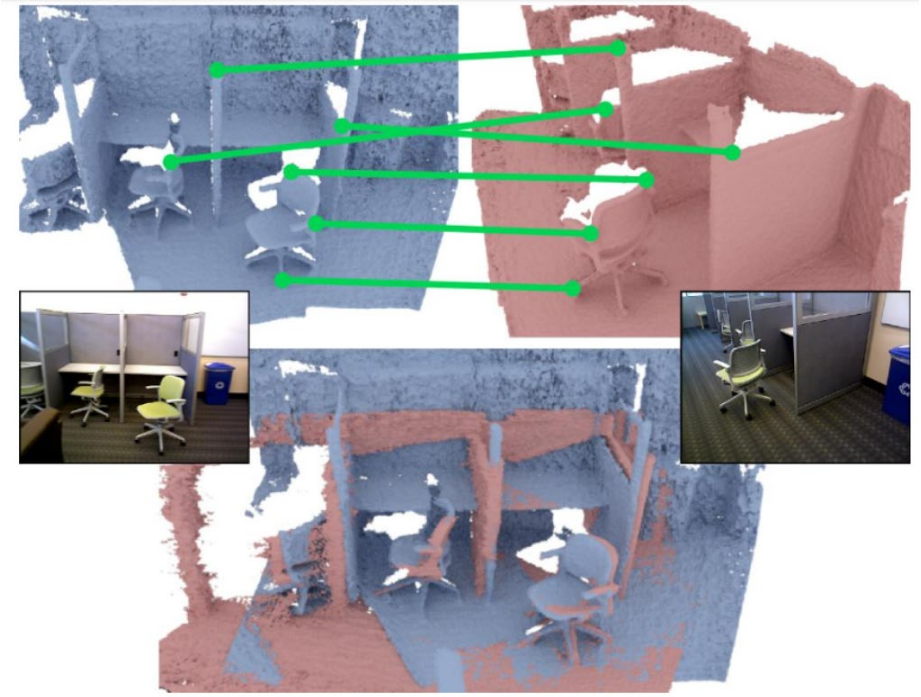

Figure 3: 3DMatch network aligning two voxelized point sets generated from RGB-D data. 3DMatch learns volumetric descriptors from 3D voxel information within a 3D structure. These descriptors are matched between two separate voxel sets and these matches are used to align the 3D data [14].

Methods such as PointNet [15], shown in Figure 4 below, and PPFNet [16] provide a way to calculate point-based features from the unstructured cloud, while the Deepest Closest Point [17] method uses a graph convolutional network in an embedded space to deduce relationships between a cloud pair. Deep learning methods are generally robust in their ability to calculate point features, but they have traditionally struggled with larger more complex outdoor datasets.

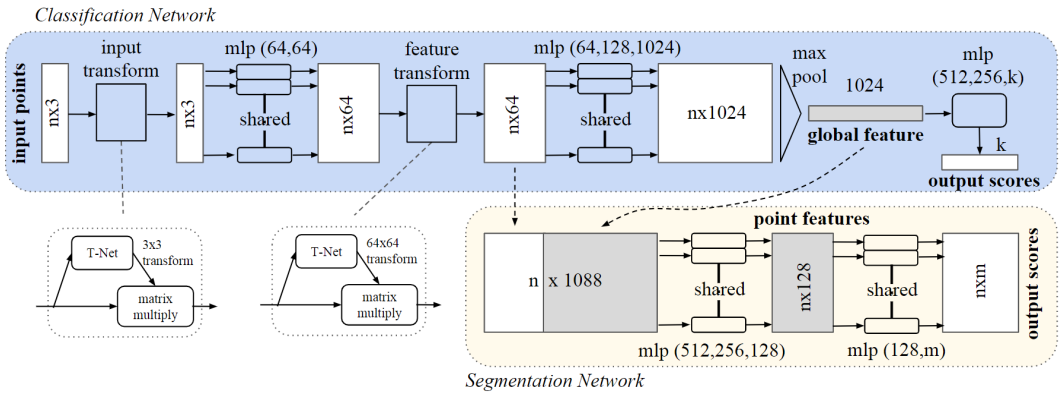

Figure 4: Pointnet architecture for feature extraction and segmentation of unstructured point clouds. Pointnet uses a series of multilayer perceptrons and a max pooling function to classify point sets into global features. Unlike prior networks for 3D feature learning that utilized preprocessing in order to construct ordered voxel grids from point sets, Pointnet was one of the first networks to offer a learning structure that operates directly on an unorder set of points [15].

2.1.2 Pairwise Fine Registration

The Iterative Closest Point (ICP) registration method is the most commonly used fine registration algorithm. This method uses a selection of points from each cloud and minimizes the distance between the two-point sets, a transformation is performed and then the process is repeated until there is little to no change in the transformations between consecutive iterations. This method is relatively simple and can be used in many applications, however, one downside is that it assumes a one to one correspondence between the two-point clouds. Differences in time of collection, noise, and variations in point density and areas of occlusion, no two LiDAR point clouds have this one to one correspondence presenting a fundamental challenge. Many variations of this algorithm have attempted to solve this problem with methods such as point-to-point, point-to-plane, point-to-projection, and plane-to-plane.

In addition, many algorithms have sought to improve efficiency. The iterative nature of ICP means that the speed suffers when processing dense, high-resolution point

clouds. Techniques like [18] use a KD- tree structure to allow for the implementation of the algorithm on a GPU, while [19] uses octree structures to allow for a multi-resolution ICP for both local and global registration which add in proper registration speed. Limitations of ICP can be summarized by the need for a good initialization to avoid settling on local minima and the need for an accurate correspondence algorithm to avoid errors in the closest point correspondences.

2.2 Feature Extraction Methods

This paper aims to produce a method of registration for high resolution aerial LiDAR data that does not require an initial coarse pairwise registration but still alleviates the need for a high multitude of fine registration iterations by performing a key feature extraction and performing fine registration on these extracted features. By design this would reduce the number of pairwise points needed to be aligned by the registration method while still maintaining the integrity of the original cloud. We will now explore several of the modern approaches for feature extraction of 3D point cloud data.

2.2.1 Non-Deep Learning Methods

The 3D Harris corner method for feature extraction is a common method for extracting corners and edges within a 3D point set. This method was based off the 2D Harris Corner [1] method for extracting key features from a 2D image. This method applied an edge filter to an image in order to extract edge features and then identified edges by highlighting junctions of 2 edges. As opposed to earlier methods, such as the Moravec corner detector [20], the Harris corner determined features based on directional differentials which greatly improved corner identification. The 3D extension of the Harris corner simply applies the directional differential labelling of Harris corner to a 3-

dimensional spatial relationship between point clusters within a point set. Though this method proved useful for early feature extraction using 3D data, it often failed to properly detect corners within sparsely populated regions and often over labeled corners in dense point clusters that are common within point cloud data. Other methods aimed to exploit triangulation of nearest neighbor clusters within a point cloud. One such example was the extraction method presented by Weber et al. [21] which utilized point set triangulation to form point normal as seen in Figure 5. These normals were then mapped and ranked based on their Eigenvalues. This allowed for edge extraction of both corners and edges from large scale point sets, but this method suffered from large computational requirements needed to properly extract normals. A later method that performed on par with this method but greatly reduced computational overhead was Bazazian's eigenvector edge extraction [22] which focused solely on determining edges by eigenvector values.

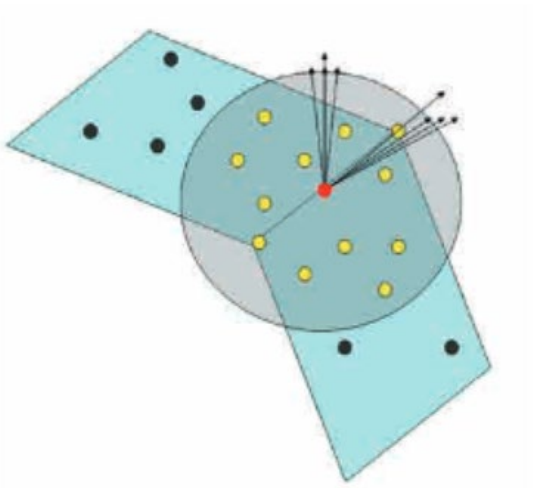

Figure 5: Illustration of point normal calculation for detection of sharp features in point sets. This method detects local neighborhood normal based on their triangulated orientation. These normals are then placed in a Gaussian mapping function to determine the sharpest normals within a cloud [21].

2.2.2 Deep Learning Methods

Another common approach for extracting features from point clouds is through deep learning networks. Often used for classification and segmentation tasks, these networks perform some sort of feature extraction on point clouds and then learn these features through classical and novel neural networks in order to perform various learning tasks. Architectures such as PointNet++ [23] and KPConv [24] learn features from point sets based on various characteristics such as point relations, clustering and kernel convolution to learn both local and global features. Other networks like VoxelNet [25] preprocess large clouds into 3D grid structures and learn from the volumetric pixel relations. Using deep learning networks to learn key features from point sets is a very powerful tool that allows features to be extracted that directly correlate to classifications of data within the point cloud. However, due to the nature of deep learning networks, these extracted features are often contained within hidden layers and feature spaces and therefore it is unknown if the extracted features would be useful for registration until the final registration has already been performed. This requires that the network be trained on data for the specific purpose of cloud registration. This training can often be time consuming and require large amounts of hardware as well as a diverse dataset that can be learned from. Even after training is properly performed, the feature extraction and classification functions of deep learning networks can still take an excessive amount of time and hardware to perform when compared to the more classical and robust counterparts.

CHAPTER 3

FEATURE EXTRACTION AND REGISTRATION

The registration method described in this paper can be broken down into two major sub-sections. The first part of the registration process is the edge extraction. The process used for extracting edge features from the large aerial point cloud data is 3D eigenvector approaches presented by [22]. This approach was selected due to the speed at which it could perform edge detection on large point sets as well as its accuracy and robustness when identifying complex 3D edges. The second section of the registration is the actual process of point set alignment. For this task the kd-tree ICP method was selected for use due to its ability to perform fine registration at high speeds. Furthermore, all of these methods are capable of being implemented without any supervision or training which is a necessary component for deep learning-based approaches.

