2009

Unsupervised Building Detection From Irregularly Spaced Lidar And Aerial Imagery

Nicholas Shorter
University of Central Florida

UNSUPERVISED BUILDING DETECTION FROM IRREGULARLY SPACED LIDAR AND AERIAL IMAGERY

by

NICHOLAS SVEN SHORTER
B.S.E.E. University of Central Florida, 2005
M.S.E.E. University of Central Florida, 2006

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy of Science
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2009

Major Professor: Takis Kasparis

ABSTRACT

As more data sources containing 3-D information are becoming available, an increased interest in 3-D imaging has emerged. Among these is the 3-D reconstruction of buildings and other man-made structures. A necessary preprocessing step is the detection and isolation of individual buildings that subsequently can be reconstructed in 3-D using various methodologies. Applications for both building detection and reconstruction have commercial use for urban planning, network planning for mobile communication (cell phone tower placement), spatial analysis of air pollution and noise nuisances, microclimate investigations, geographical information systems, security services and change detection from areas affected by natural disasters. Building detection and reconstruction are also used in the military for automatic target recognition and in entertainment for virtual tourism.

Previously proposed building detection and reconstruction algorithms solely utilized aerial imagery. With the advent of Light Detection and Ranging (LiDAR) systems providing elevation data, current algorithms explore using captured LiDAR data as an additional feasible source of information. Additional sources of information can lead to automating techniques (alleviating their need for manual user intervention) as well as increasing their capabilities and accuracy. Several building detection approaches surveyed in the open literature have fundamental weaknesses that hinder their use; such as requiring multiple data sets from different sensors, mandating certain operations to be carried out manually, and limited functionality to only being able to detect certain types of buildings.

In this work, a building detection system is proposed and implemented which strives to overcome the limitations seen in existing techniques. The developed framework is flexible in that

it can perform building detection from just LiDAR data (first or last return), or just nadir, color aerial imagery. If data from both LiDAR and aerial imagery are available, then the algorithm will use them both for improved accuracy. Additionally, the proposed approach does not employ severely limiting assumptions thus enabling the end user to apply the approach to a wider variety of different building types. The proposed approach is extensively tested using real data sets and it is also compared with other existing techniques. Experimental results are presented.

This work is dedicated to my sister

Nichole April Shorter

ACKNOWLEDGMENTS

I would like thank my PhD Committee Dr. Takis Kasparis, Dr. Georgios Anagnostopoulos, Dr. Michael Georgiopoulos, Dr. Andy Lee, Dr. Linwood Jones and Dr. Wasfy Mikhael. The valuable feedback the committee provided me with during my PhD Candidacy Exam, my dissertation defense and at several other critical stages in my doctoral research greatly enhanced the quality of my dissertation. I thank Dr. Takis Kasparis for mentoring me for years on end during my graduate studies. His experience and knowledge in Digital Image and Signal Processing was a great asset to me for my research. I thank Dr. Georgiopoulos and Dr. Anagnostopoulos for their tutelage in unsupervised learning. I thank Dr. Mikhael for his instruction on a plethora of Digital Signal Processing related concepts, and for also sharing with me practices for modeling and accounting for various phenomena when developing a proposed system. I am thankful to Dr. Lee and Dr. Mahalanobis for providing me with insight on industrial perspectives on LiDAR and 3D Reconstruction. I also thank Dr. Lee for informing me about various LiDAR programs and data sets.

I thank my immediate family, Nichole, Kathy and Sven Shorter for all of the motivation and encouragement they provided me with on a daily basis. Their compassion was of great help to me during difficult times of struggle. I thank my girlfriend Amanda McNally for her unconditional love she has blessed me with and her unending support.

I am very grateful to all of the researchers and companies whom have donated data for me to use for my dissertation, data valued at thousands of dollars. I thank Dr. Simone Clode and Dr. Franz Rottensteiner for giving me access to the LIDAR data set known as 'the Fairfield data set', which was collected and donated by AAMHatch. I thank Mr. John Ellis with AeroMap for providing me with the LiDAR data set known as 'the Anchorage data set'.