3.1 Eigenvector Edge Detection

The edge detection and extraction method presented in [22] is based largely on the sharp feature detection algorithm presented by [21] This algorithm applies a k-nearest-neighbor triangulation approach where each point p and its k nearest neighbors are used to create a set of triangles for p where each triangle is represented by a normal vector, thus creating a collection of normals for each point. These normals are then mapped to a Gaussian map where each normal cluster is considered a point on the map. The clustering maps would be hierarchally merged and their placement on the Gaussian maps would determine their label as a plane point or a feature point. Due to the need to

form a triangulation cluster with each point in the set, Weber's method would often take up large amounts of time and hardware resources in order to complete edge detection. Thus, Bazazian asserted an approach that would utilize Principle Component Analysis (PCA) to determine the normals for each point followed by Weber's approach for merging and mapping to label feature and plane points. Since the PCA approach used by Bazazian was based on eigenvalues of covariance matrices, another method was implemented that simply utilized the eigenvalues to determine points representing edges. This allowed for an evaluation and labeling of points within a point set as either representing a plane or an edge while greatly increasing the speed and decreasing the computational overhead needed for Weber's approach. To do so this method utilizes a covariance matrix *C* for determining eigenvalues of a particular point.

$$C = \begin{bmatrix} Cov(x,x)\ Cov(x,y)\ Cov(x,z) \\ Cov(y,x)\ Cov(y,y)\ Cov(y,z) \\ Cov(z,x)\ Cov(z,y)\ Cov(z,z) \end{bmatrix} \tag{1}$$

The covariance of each element of the matrix can be calculated as

$$Cov(x,y) = \frac{\sum_{i=1}^{k}(x_i-\bar{x})(y_i-\bar{y})}{n-1} \tag{2}$$

The eigenvalues of the covariance are then used to analyze the surface variation as follow

$$\sigma_k(p) = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2} \tag{3}$$

Where the eigenvalues from C are

$$C{:}\lambda_0 \leq \lambda_1 \leq \lambda_2 \tag{4}$$

This surface variation, σ_k, is calculated for each point p with k nearest neighbors. The results variation tells if the point lies within a plane or is representing an edge. Figure 6 below shows the resulting block diagram for the edge extraction.

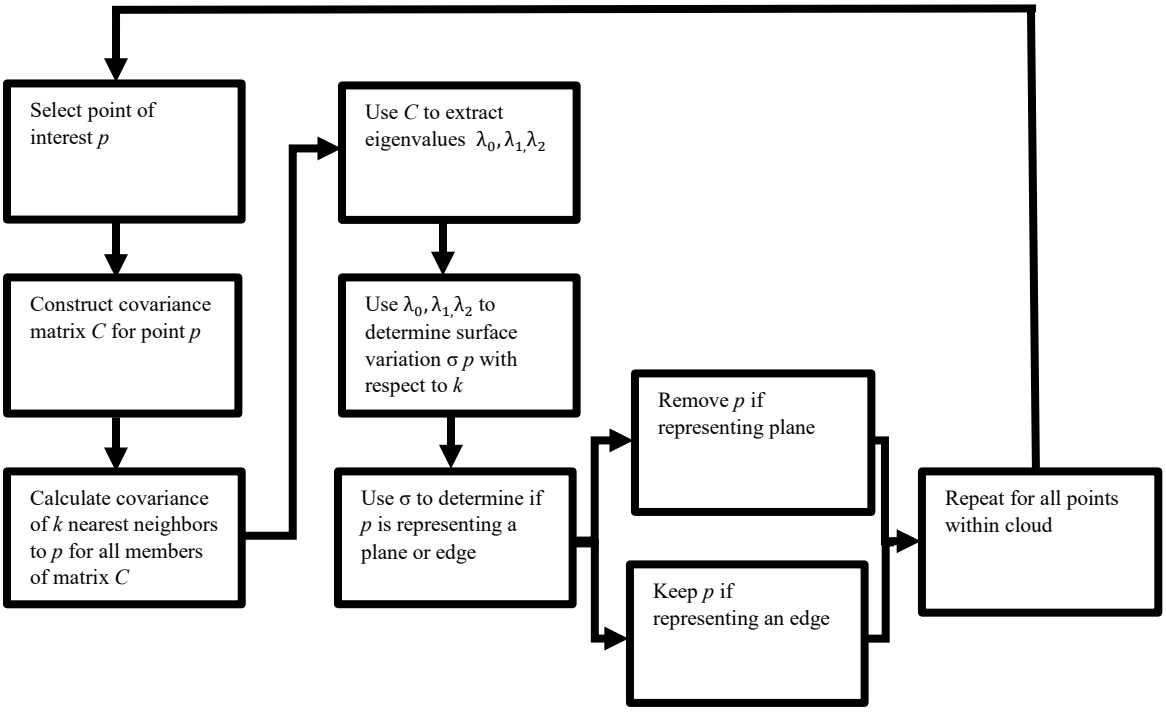

Figure 6: Block diagram for extraction of edge points using eigenvalues.

Figure 7 below shows the results of performing eigenvector edge extraction on the Surrey point cloud data when compared to the original cloud.

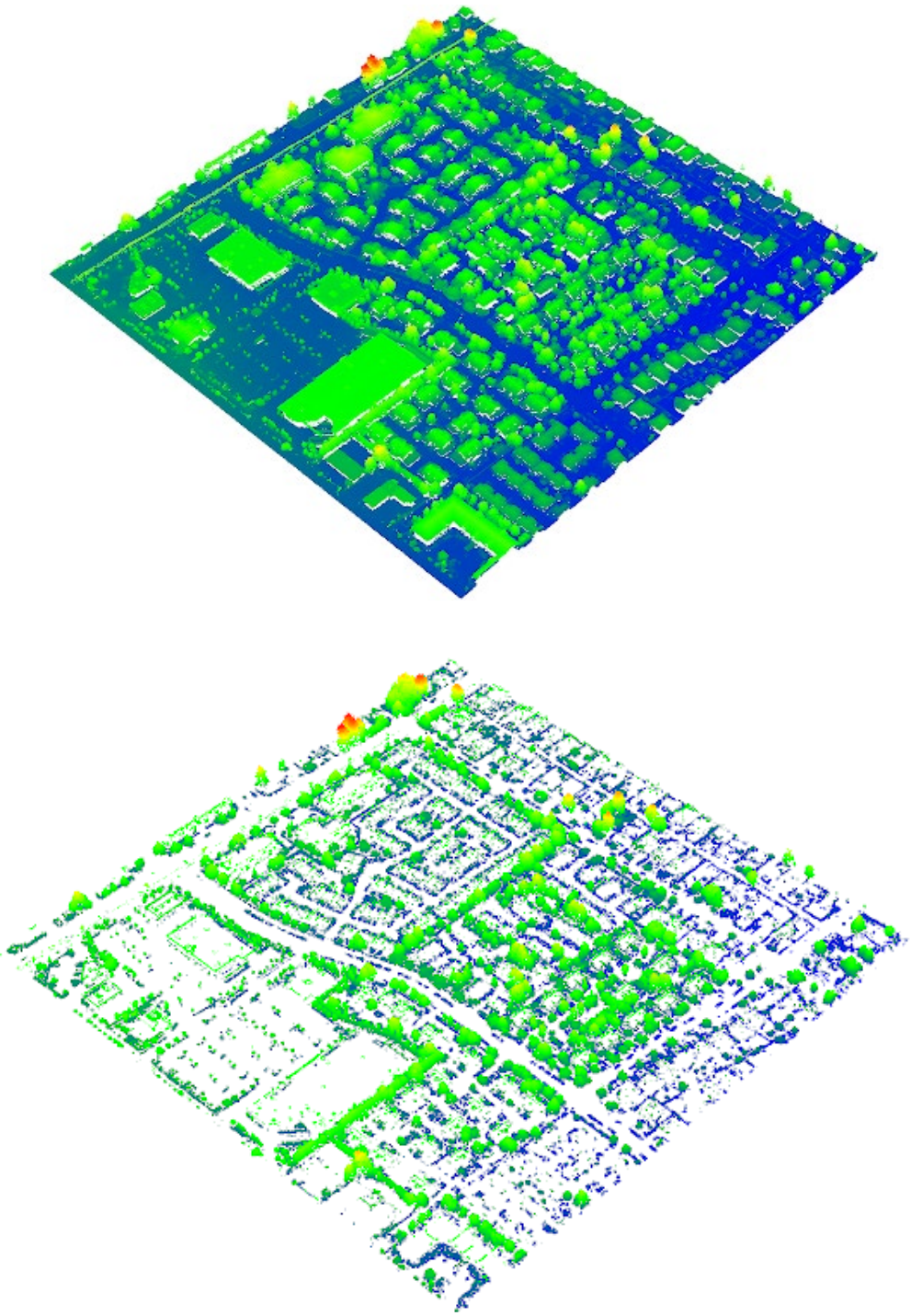

Figure 7: Effects of key feature extraction via eigenvector edge extraction. The top image depicts the original Surrey point cloud consisting of roughly 11 million points. The bottom image shows the same point cloud after applying the eigenvector edge extraction. The eigenvector edge extraction effectively removed the ground plane while maintaining the integrity of important key points for registration, reducing total cloud points to 2.7 million points.

3.2 Edge Differencing

The edge extraction method initially performed very well on large scale aerial point cloud data. However, though the method was capable of extracting sharp edge features from man-made objects, it also often labeled a large number of feature points that represented natural objects such as trees and shrubbery. Though points are correctly identified by the algorithm as representing non-plane points, the large variance of close edges within these objects and the variance of points detected as edges in them often meant that these natural objects were represented by large and incoherent sets of feature points that would only serve to confuse the ICP step used for registration. These large clusters of natural points provide a false sense of proper registration and low error for the ICP algorithm and therefore are often selected for pairwise matching over the more appropriate manmade features. Since the registration of the natural points provide a false sense of accuracy for the registration, the negative results of utilizing them for key point registration is only noticeable to the algorithm after the final transform is performed and the final error is calculated. However, since the nature objects also represented a large number of sharp edges, raising the eigenvector threshold would often remove imported man-made edges such as buildings and cars before removing the natural objects. Therefore, two separate edge detections on each cloud where the first detection used a lower threshold value in order to extract both the manmade and natural features and the second detection used a higher threshold value to extract mainly natural features. Point set differencing was then used to remove points that were present in both clouds, effectively removing large clusters of points representing natural objects while

maintaining edge extraction on man-made features. Figure 8 shows the block diagram of the edge differencing function.

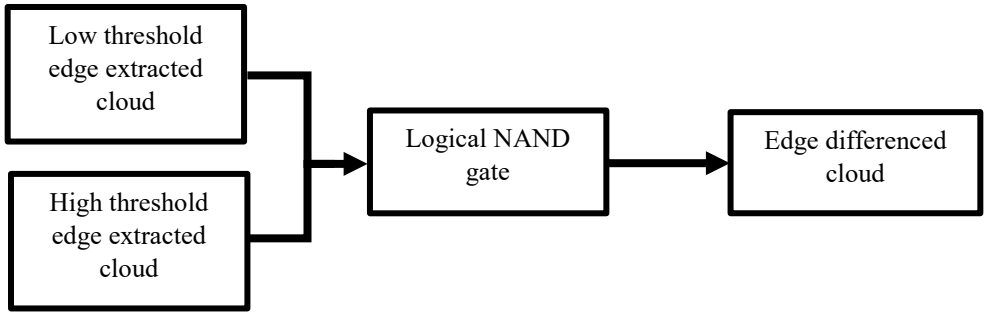

Figure 8: Block diagram for edge differencing function.

Furthermore, the lower threshold edge extraction is capable of lower angle manmade edges within the point cloud that were originally missed by the higher threshold. This allows for the point differencing to pull out more manmade edges that were originally missed, giving more key points to perform the registration on. Figure 9 below shows the effects of the point differencing on the point cloud. Figure 10 shows a more prominent example of the point set difference's ability to remove nature objects within a scene.

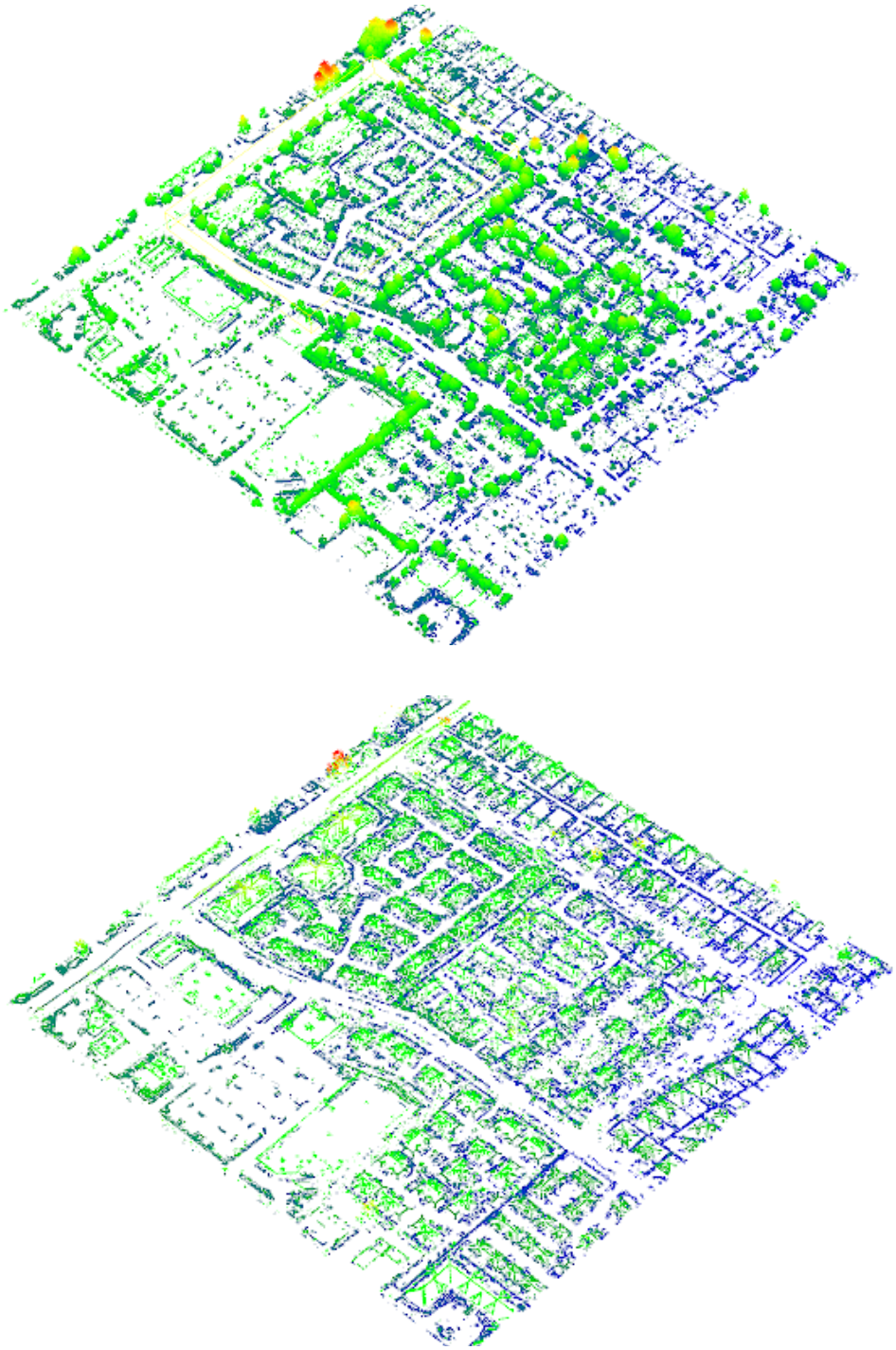

Figure 9: Effects of point differencing using two different eigenvector extracted clouds. The top image shows a cloud with edge extraction performed at an eigenvector threshold of .3 where the cloud is represented by roughly 2.7 million points. The bottom side shows a cloud formed by point differencing two clouds. The first cloud used an eigenvector threshold of .3 while the second cloud used an eigenvector threshold of .5. The differenced cloud is represented by roughly 1.1 million points.

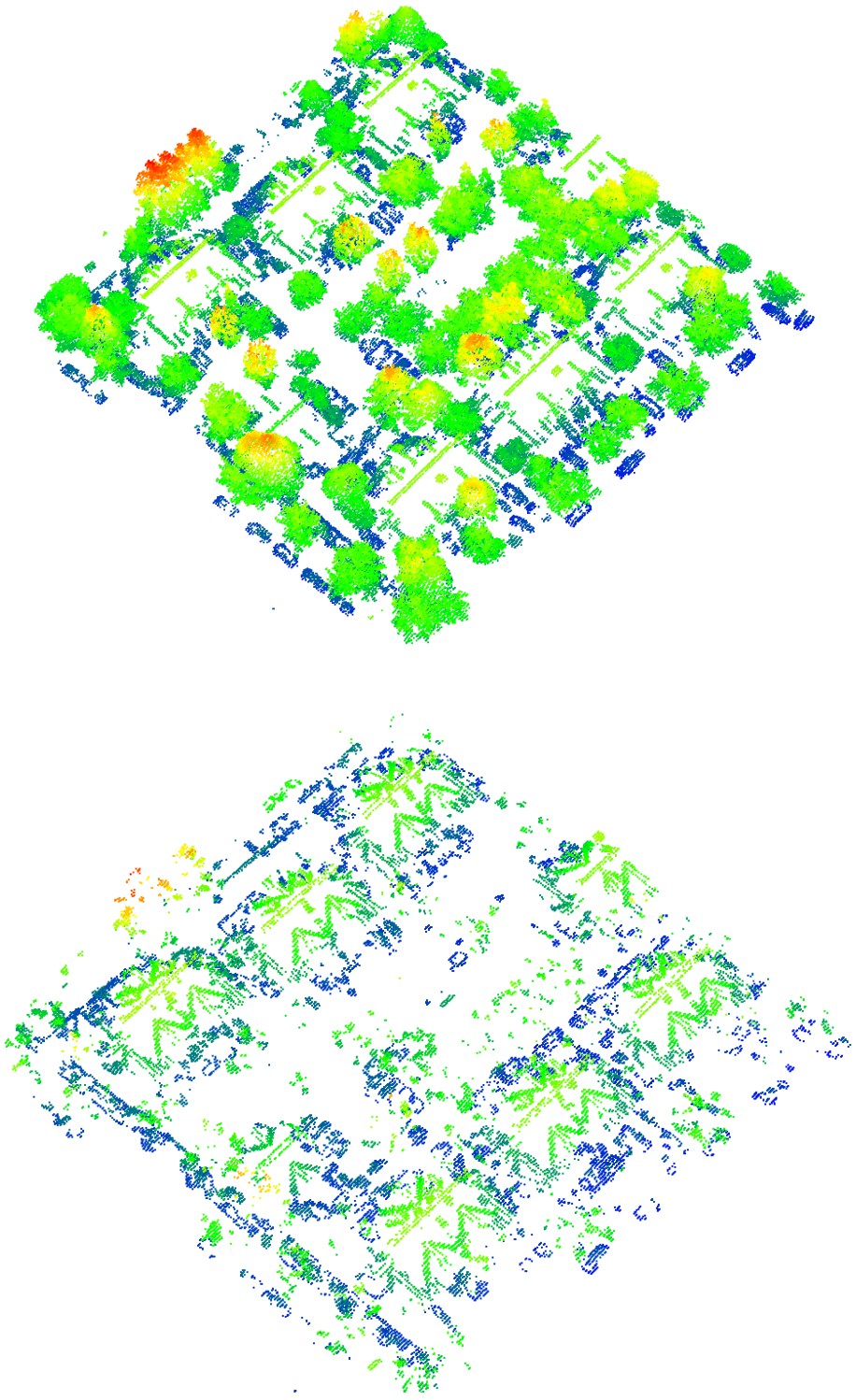

Figure 10: Example of effects of point differencing on removing large point clusters representing natural objects. The top image shows an edge extracted cloud image without any differencing. The bottom image shows the cloud after point differencing. This allows for a cloud with better key feature representation.

3.3 Registration

The method used for the registration process was the iterative closest point method. Though this method is the oldest approach for registration of multi-dimensional data, it is still considered to be one of the dominant methods used for processing 3D point cloud datasets. The original ICP method, first presented by Paul J. Besl and Neil D. McKay [7] as a method for registering several different types of 3D data including triangle meshes, polylines and point sets. As the name would suggest, this initial design for registration simply computed the nearest neighbor of each point within the sample cloud and then transformed the sample cloud appropriately such that the distance between the nearest neighbors was reduced for the majority of the points.