I thank Dr. Andy Lee, Mr. Jay Hackett, and Mr. Jack Needham and Harris Corporation for their funding and for their feedback provided at numerous technical meetings at Harris.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

3D	Three Dimensional
BC	Building Connected Set
DEM	Digital Elevation Model
DSM	Digital Surface Model
DTM	Digital Terrain Model
ESOM	Ensemble of Self Organizing Maps
FGC	First Generation Connected Set
FP	False Positive
FN	False Negative
HSV	Hue Saturation Value
GC	Ground Connected Set
GIS	Geographic Information System
GLCM	Grey Level Co-occurance Matrix
GPS	Global Positioning System
HT	Homotopy Tree
INS	Inertial Navigation System
LIDAR	Light Detection and Ranging
NASA	National Aeronautical Space Administration
NAHB	National Association of Home Builders
NBC	Non-Building Connected Set
NDVI	Normalized Difference Vegetation Index

NWC	Non-Wall Connected Set
NWT	Non-Wall Triangle
PHT	Pseudo Homotopy Tree
RGB	Red Green Blue
TP	True Positive
TIN	Triangulated Irregular Network
UAV	Unmanned Aerial Vehicle
VCF	Volume Correlation Filter
WC	Wall Connected Set
WT	Wall Triangle

CHAPTER ONE: INTRODUCTION TO LIDAR AND BUILDING DETECTION

The art of automatically detecting man made buildings from various sources of data has existed for several decades now. Methodologies for solving this problem and even extending its applications have evolved with the advent of new technologies which deliver new and/or improved sources of data. The research presented focuses on automatic building detection via Light Detection and Ranging (LIDAR) data and/or a single nadir aerial image, both of which depict the same scene.

In this introductory chapter, first a brief overview of what LiDAR is and how it is procured is presented, followed by a discussion of some of the various applications of building detection. Then, several of the different types of noise which corrupt LiDAR and some of the limitations of the data are cited. Following the limitations of LiDAR data, the limitations of the use of aerial imagery for building detection are also mentioned. Then the complimenting characteristics of the LiDAR data and the aerial imagery are discussed. Finally, the specific problem in which this area of research addresses is presented.

1.1 LIDAR Overview

LiDAR data procured from a nadir (facing directly down from the sky) perspective is typically done with an aircraft, helicopter or plane, which has a Global Positioning System (GPS), an Inertial Navigation System (INS) and a LiDAR sensor system mounted on it. The GPS returns the longitude and latitude coordinates of the aircraft's current position. The INS tracks the altitude of the aircraft. The LiDAR sensor system is an active remote sensing

instrument which consists of an emitter and receiver. The emitter sends out a pulse of light (electromagnetic radiation) into the atmosphere. The telescope (receiver) measures the intensity of the signal scattered back to the sensor system after that signal has interacted with various constituents in the atmosphere [36], [44]. The time from departure to return is also recorded thus enabling the calculation of the distance of the sensor to the target (or in this case the sampled terrain).

Figure 1: Capturing LIDAR Data

In the case of a building surface, the laser will reflect off of the building's surface and return to the sensor. In the case of tree foliage or vegetation, two possible scenarios arise: the laser beam could pass through the foliage or vegetation and hit the ground, or the laser could interact with the foliage or vegetation. The first pulses to return to the sensor are labeled first return pulses. These pulses consist of laser beams which interacted with the top of foliage and vegetation and building structures. The last pulses to return to the sensors are labeled last return pulses. These pulses consist of laser beams which passed through foliage and/or vegetation and interacted with the ground. These pulses also consist of laser beams which interacted with

building surfaces as well. Based on the time it takes from the emission of the laser from the sensor to the return of the laser beam after it has interacted with a given target, the range from the sensor to the target can be calculated. It is possible to identify vegetation and building edges by examining the differences between first and last return pulses. Trees and building walls are typically greater than 3 meters or approximately 9.84 feet in height. Therefore, if a first and last return pulse for a given location has an elevation difference of 3 meters, there is a good chance that location corresponds to a building edge or tree and not the middle of a given surface. The middle of building roof panels or the bottom of the ground will have differences much less than 3 meters. When the laser beam, emitted, from the LIDAR sensor hits a building surface, the majority of the beam is immediately reflected leaving minimal difference between first and last returns.

While some older LIDAR systems only return the longitude, the latitude, and the elevation of a given returned point, newer systems can also capture the sampling time, the intensity of the returned signal, and the first and last return pulses.

1.2 Applications of Building Detection

Applications of building detection have valued use for both militaristic and commercial purposes. For the military, the analysis of LiDAR data can be used for target recognition applications. Work in training volume correlation filters (VCFs) to recognize tanks and other military vehicles within LiDAR data has recently been developed [15]. There is investigation underway for mounting LiDAR sensors on unmanned aerial vehicles (UAVs) [12]. This would enable aerial surveying of terrains in which military forces were denied access too. Scenes

surveyed by an UAV or high flying plane with a LiDAR sensor could then be analyzed for detected buildings.

For commercial uses, the demand for automatic building detection has applications such as urban planning, network planning for mobile communication, spatial analysis of air pollution and noise nuisances, microclimate investigations, geographical information systems, and security services. For entertainment purposes, building detection can be used for tourism information systems. Also, automatic building detection has change detection applications [42], where buildings demolished by natural disasters, are automatically identified via histogram difference thresholding comparing a pre and post event LiDAR analysis.

1.3 LiDAR Advantages, Limitations and Noise Sources

There are several advantages to using LiDAR data for automatic building detection. The LiDAR sensor itself is an active sensor and therefore is not affected by the time of day the data is captured and the lighting conditions of the sampled terrain. Furthermore, because the LiDAR data captures the height information of the terrain, it is possible to extract break lines regardless if they exist on the sun-side or dark-side of the sampled building.

Unfortunately, there is no color information in the captured LiDAR data making vegetation sometimes appearing similar to buildings. Furthermore, the raw, unprocessed data is irregularly distributed. Therefore one must employ some kind of interpolation procedure to identify building edges as straight, continuous lines or to process the LiDAR data with conventional image processing algorithms (which typically assume the data exists at a rasterized grid). The LiDAR data typically has lower resolution (by as much as several orders of

magnitude) when compared to aerial imagery; however with newer sensors this resolution difference is diminishing.

Several phenomena contribute as sources of noise during the various processes involved in procuring the LiDAR data. Furthermore, not only do various sources of noise corrupt the captured data, but also limitations within the sensors themselves further distort the data. It is imperative to understand these sensor limitations and noise sources so that they can intelligently be accounted for.

Simone et. al., for data in which they received covering terrain in Fairfield, Australia, reported in [4] that first and last returns differing in less than 4.6 meters in elevation are not valid. The reason for this was found to be a limitation in the LIDAR sensor itself. The sensor had to reset itself before a second return could be recorded. If a second return comes back to the sensor before the reset time has passed, a dual return was being recorded. Hence if the first and last return were less than 4.6 meters apart, the two returns arrived back at the laser before it could reset itself in time to record the second return, resulting in the LIDAR system simply recording the same return for both first and last return pulses.

There is a plethora of noise sources influencing the processes involved with procuring LiDAR data. Filin [5] reports that because the geo-location of a single point results from the integration of three subsystems (GPS, INS, and LiDAR), it is possible that errors can come from any combination of all three sensors. Specifically, errors resulting from the three subsystems include but are not limited to the following: a constant offset in the range determination, inaccurate scan angle determination, mounting bias from the misalignment between the INS and LiDAR sensor, GPS offset and drift, and INS system drift. These reported errors only exist as systematic discrepancies between the integration and limitations of the sensors themselves. As

the electromagnetic radiation that is the laser is propagated through the atmosphere both its intensity and path are distorted by the interaction between the beam and the atmosphere itself [36], [44]. An additional aspect in which influences the LiDAR procurement process is artifacts resulting from scan angles not uniformly interacting with the sampled terrain due to obstructions from the terrain itself. Consider as an aircraft flies adjacent to a tall skyscraper, the laser pulses will interact with the side of that tall skyscraper but not with the terrain behind it resulting in a shadowing effect [42].

1.4 Aerial Image Limitations

The aerial imagery of a given scene captures the electromagnetic radiance (luminescence) of the scene. This makes it possible to detect shadows, view the colors of the given scene and determine the illumination angle. The resolution of the aerial imagery is typically an order of magnitude greater than LiDAR data (15 to 30cm/pixel resolution and increasing). The data is regularly distributed or rasterized which attracts the use of conventional image processing algorithms for building detection.

The limitations of aerial imagery include the fact cameras are a passive sensor, no height information is directly available and that sometimes the vertical accuracy of a given image is compromised during the ortho-rectification of that image (i.e. tall buildings in the image become warped). Because a camera is a passive sensor, the captured aerial imagery is affected by cloud coverage and lighting conditions. Varying lighting conditions themselves pose challenges to automatic building detection algorithms. Not having the height information available makes it

difficult to distinguish between building and sidewalks, especially when those sidewalks exist adjacent to the building's sun-side.