This method was then iteratively repeated until either a maximum number of iterations had been completed or an acceptable RMS error between the sample point set and the fixed-point set had been reached. Though this showed a vast improvement in speed over other nonlinear optimization methods at the time, the algorithm was tested on rather small point sets or polyline meshes. Due to the large scale of aerial LiDAR point sets, performing a one-to-one distance comparison of each point would be impractical and would require too much time. Another version of ICP was developed by Michael Greenspan and Mike Yurick [18] to utilize the KD-tree structure to greatly improve the speed at which the ICP is performed. First developed by Jon Louis Bently [26], the KD-tree is a data structure that allows for the spatial partitioning of data sets with k-dimensions. This structure was later used in many nearest neighbor searching algorithms to improve the time needed to find nearest neighbors within large multidimensional datasets. The ICP method utilize RMS error between matched points as a metric to

determine accuracy of the registration. The RMS error between *n* number of points can be calculate with

$$error = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{i1} - x_{i2})^2 + (y_{i1} - y_{i2})^2 + (z_{i1} - z_{i2})^2} \quad (5)$$

Due to the increased speed generated by the KD-tree approach, the KD-tree based ICP is used for testing ICP registration. Figure 11 below shows the block diagram for the entire edge differencing registration function with the KD-tree ICP.

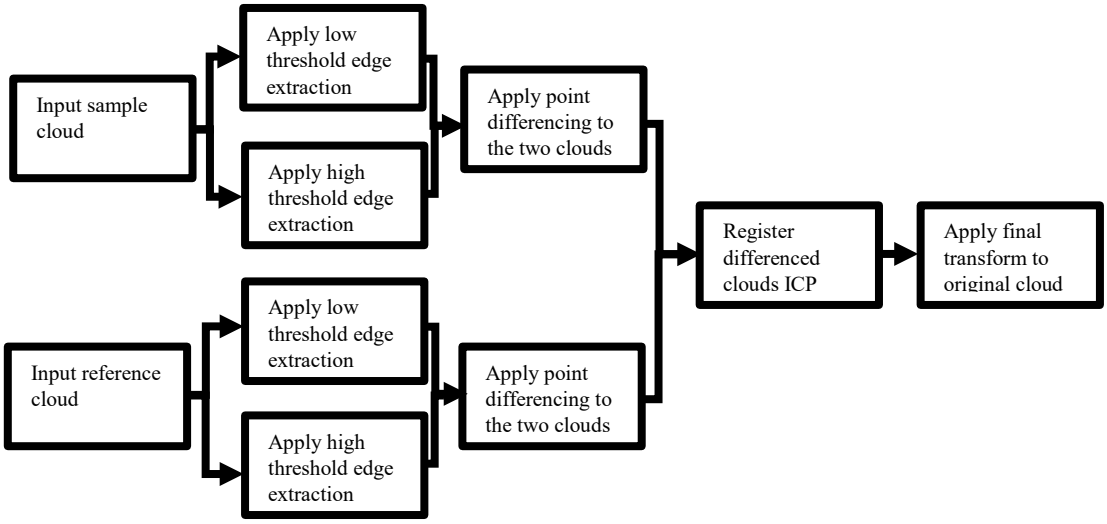

Figure 11: Block diagram for edge differencing network.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Dataset

For proper evaluation of the feature extraction registration approaches, a large-scale aerial-based point set would be needed. The Surrey Aerial LiDAR Dataset was chosen for use in the testing of the registration. This dataset contains multiple captures of the city of Surrey, British Columbia. The dataset contains 3D point cloud tiles that each represent a 500 meter by 500-meter ground area and include both urban and suburban areas. The dataset includes captures from 3 different scan periods, one in 2009, 2013 and 2018. Multiple flyovers of each tile were performed to ensure high quality point representations from each angle. The tiles from each year range in resolution with the tiles from 2009 being represented by roughly 600,000 points per tile, the 2013 being represented by 6 million points and the 2018 tiles represented by 11 million points per tile. Though there are several other large-scale point cloud datasets available, many, such as ScanNet [27] and Sydney Urban Objects [28], are captured using a ground based or mobile sensor. Those that are captured using aerial LiDAR, like ISPRS [29], are of considerably lower resolution than the Surrey dataset and would produce results inconsistent with more modern, high resolution datasets. Furthermore, having matching cloud captures from 2009, 2013 and 2018 will allow for testing of the registration approach at varying resolutions to ensure the approach maintains integrity over varying resolutions. Figure 12 below shows the sample cloud captures from various years.

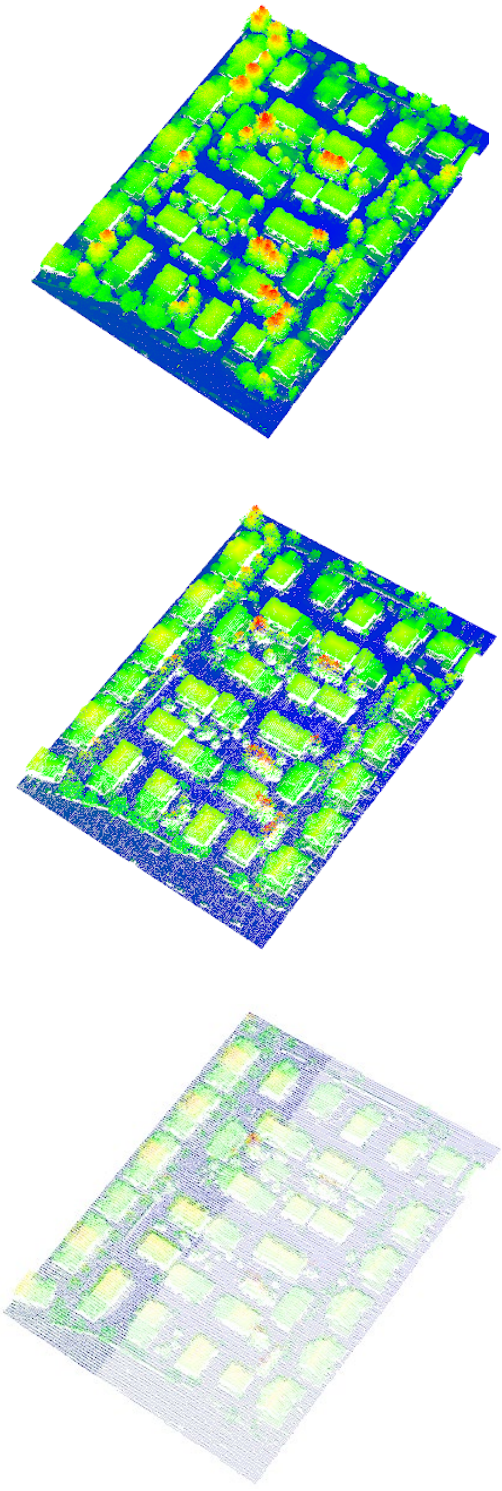

Figure 12: Sample cloud captures of the Surrey LiDAR data from varying years and resolutions. The top cloud is from the 2018 capture and is represented by 1.6 million points. The middle cloud is from the 2013 capture and is represented by 825,000 points. The bottom cloud is from the 2009 capture and is represented by 100,000 points.

4.2 Performance Evaluation

The tile selected for evaluation of the method had a combination of urban scenery, such as large buildings defined by sharp edges, and suburban scenery consisting of housing developments and small vehicles with less defined edges. The first test performed aimed to determine the effectiveness of the edge extraction when performing ICP registration on two entirely overlapping tiles. The sample tile points were first randomly shifted using a gaussian jitter function to ensure that the two tile points were not identical. The sample tiles were then all shifted and rotated so as to not match the reference cloud. All clouds were translated using a standard transform matrix to ensure testing integrity. The tiles were then registered without any feature extraction for 20, 100 and 400 iterations. The final registration score was calculated using an RMS calculation of 50,000 randomly selected point pairs. The same procedure was then performed on the clouds after the edge extraction was performed on both clouds. Finally, the edge extraction differencing was performed and tested on the clouds. These tests were performed on an identical area for each year of data to test the integrity of the registration method given different resolution data. Table 1 below shows the results of the registration tests for the entirely overlapping tiles.

Table 1: Results from fully overlapping cloud registration

Test Type	Cloud Year	Points per Cloud after Preprocessing	RMS after 20 Iterations	RMS after 100 Iterations	RMS after 400 Iterations
Full cloud with no extraction [ref]	2009	691,106	6.7139	1.94972	1.43621
	2013	6,362581	7.13262	2.09883	1.4794
	2018	10,983,837	6.75681	1.64969	1.62538
Full Cloud with Edge Extraction (proposed)	2009	189,893	7.95535	0.935654	0.932117
	2013	1,425,332	7.79627	1.38288	1.26283
	2018	2,729,850	7.93405	2.85238	1.29862
Full Cloud with Edge Differencing (proposed)	2009	118,846	8.1466	.664902	0.65007
	2013	715,617	8.11837	1.00897	1.02132
	2018	1,086,119	7.08847	1.08254	1.10016

Figures 13, 14 and 15 below shows the results of the registration on the 2013 cloud tiles. Figure 16 shows the results of applying the edge differencing registration transform to the full cloud.