1.5 Complementary Characteristics of LiDAR and Aerial Imagery

For purposes of improved image classification, it is often advantageous to combine LiDAR data and aerial imagery (provided they have overlapping coverage of the same depicted scene) because of the complementary characteristics the two sources of data possess. However, in order to make use of features which exist in the LiDAR data and then the same features for that corresponding location in the aerial imagery, the two sources of data have to be registered together. The registration of the sources and then the use of their combined features can overcome the two sources' individual weaknesses. For example, with only the height information available and no color information, it is sometimes difficult to disambiguate vegetation from small buildings in the LiDAR data. However, the color available in the aerial imagery can distinguish that vegetation. In a single, nadir, aerial image alone, it is challenging to differentiate buildings from concrete surroundings, especially when the concrete surroundings (such as side walks, parking lots, roads, etc.) exist adjacent to that building's sun-side. However, the height information in the LiDAR data can identify building break-lines regardless of lighting conditions.

1.6 Problem Statement

The building detection problem is being tackled by a plethora of disciplines including image processing, electrical engineering, civil engineering and computer vision. One can think

of the building detection problem as a black box. The problem consists of taking one or tactically fusing multiple correlated forms of data as an input and then the output being pixels/points accurately classified as buildings or other phenomena (such as vegetation and/or ground) from the depicted sources' data. Typically those sources of input range from LiDAR data (from a nadir or ground perspective), aerial photography (nadir, stereo pairs, video sequences), multi-spectral imagery, and GIS plans/databases.

The ideal building detection algorithm would have the following attributes. First and foremost, it is desirable for the building detection algorithm to accurately identify as many buildings as possible. At the same time the algorithm should not label areas as buildings which are in fact not building.

Second, it is desired to have a generalized algorithm that can detect as many different types of buildings as possible. Conversely, it is undesirable to employ the use of overly restricting assumptions which limit the algorithm's applicability to a small subset of buildings. Examples being assuming buildings are rectilinear (have right angle corners with parallel sides [13], [34]), assuming all buildings are parallel to one another ([13]), assuming buildings have flat roof tops ([11]), assuming buildings have only a certain color or subset of colors of roof tops ([20], [25], [34]), assuming all data sets have the same illumination angle, assuming all buildings have only a single color or single texture roof top ([34]), assuming buildings have a maximum size, and so on and so forth. Instead the assumptions employed should be as general as possible. For example it is better to assume a building exterior's shape is a polygon instead of a rectangle. Many buildings have multiple colors on their roof tops and multiple textures. These buildings come in a variety of different shapes, sizes and orientations. Data sets contain different lighting conditions and illumination directions. Unfortunately, generalizing the algorithm to be

applicable to many different types of buildings sometimes comes at the expense of the algorithm's building detection accuracy. Implementing limiting assumptions makes it easier to accurately detect certain subsets of buildings at the expense of excluding the algorithm's applications to other buildings. At the same time, realizing a more general algorithm can sometimes come with the reward of being applicable to all buildings but at the expense of the algorithm's building detection accuracy.

Third, the algorithm should be automated as much as possible to minimize user intervention. Scientists, engineers, and various other researchers develop building detection algorithms for their application to the industry. It would therefore be undesirable for the industry to have to pay a technician for any length of time to help the algorithm at various stages of its execution. Therefore, if the algorithm has parameters, it is obviously better if those parameters do not have to be manually adjusted during the algorithm's execution across different data sets. It is even more desirable if those parameters do not have to be manually adjusted across single data sets to account for building variability within those data sets. If the algorithm makes use of multiple sources of data, it is therefore desirable to have those sources registered automatically. If the algorithm has training phases, the smaller the amount of data the algorithm is required to be trained with, the better. Furthermore, it would be optimal if the features in which the algorithm is trained on did not overly restrict the algorithm's applicability to buildings possessing specific characteristics and do not require the algorithm to be retrained for different data sets.

Fourth, the algorithm should ideally be able to incorporate additional data sources as they become available. For example, the algorithm should start with a fundamental requirement, say raw LiDAR data or aerial imagery with a pre-defined minimum resolution and then be capable of including additional sources of information as they are available to improve the detection

accuracy and/or classification abilities [17]. Note, additional sources of information could be entire data sets from different sensors or simply just a-priori information such as illumination direction, building attributes, etc.

Finally, a fifth ideal attribute is it would be ideal if the algorithm can classify the image beyond simply building and non-building. If instead the algorithm distinguishes the content of the image as building, vegetation, ground, pavement, non-building, etc. This can in turn extend the algorithm's use to a wider range of applications. Of course, it would be ideal to develop a multi-class classifier without sacrificing the algorithm's detection rate.

The system implementation presented in this dissertation strives to make progress towards realizing those aforementioned ideal attributes with the following features. A minimum of either LiDAR data having at least $1.5m^2$ point density or color (red, green blue) aerial imagery having 15 cm pixel resolution is required, both taken from a nadir perspective. If both sources are available and have overlapping coverage, then both are made use of for improved image classification. The only assumptions made about buildings existent in the LiDAR data is that they abruptly protrude as a convex hull from their surrounding terrain with a predefined minimum size. Buildings existent in the aerial imagery are assumed to have a minimum size and exist as one or several convex roof top texture segments. Buildings detected from either source are done so automatically. Automatically meaning there are no training phases, the algorithm does not rely on user input nor parameter adjustment. If both sources (aerial imagery and LiDAR data) are available, then they are automatically registered together and features from both are used to improve the algorithm's classification ability. The algorithm classifies content in the LiDAR data as building, non-building, ground and in the aerial image as building, non-building, and vegetation and from both sources as building, non-building (mostly shrubbery/trees), ground,

and vegetation (grass, weeds). A block diagram depicting the implemented system is shown in Figure 2.

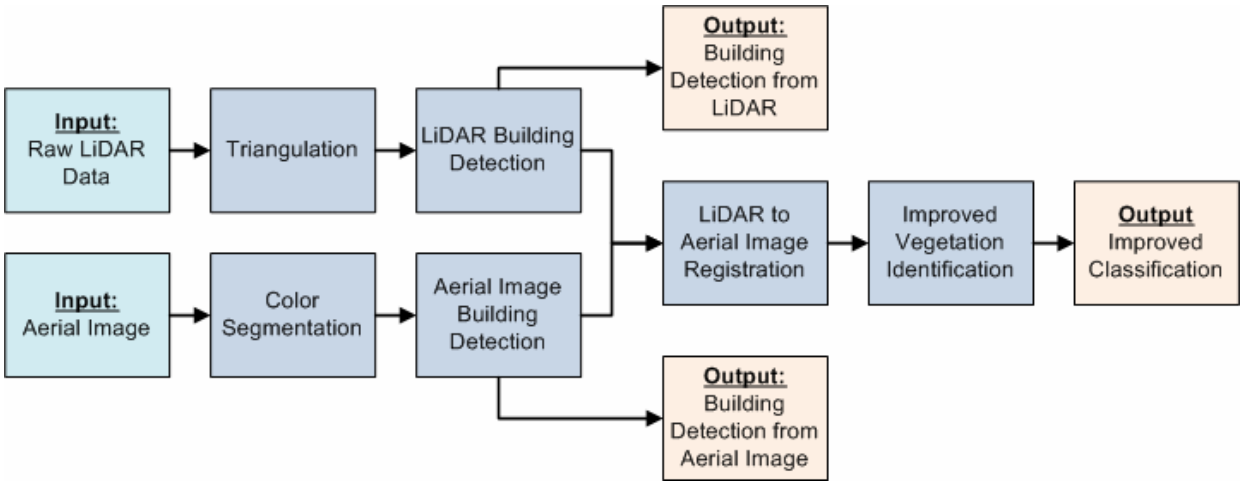


Figure 2: System Implementation Block Diagram

CHAPTER TWO: EXISTING BUILDING DETECTION METHODOLOGIES

The implemented system automatically detects buildings from either aerial imagery or LiDAR data, or both if they are available and if both have overlapping coverage of the same terrain. First, building detection from LiDAR data approaches, where the LiDAR data is taken from a nadir perspective, are reviewed. Second, building detection methods dealing exclusively with a single, nadir aerial image are reviewed. Finally, some methods dealing with building detection from both LiDAR data and aerial imagery are reviewed.

2.1 Approaches for Building Detection from LiDAR Data

Several methodologies published in the literature use morphological filters to differentiate terrain from non-terrain points [19], [28], and [45]. Zhang et. al. in [45] report problems in using a static window size in the morphological operations. An optimum window size (if it does exist) is not always the same throughout the processing of a single data set. Therefore in [19] and [28] the window size is tuned depending on characteristics in the data set. The proposed algorithm eliminates these problems because it does not rely on windowing techniques.

Verma et. al. in [39] propose a model based automatic building detection and reconstruction method. For the building detection, all points which are not locally flat are removed and then the remaining points are grouped to form connected components. The connections between these components (points) are made provided that each point is connected to another point within a certain distance. The authors then assume that the largest connected

component would correspond to all the points belonging to the ground. However, based on varying terrain characteristics, this assumption may not always hold. For instance, what happens if a river, a trench, or an elevated highway partitions the data set into multiple ground planes? Furthermore, what happens if portions of the terrain are surrounded by fences, power lines, neighborhood building walls, or other enclosures? In these scenarios multiple ground planes will be created and thus the assumption of the largest ground plane containing all ground points no longer holds. While it is true that non-locally flat points are removed, the locality again is terrain dependant.