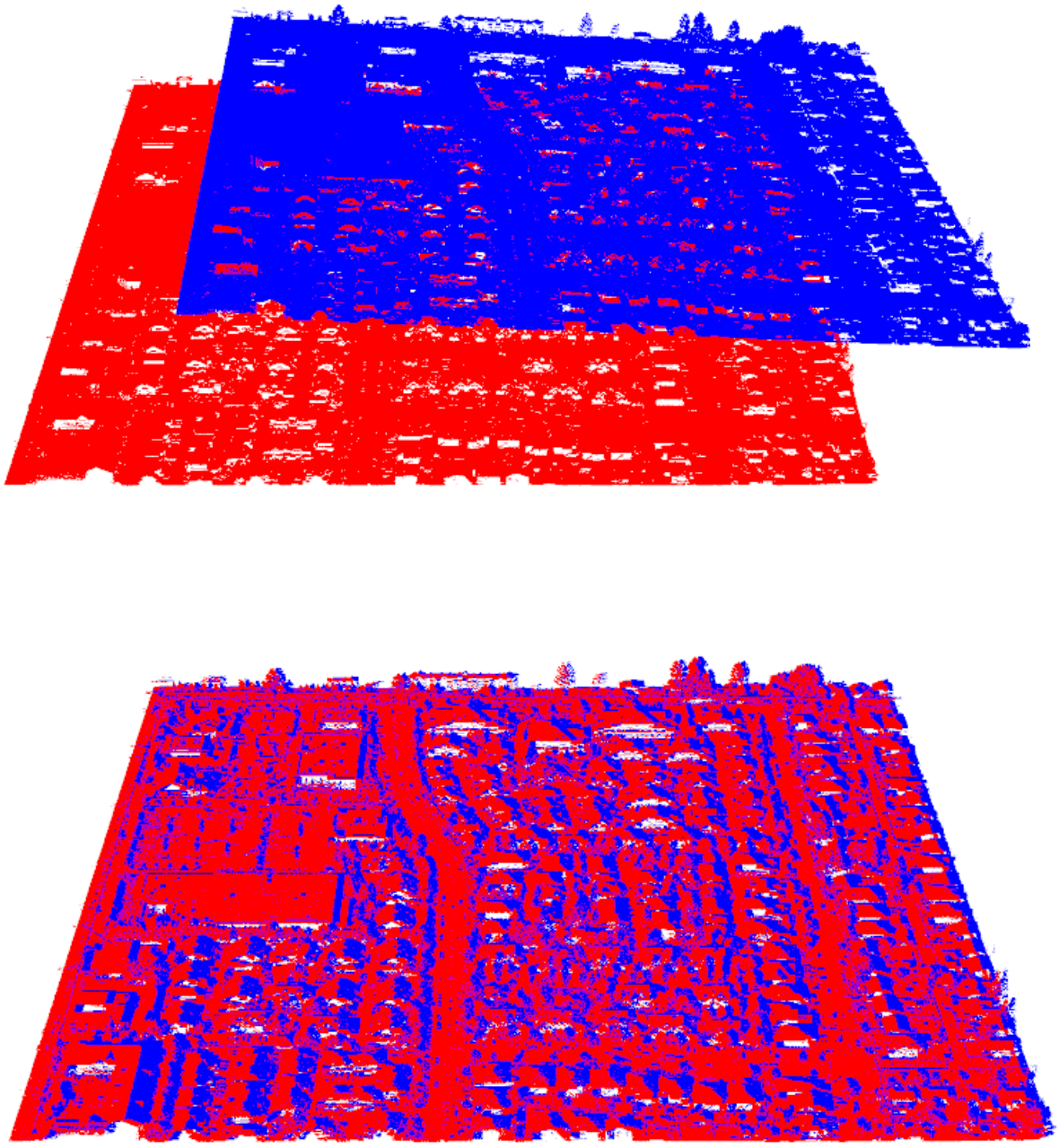

Figure 13: Results of the fully overlapping clouds from the 2013 tile after 100 iterations without any key feature preprocessing. The top image shows the reference cloud (red) and the sample cloud (blue) prior to registration. The bottom picture shows the two clouds after 100 iterations of ICP registration.

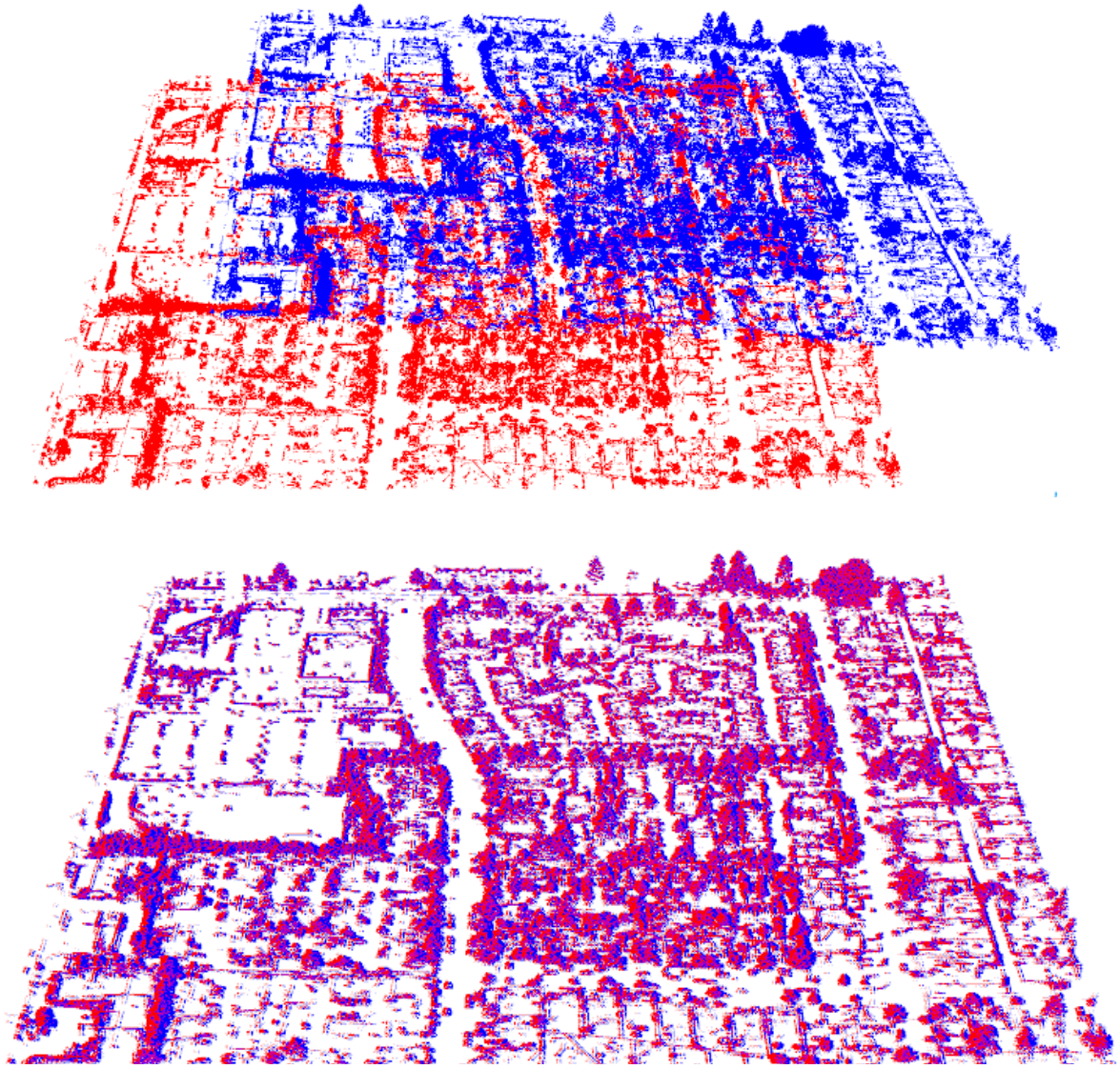

Figure 14: Results of the fully overlapping clouds from the 2013 tile after 100 iterations with edge extraction. The top image shows the reference cloud (red) and the sample cloud (blue) prior to registration. The bottom picture shows the two clouds after 100 iterations of ICP registration.

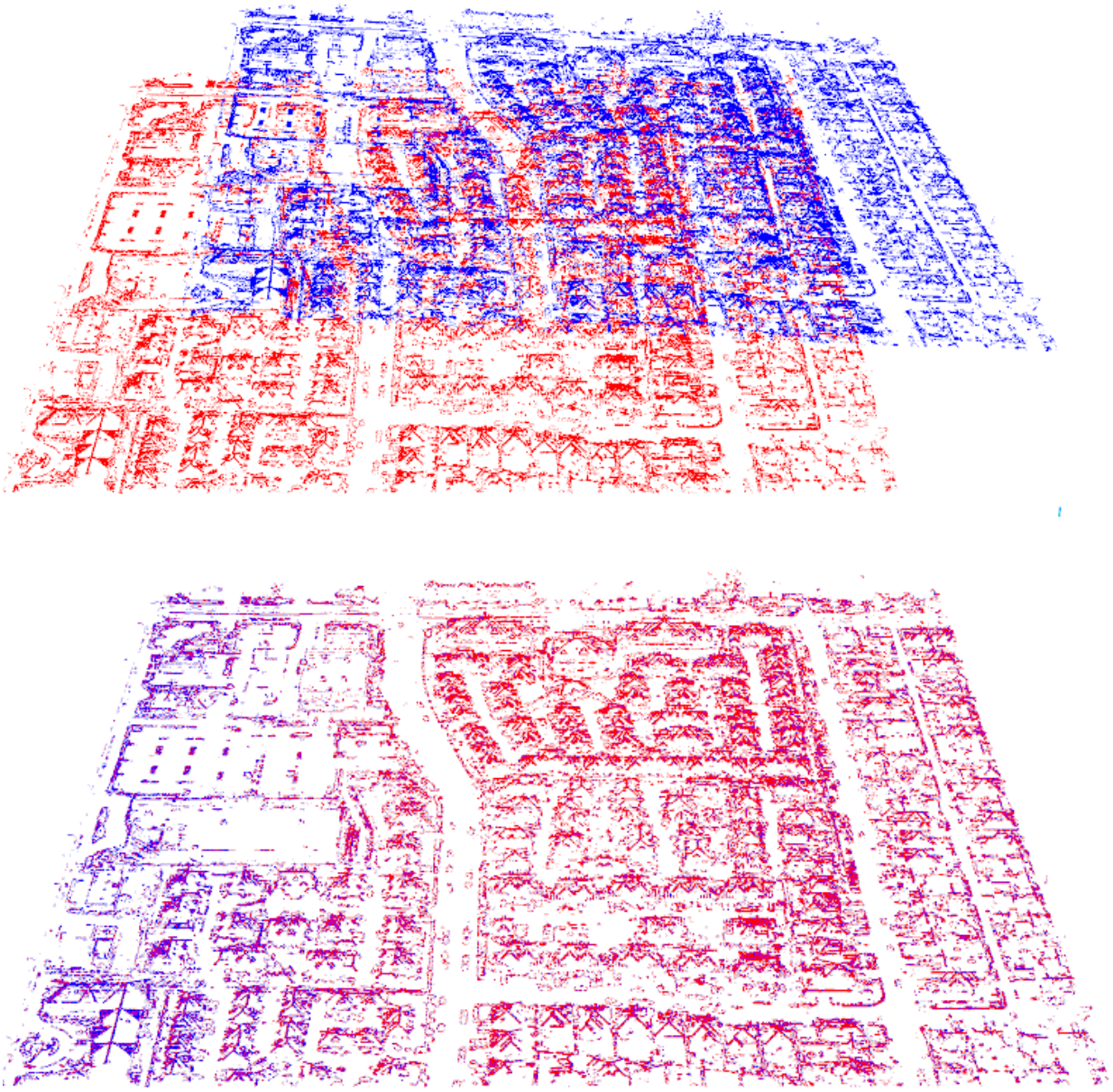

Figure 15: Results of the fully overlapping clouds from the 2013 tile after 100 iterations with edge extraction and edge differencing. The top image shows the reference cloud (red) and the sample cloud (blue) prior to registration. The bottom picture shows the two clouds after 100 iterations of ICP registration.