Ortner et. al in [23] propose an approach using stochastic geometry. Their approach requires only a digital elevation model (DEM) that can be derived from either LiDAR data and/or multiple or single view aerial imagery. Point processes models, which are part of the wider stochastic geometry field, allow the modeling of images as random configurations of geometric shapes and provide a natural setup for the inclusion of a priori knowledge on the spatial pattern of features. The authors use both line segments and rectangles as the geometric shapes and parameterize the rectangles as a random configuration having center coordinates, an orientation, a length and a width. One problem with this method is that it is based on the a priori assumption that buildings can be described by rectangles. Buildings with exteriors not having right angles have to be described by multiple rectangles. With the line segment process, sometimes the building exterior contours are not bounded. Furthermore, on a data set with 150 buildings, the proposed approach took 6 hours on a 3 GHz Pentium 4 machine to execute.

Matikainen et. al. in [16] propose the use of a region-based segmentation method which segments a DSM into homogenous regions. Utilizing height differences between the DSM and DTM, textural characteristics of the DSM and of the returned LiDAR intensity, and shapes of the

regions as features, they classified the regions into building, tree or ground surface. The authors analyze the following three attributes for classification: grey level co-occurance matrix (GLCM) homogeneity of height (texture measure), the GLCM homogeneity of intensity and the average length of edges in a 'shape polygon' created from the segmented region. They manually extracted these aforementioned features for a subset of the data (dubbed the training set) and then tested their algorithm on the rest of the data. They developed fuzzy membership functions for recognizing buildings based on the distributions of the features as they pertained to the manually extracted classes (building, tree, and ground). The authors report a 90% completeness and 85% correctness for their approach tested on 259 buildings.

Zhou and Neumann in [46] detect vegetation and ground in LiDAR data and then assume buildings exist as everything else in the data set. The authors strategically pick features to identify vegetation in which are not data dependent but rather global features. Then they propose the use of an SVM classifier, which is a supervised classifier, meaning it requires a training phase where classes have already been identified, to identify vegetation. The authors boast that because of the global features they have selected, the SVM only has to be trained once and not re-trained for additional data sets. This is particularly note worthy as often times approaches employing supervised classifiers have to be retrained for different data sets because of the selected features used to train the classifiers differ in nature across multiple data sets. The authors proposed algorithm works off of the irregular LiDAR data. The authors detect buildings by removing vegetation and identifying ground. This ideology inherently assumes that the only three classes existent in the data set are buildings, ground and vegetation. One immediately asks what would cars, trucks, industrial freight crates, steep hills, power lines, and other miscellaneous objects be categorized as. The features the authors use to train their SVM

classifier to recognize vegetation may also wind up classifying residential buildings as vegetation as well. Two features, 'flatness' and 'Normal distribution' in particular raise concerns. What happens for under sampled residential buildings with multiple roof planes? Those residential buildings probably would have 'flatness' and 'normal distribution' close to that of vegetation. The authors do report in their experiments section that their algorithm, when applied to residential areas of some cities, result in trees becoming the majority objects. The authors report that the gamma and eta parameters have to be set for each data set. It seems contradictory to say that the approach is fully automatic and then have parameters that have to be tuned for each data set.

The algorithms presented in [16], [19], [23], [28], [29], [30], and [45] all natively rely on the raw, irregularly spaced LiDAR data being rasterized/interpolated to fixed point spacing. The methods proposed in [19] and [28] require manual inspection of the terrain to determine the optimal morphological filtering window size; note however that Rottensteiner et. al. in [29], [30] implement additional procedures to find the optimal window size.

2.2 Approaches for Building Detection From Aerial Imagery

Although overlapping geospatial information from multiple sources is becoming more and more available (LiDAR, multispectral, and stereo pair aerial imagery), many areas and data sets still exist with coverage from only a single nadir aerial image. Even with the increasing resolution of newer sensors to capture overhead aerial imagery, several factors still present challenges which hinder perfect building detection. These factors include the scene complexity, building variability, and sensor resolution [17]. One could argue detecting a single farm house existent on a barren hill would be easier than detecting hundreds buildings located in a busy

downtown market place. The more extraneous objects cluttering the image, the more difficult it becomes to rule out what is and what is not building. To further complicate matters, buildings have little to no universal characteristics in which to identify them with. Buildings can have homogeneous (single section or single texture) roof tops and heterogeneous (multiple texture or multiple section) roof tops; their exterior contours can exist as rectilinear or polyhedral or even spline shaped; they can cast long shadows at sunset or minimal shadows during high noon and at different directions dependant at the time of day; and can have a variety of different colored roof tops. Furthermore, not only do the buildings themselves vary in both luminescence and geometry, but the datasets also vary in nature - both in illumination direction and in the type of buildings contained within the dataset. Therefore implementing rule based assumptions or training an algorithm on a data set with buildings existing at a particular color range, or having exterior contours with right angle corners or shadows in only a single particular direction will all result in a loss of generality when being tested against other data sets with buildings containing contrasting characteristics. Therefore it would be ideal to utilize features for training or to implement rules based on assumptions in which do not overly restrict the algorithm to specific circumstances particular only to the data set (i.e. claiming all buildings have red roofs and can only have square exterior contours).

One could also argue, in respect to the challenges of building detection in general, that multiple overlapping sensor coverage on a single given scene would reduce the difficulty of detecting buildings for that scene. Rottensteiner et. al. in [29] and [30] and Vosselman et. al. in [41] make use of both Light Detection and Ranging (LiDAR) data and a single nadir, aerial image for automatic building detection and vegetation identification.

Most building detection from aerial imagery approaches can be classified by whether they are automatic or supervised (require a training phase) and whether they extract geometric features such as lines, corners, etc. or are area based. There are of course exceptions and some methods existing as hybrids employing both geometric features and areas or having some phases automatic and some semi-automatic or supervised phases. Lefevre et. al. in [13] propose an area based automatic building detection from aerial image approach which employs morphological filtering. First, binary images are created by clustering the aerial image's grey scale histogram. Then, multiple clusters are fused together and added to the original set of binary images. Finally, morphological opening, followed by the hit or miss transform and then geo-disc reconstruction are performed for building detection. Their approach was tested on an image having 17,673 pixels and realized a pixel level completeness of 63.6% and a pixel level correctness of 79.4%. No building level completeness or correctness were reported. While the authors approach is automatic, it unfortunately implements the assumption that buildings are square or rectangular. The authors compute a bi-dimensional granulometry of the binary images varying the width and length of the rectangular window. Because they do not vary the orientation of the window, they assume the rectilinear buildings are all parallel with one another and that the image has been rotated so that the sides of the buildings are parallel with the edges of the image.

Muller and Zaum in [20] implement an area and feature based algorithm for building detection from aerial imagery. The image is converted to grayscale and then segmented with a region growing algorithm. Then both photometric and geometric features are detected in each segmented region. Finally, a linear regression classifier then identifies building regions based on the extracted features. Their method is automatic and only takes 45 to 75 minutes on a 6400x6400 pixel image. Unfortunately, they implement the assumption that building rooftop

hues primarily exist in the red channel of the RGB image. Furthermore, of the results pictorially presented in their paper for the 240 tested buildings (most of which have red roof tops), their algorithm had trouble correctly identifying several of the non-red roof top buildings. The authors report a mean completeness and correctness for 79.5% and 78.5% (respectively) for the data sets their algorithm was tested on.

Persson et. al [25] implement a supervised approach for building detection using an ensemble of self organizing maps (ESOM). Then, using the Hue Saturation Value (HSV) representation of the color aerial image, the ESOM is trained to recognize red roofs, light roofs, dark roofs and copper roofs. Rectangles are detected and then classified by ESOM as building or non-building. Their approach realizes a completeness and correctness of 53% and 93% respectively for a campus area. Because their approach is supervised, it requires a training phase. Furthermore, because only rectangles are classified as building/non-building, the approach assumes buildings are rectilinear. Results based on testing across 17 buildings are presented.

Sirmacek and Unsalan in [34] have developed a feature and area based approach employing color invariant features [7] and shadow information for building detection from aerial imagery. Shadows are detected by Otsu Thresholding [24] a blue color invariant image. The authors identify red building roof tops by Otsu Thresholding the red color invariant image. They estimate the illumination direction by calculating the average direction between all the red roof top centroids and their adjacent shadow centroids. The authors then use the illumination angle to find other non-red roof top regions by searching from the shadow region opposite of the illumination angle to nearby adjacent regions. A canny edge detector is run on the image and then a novel box fitting algorithm is grown within candidate building regions by minimizing an

energy function. The inside of the rectilinear box is then assumed to be a building. The authors present results for only a 177 buildings with a completeness of 86.6% (no correctness reported). Their approach assumes buildings are rectilinear and at least some portion of the data set contains red roofs. The illumination angle estimation is based on red roofs only and then is used to verify which adjacent region shadows have been cast from. Furthermore, their approach also assumes buildings are composed of either a single texture or a single roof plane. The canny edge detector will pick up edges where two textures or roof panels join and the novel box fitting algorithm will stop prematurely at the single roof panel adjacent to the shadow, not growing to accompany other adjoining panels of different color or texture.