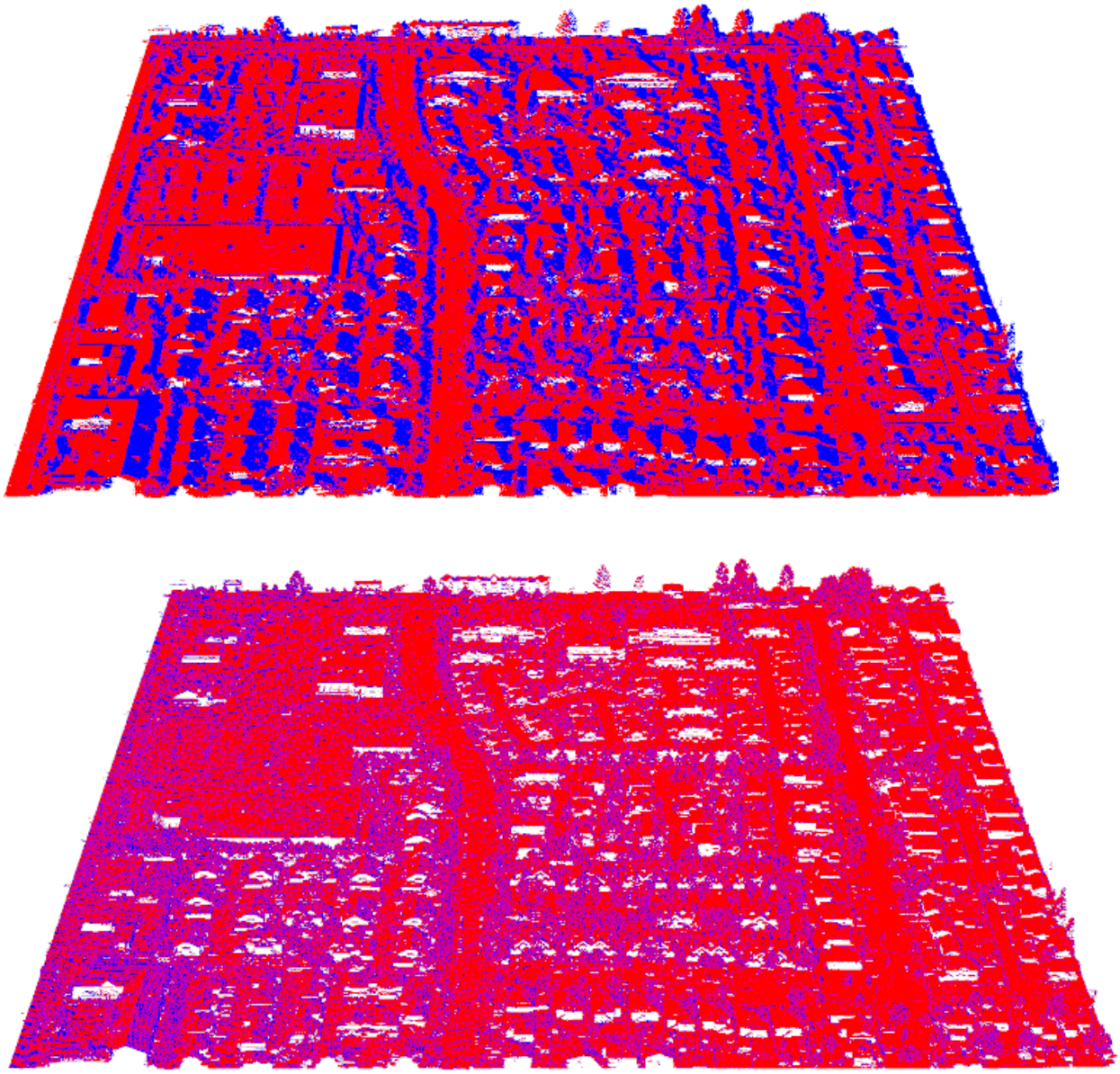

Figure 16: Final transformation of 2013 fully overlapping cloud data after 100 registration iterations. The top image shows the final transformation of the reference cloud (red) and the sample cloud (blue) after 100 iterations of registration without any key feature preprocessing. The bottom picture shows the final transformation of the point differencing registration seen in Figure 7 when applied to the original full cloud.

The test was then performed on tiles that only consisted of a 50% overlap to test the integrity of the feature extraction and registration approaches when the sample and reference point clouds were not taken over identical areas. This test was also performed first without any feature extraction, followed by edge extraction and finally edge differencing. Table 2 below shows the results for the partially overlapping tile registrations.

Table 2: Results from partially overlapping cloud registration

Test Type	Cloud Year	Points per Cloud	RMS after 20 Iterations	RMS after 100 Iterations	RMS after 400 Iterations
Overlapping clouds with no extraction	2009	691,106	3.06619	1.45635	1.10594
	2013	6,362581	3.07143	1.6094	1.35133
	2018	10,983,837	3.11554	2.42616	1.47044
Overlapping clouds with Edge Extraction	2009	189,893	3.77475	.519038	.51936
	2013	1,425,332	3.51934	2.46088	.841948
	2018	2,729,850	3.7002	2.96091	2.73174
Overlapping clouds with Edge Differencing	2009	118,846	4.03427	0.25937	0.25559
	2013	715,617	3.60644	.79413	.761031
	2018	1,086,119	3.59429	1.37628	.534534

Figures 17, 18 and 19 show the results of performing registration for 100 iterations on the 2013 overlapping tile data without edge extraction, with edge extraction and with edge differencing. Figure 20 shows the results of applying the final transform from the edge differencing registration on the entire cloud.

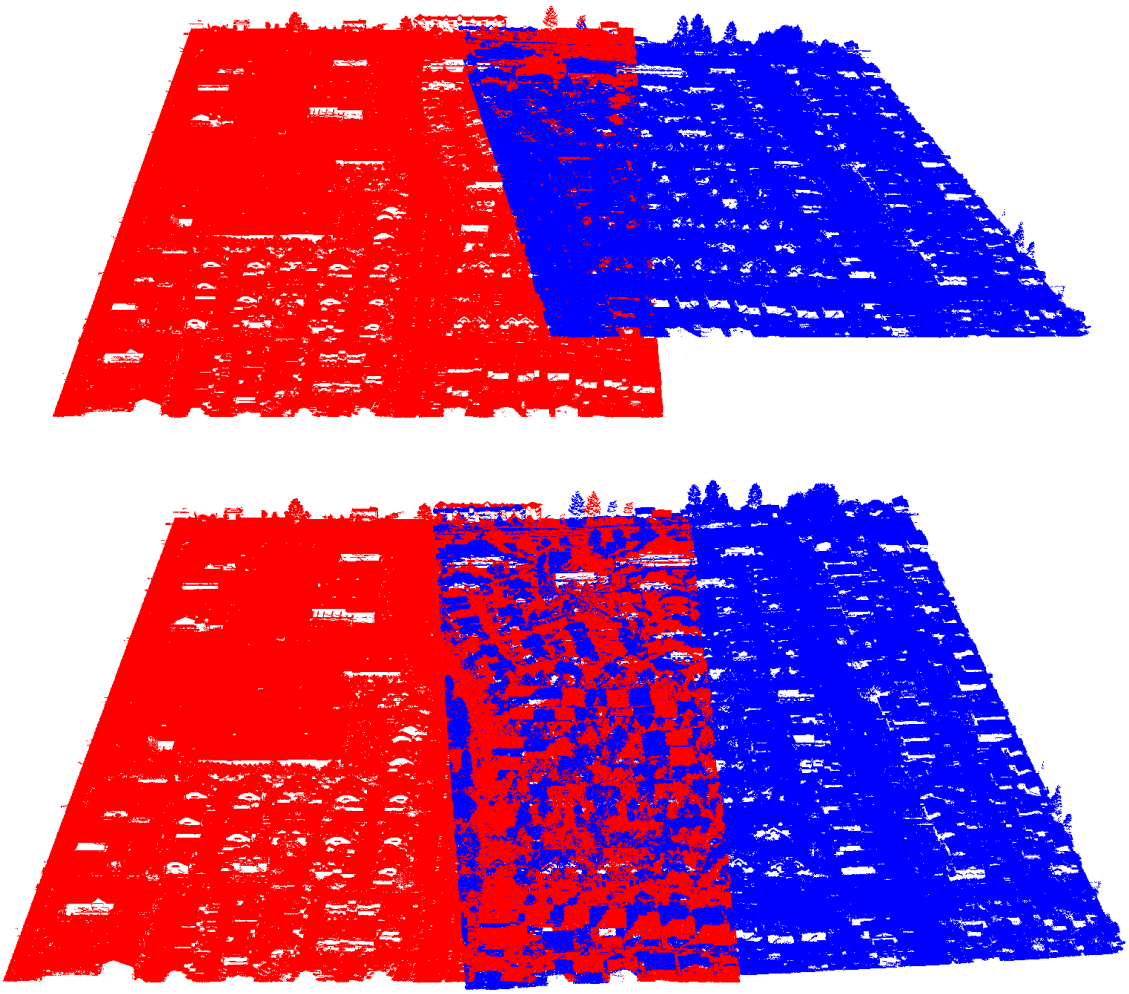

Figure 17: Results of the partially overlapping clouds from the 2013 tile after 100 iterations without any key feature preprocessing. The top image shows the reference cloud (red) and the sample cloud (blue) prior to registration. The bottom picture shows the two clouds after 100 iterations of ICP registration.

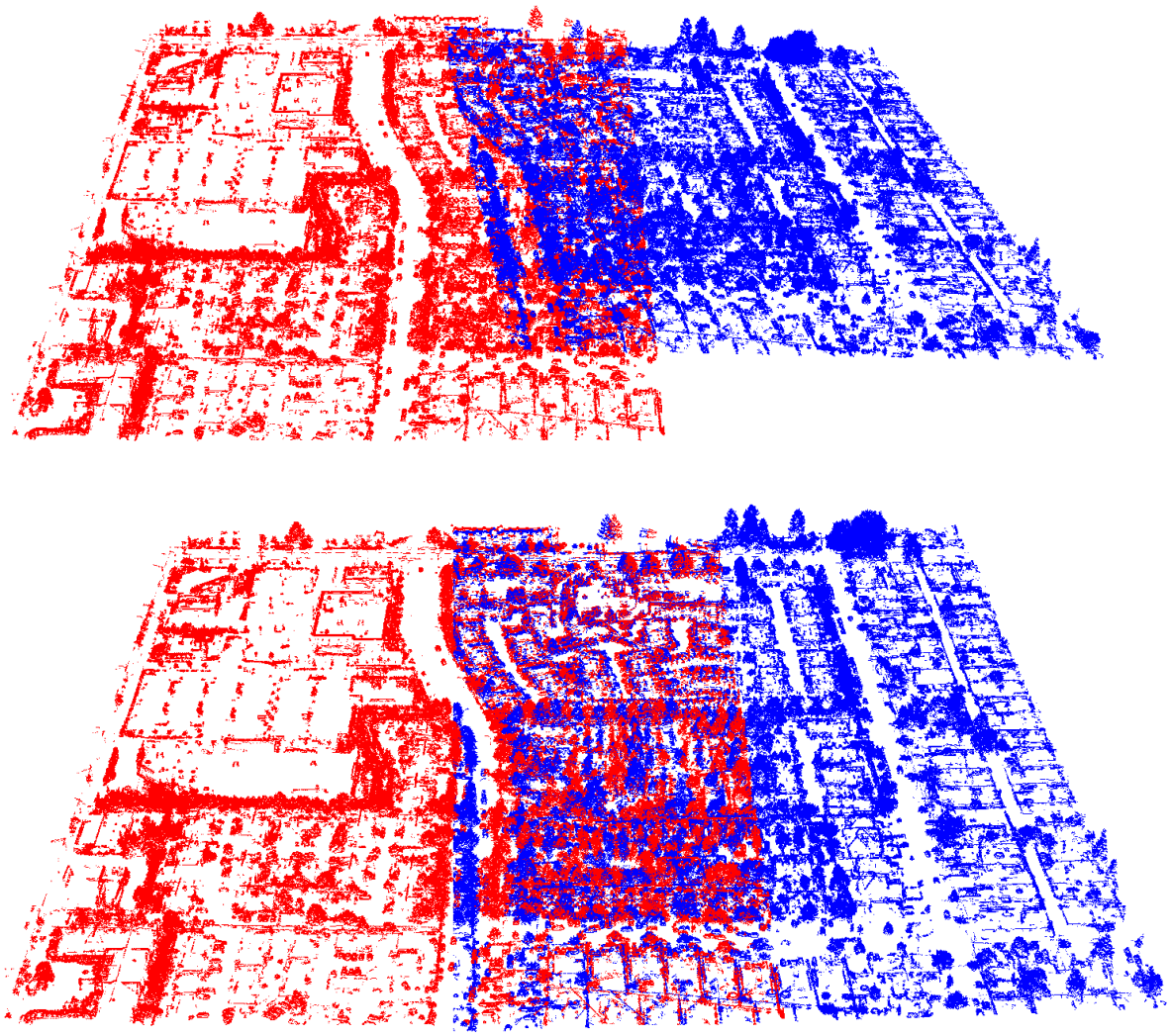

Figure 18: Results of the partially overlapping clouds from the 2013 tile after 100 iterations with edge extraction. The top image shows the reference cloud (red) and the sample cloud (blue) prior to registration. The bottom picture shows the two clouds after 100 iterations of ICP registration.

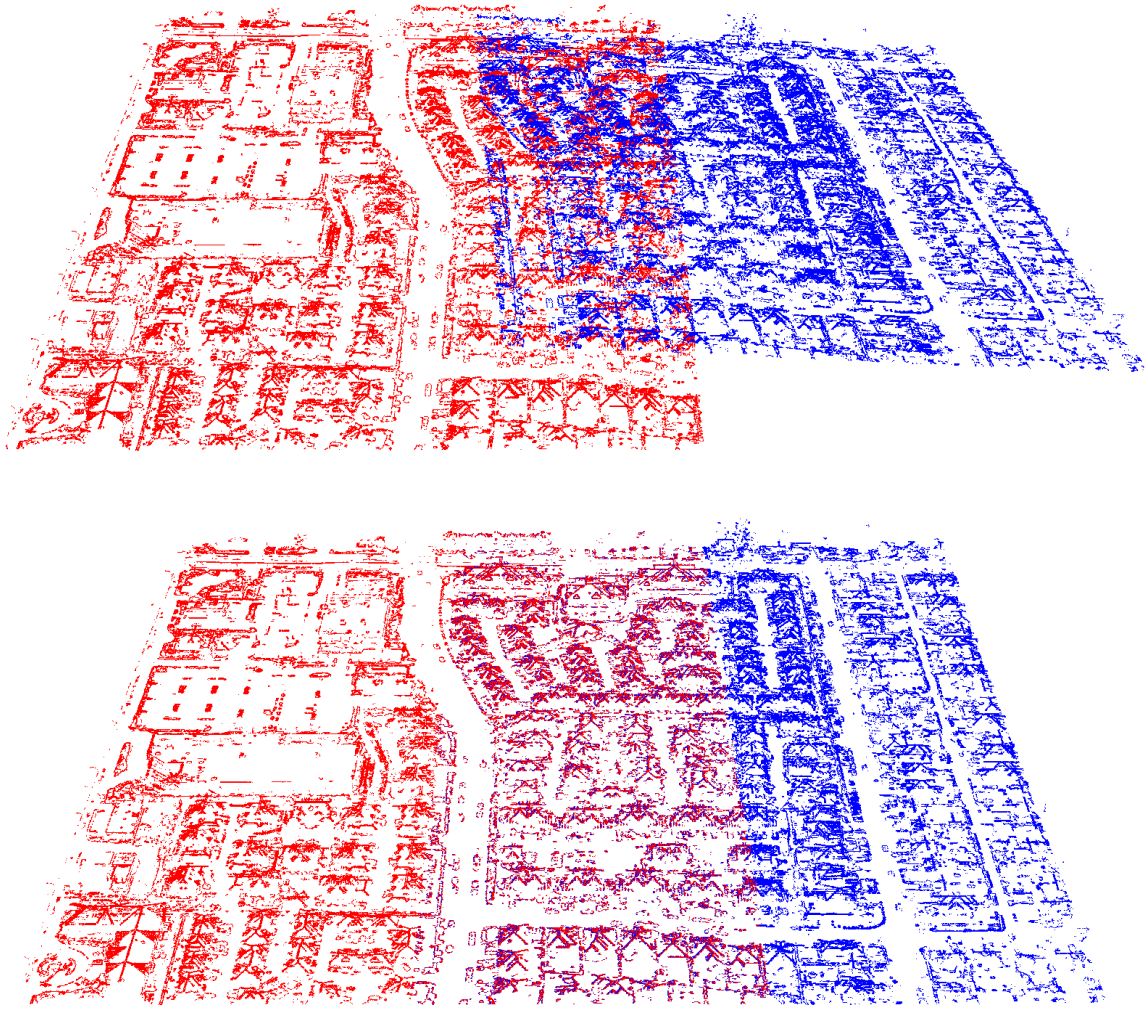

Figure 19: Results of the partially overlapping clouds from the 2013 tile after 100 iterations with edge extraction and edge differencing. The top image shows the reference cloud (red) and the sample cloud (blue) prior to registration. The bottom picture shows the two clouds after 100 iterations of ICP registration.

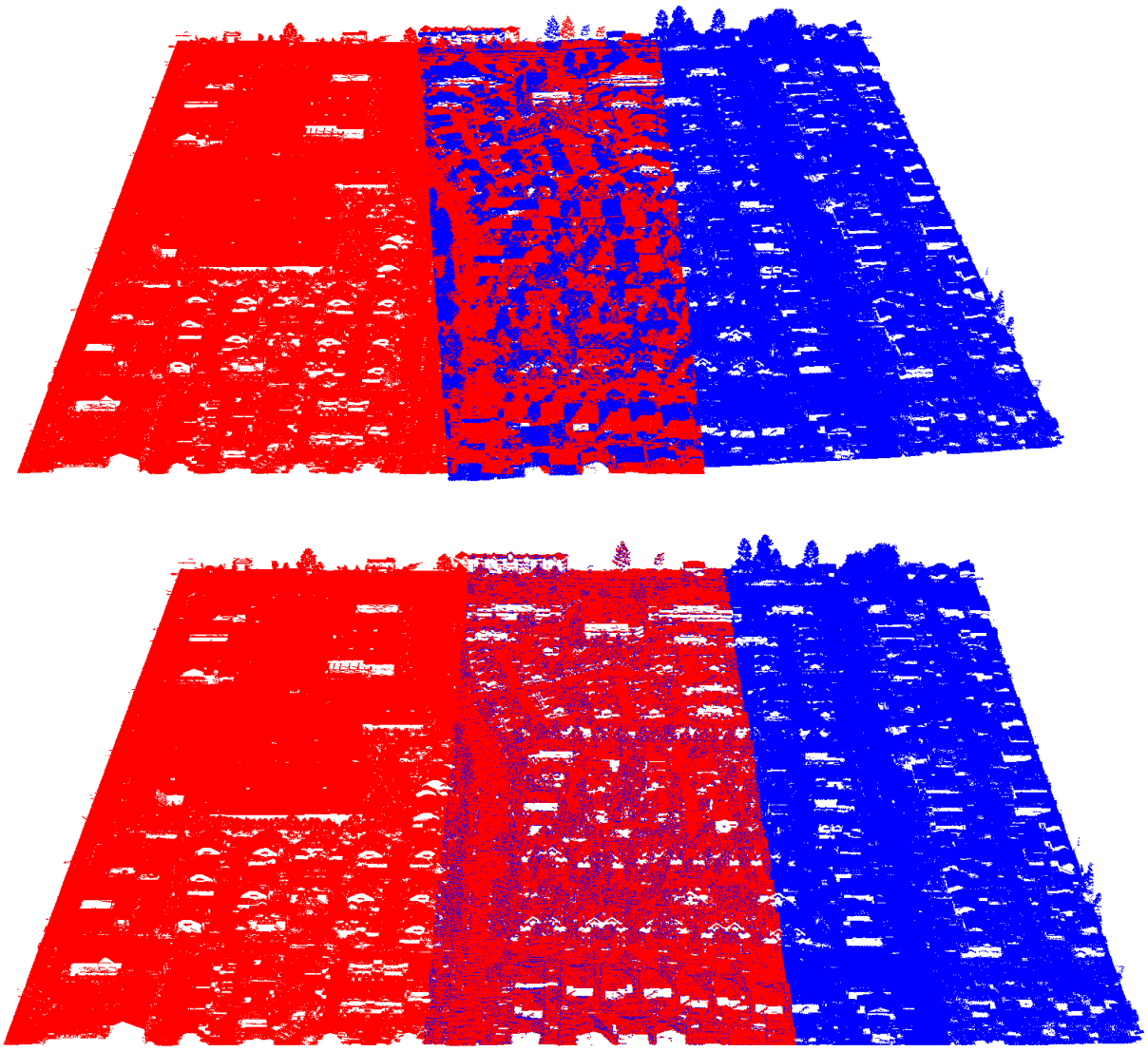

Figure 20: Final transformation of 2013 partially overlapping cloud data after 100 registration iterations. The top image shows the final transformation of the reference cloud (red) and the sample cloud (blue) after 100 iterations of registration without any key feature preprocessing. The bottom picture shows the final transformation of the point differencing registration seen in Figure 7 when applied to the original full cloud.

4.3 Discussion

The goal of the feature extraction registration method was to improve the pairwise fine registration method for large scale aerial LiDAR point clouds without the introduction of pairwise coarse registration or deep learning methods for registration. For the fully overlapping cloud registration there is a clear improvement to the registration accuracy with the edge extraction and differencing methods implemented. Table 1 shows a clear improvement after 100 iterations for both the low, medium and high-quality LiDAR clouds. The RMS error was reduced to .0935654 after performing registration with edge extraction when compared to the original error of 1.94972 without any feature extraction. This error was reduced by 3 times when performing the edge differencing to a final error of only .664902. These results hold true for 400 iterations as well with a reduction from 1.43621 of the original cloud to .932117 with feature extraction registration and .65007 with edge differencing. These results hold true for all resolutions of data, though the error reduction is slightly less as the quality of data increases. For the medium quality data, a reduction from 2.09883 to 1.38288 can be seen when performing edge extraction registration, which is further reduced to 1.08254 using edge differencing after 100 iterations. The high-quality data saw an increase in RMS error after 100 iterations for the edge extraction versus the original cloud registration. However, this outlier was corrected after performing edge differencing, reducing RMS from 1.64949 to 1.08254 after 100 iterations. After 400 iterations, the medium and high-quality RMS errors are consistent with the results from the low-quality registration tests, showing a consistent decrease in error after performing edge extraction and edge differencing. For all qualities of data after 100 and 400 iterations, the lowest error achieved by the

registration was achieved using the edge differencing approach, proving the success of the registration method. Something to note is that early iterations tend to show less error for the full cloud as opposed to higher iteration tests. This is most likely due to the ICP registration matching the ground plane, which is a uniform plane that can provide many early local minima and a low RMS error. However, this is a false sense of success as after aligning the planes the registration method has a hard time matching more non-uniform points such as buildings and cars due to the large number of matches represented by the ground plan. Due to the remove of the ground plane in the edge extraction and differencing methods, these earlier iterations show a larger error, but this is quickly corrected given enough iterations. It should also be noted that the RMS error after 20 iterations achieved the lowest error of 6.7139, which is far too large to consider successful for a registration approach. In contrast, the lowest RMS error achieved was the .65007 on the low quality 400 iteration edge differencing approach, further showing the best registration is achieved through edge differencing. Figure 16 shows the clear success of the edge differencing method when the final transform for that registration is applied to the original cloud. In this figure a clear difference between the sample and reference cloud can be seen in the top image for registration performed without any edge extraction but there is very little difference in the bottom image where the edge differencing transform was applied.

The partially overlapping registration tests follow the trend seen in the fully overlapping tests. For the lower iteration tests the test with no edge extraction performed with the lowest RMS error again due to plane alignment and the large number of points representing the ground plane. However, once higher iterations were reached, a clear

advantage was seen using edge extraction and edge differencing for registration. For the 2009 low resolution data, there is a 6 times improvement from 1.45635 to .25937 error when utilizing the edge differencing method after 100 iterations. This holds true for 400 iterations, were more than a 4 times reduction can be seen. For the medium and high-quality data, the error is consistently halved with the implementation of the point differencing method, with the high-quality data seeing an almost 3 times reduction in RMS error after 400 iterations when using the edge differencing method. Something to note is the edge extraction on the high-quality data performed worse in the partially overlapping tests. This can be accounted for by the point clusters present in the non-differenced edge extraction from natural features. These clusters would cause more local minima and may make it harder to properly orient and shift the cloud, a task that is far more important in partial registration than it is in full overlap registration. However, this error was resolved with the point differencing method that removed these clustered points. In Figure 20 a clearly superior registration can be seen in the bottom image were edge differencing was used as opposed to the top image, where no edge extraction was used. In the top image, the clouds are not even properly oriented, let alone properly finely registered. In contrast, the bottom image shows a final result where it is hard to distinguish the sample cloud from the reference cloud in the overlapping section, similar to the results in Figure 16.

CHAPTER 5

CONCLUSION

This paper aimed to introduce a method for feature-based registration of large-scale aerial LiDAR point clouds in order to alleviate the stress of performing fine pairwise registration on large point sets and to remove the need for coarse pairwise registration. A feature extraction method for finding key edge points based on 3D eigenvector values was utilized to create a key feature point set to represent the original cloud. This feature point set was then registered using a fine pairwise registration method based on the iterative closest point registration for point set data. This method of key feature representation allowed for the registration of large-scale point clouds to be performed accurately using less points to represent the cloud, effectively reducing computational overhead and time needed to perform proper fine registration. The results from the fully overlapping point cloud registration tests, seen in table 1, show that our method for key point extraction allows for the point cloud data to be represented by a key feature cloud that can still maintain the integrity of the original cloud for registration purposes but that greatly reduces the number of points representing the cloud, easing the load on the fine pairwise registration. This reduction in points allows for a significant decrease in both hardware overhead and time for registration iteration. Furthermore, the key feature clouds showed a significant decrease in RMS error between sample and reference cloud when compared to the original cloud registration. Our method of point differencing to remove large clusters of low-quality points further improved the results for the registration. This shows that our method of feature extraction and edge

differencing not only successfully decrease the amount points needed to successfully register the cloud, which in turn decreases hardware and time costs, but that it also increases the overall performance of the fine pairwise registration method. These improvements can be seen throughout the different time periods for the aerial captures, illustrating our method's ability to perform regardless of resolution. This method for registration showed similar improvements when given partially overlapping clouds as well. This new method for aerial registration can be used to improve cloud registering for large scale point clouds, which is often a key function of various 3D computer vision functions such as segmentation and change detection.

BIBLIOGRAPHY

[1] C. Harris and M. Stephens, “A Combined Corner and Edge Detector,” *In Proc. of the Fourth Alvey Vision Conference,* Manchester, UK, 1988, pp. 147-151

[2] I. Sobel and G. Feldman, “A 3x3 Isotropic Gradient Operator for Image Processing,” *Pattern Classification and Scene Analysis, John Wiley and Sons,* Menlo Park, CA, 1973, pp.271-272

[3] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 2015, pp. 922-928

[4] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 2015, pp. 681-687

[5] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Las Vegas, NV, USA, 2003, pp. 2743-2748

[6] N. Varney, V. K. Asari and Q. Graehling, “DALES: A Large-scale Aerial LiDAR Data Set for Semantic Segmentation,” *Computer Visio and Pattern Recognition (CVPR) Workshops,* 2020, pp. 717-726

[7] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992, vol. 14, no. 2, pp. 239-256

[8] S. Gold, A. Rangarajan, C. P. Lu, S. Pappu and E. Mjolsness, “New Algorithms for 2D and 3D Point Matching: Pose Estimation and Correspondence,” *Pattern Recognition Society: Pattern Recognition*, 1998, vol. 31, no. 8, pp. 1019-1031

[9] R. B. Rusu, N. Blodow and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," *2009 IEEE International Conference on Robotics and Automation*, Kobe, 2009, pp. 3212-3217

[10] M. Mohamad, D. Rappaport and M. Greenspan, "Generalized 4-Points Congruent Sets for 3D Registration," *2014 2nd International Conference on 3D Vision*, Tokyo, 2014, pp. 83-90

[11] D. G. Lowe, "Object recognition from local scale-invariant features," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, 1999, pp. 1150-1157

[12] H. Yu, S. Keshavamurthy, H. Bai, S. Sheorey, H. Nguyen and C. N. Taylor, "Uncertainty estimation for random sample consensus," *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, Singapore, 2014, pp. 395-400

[13] Xue Yuan, Jianming Lu and Takashi Yahagi, "A method of 3D face recognition based on principal component analysis algorithm," *2005 IEEE International Symposium on Circuits and Systems*, Kobe, 2005, pp. 3211-3214

[14] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao and T. Funkhouser, "3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 199-208

[15] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 77-85

[16] H. Deng, T. Birdal and S. Ilic, "PPFNet: Global Context Aware Local Features for Robust 3D Point Matching," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 195-205

[17] Y. Wang and J. Solomon, "Deep Closest Point: Learning Representations for Point Cloud Registration," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 3522-3531

[18] M. Greenspan and M. Yurick, "Approximate k-d tree search for efficient ICP," *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, Banff, Alta., 2003, pp. 442-448

[19] M. Vlaminck, H. Luong and W. Philips, "Multi-resolution ICP for the efficient registration of point clouds based on octrees," *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, Nagoya, 2017, pp. 334-337

[20] H. Moravec, “Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover,” *Tech. Report, CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University & doctoral dissertation,* Pittsburgh, PA, 1980

[21] C. Weber, S. Hahmann and H. Hagen, "Sharp feature detection in point clouds," *2010 Shape Modeling International Conference*, Aix-en-Provence, 2010, pp. 175-186

[22] D. Bazazian, J. R. Casas and J. Ruiz-Hidalgo, "Fast and Robust Edge Extraction in Unorganized Point Clouds," *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Adelaide, SA, 2015, pp. 1-8

[23] Y. Lian, T. Feng and J. Zhou, "A Dense Pointnet++ Architecture for 3D Point Cloud Semantic Segmentation," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, 2019, pp. 5061-5064

[24] H. Thomas, C. R. Qi, J. Deschaud, B. Marcotegui, F. Goulette and L. Guibas, "KPConv: Flexible and Deformable Convolution for Point Clouds," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 6410-6419

[25] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 4490-4499

[26] J. L. Bentley, "Multidimensional Binary Search Trees in Database Applications," in *IEEE Transactions on Software Engineering*, vol. SE-5, no. 4, pp. 333-340, July 1979

[27] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser and M. Nießner, "ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 2432-2443

[28] A. Quadros, J. Underwood and B. Douillard, “Sydney Urban Objects Dataset,” *Australian Centre for Field Robotics, Univertsity of Sydney*, Sydney, Australia, 2013

[29] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf, “The ISPRS benchmark on urban object classificationand 3D building reconstruction,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2013, pp. 293–298