Liu and Prinet in [14] use a feature and area based approach coupled with a probability function to identify building regions. Their algorithm starts out segmenting the image. Then a set of features (such as contour edges, shadow ratios, shape features, region entropy, etc.) are identified in each region. Then the probability function calculates the confidence value that the given region is in fact a region corresponding to building. Some of the parameters of the probability function are determined from a training set where buildings were manually identified. The authors report a completeness and correctness of 94.5% and 83.4% on a data set having 277 buildings. The authors notice a problem with shadows as they vary under different illumination directions. The authors plan to extend their work by testing their approach on different imagery at various resolutions.

Surveying the related literature, several common issues arrive in automatic and semi-automatic/supervised building detection approaches from aerial imagery. Some will make limiting assumptions which may reduce the ability of the algorithm when executed for data sets other than what were presented in the associated paper. Examples of such limiting assumptions

would be buildings only having specific root top colors ([20], [25], [34]), buildings only existing at right angle corners and/or buildings having parallel sides ([13], [34]). Rather than saying all buildings have parallel sides or right angle corners, a more general assumption would be that buildings tend to have convex hull rooftop sections. Furthermore, many houses have a variety of different color roof tops (other than red) and commercial and industrial building roof tops include grey, black and white. Finally, the results should be tested on several hundred or even thousands of buildings of varying sizes, shapes, orientations, roof top colors, and roof top textures and from more than one data set if possible to thoroughly benchmark the generality of the algorithm. Several of the current related papers in the literature ([14], [20], [25] and [34]) only present results from a couple of hundred buildings and some from only a single data set. While the semi-automatic/supervised approaches ([14] and [25]) tend to have good accuracy, they rely on manually extracting features from 1/4th to as much as ½ the data set. Furthermore, unless an emphasis was placed on training the algorithm with data set invariant features, the algorithm will have to be retrained for each data set to achieve relatively close accuracy as what is presented in their associated paper. Furthermore, both completeness and correctness should be reported both on a global pixel rate as well as a function of the building size.

2.3 Building Detection From LiDAR Data and Aerial Imagery

There are several approaches in the literature which make use of both LiDAR data and aerial imagery taken from a nadir perspective which both contain overlapping coverage of the same terrain. Rottensteiner et. al in [30] propose the use of the Dempster-Shafer fusion method to implement building detection based on the use of five feature sets: height differences between

digital terrain model (DTM) and digital surface model (DSM); strength of surface roughness; directedness of surface roughness; height differences between the first and last pulse DSMs; and an NDVI image (page 452 of [37]). The DTM only encompasses bare earth (no building, vegetation, etc.) whereas the DSM contains everything sampled by the LiDAR data. The NDVI image is constructed by using the red band from the aerial image and the returned LiDAR intensity values for the infrared band. This method therefore also relies on the registration of the LiDAR data to aerial imagery if the red aerial image color components are to be paired with the corresponding returned LiDAR intensity existing at the same location. Furthermore their proposed method requires visual inspection of the data to estimate the 'P_T' algorithm input parameter which corresponds to the percentage of terrain covered by trees. The authors report that their algorithm performs optimally provided that the 'P_T' parameter is known within 5% of its true value.

Vosselman et. al in [41] develop a data driven building detection approach which uses LiDAR and aerial imagery data. The LiDAR point cloud is divided into sets of parallel thin slices in the XY plane. A spanning tree is then computed for each profile and by removing tree edges that exceed a certain slope or length threshold, the tree is split into line segments which are then merged with surface segments. Then, segments with a low portion of line segments that are above neighbouring line segments are classified as bare earth. The authors claim their approach is automatic, however they require that the LiDAR data be registered to the aerial imagery. Automatic registration of LiDAR to aerial imagery is not by any means trivial and the authors do not provide a solution to this in their paper. Their algorithm also has issues with sometimes classifying walls of buildings as vegetation and vegetation close to buildings as part of the

building. The authors report the algorithm having completeness and correctness values of 85% and 90% respectively for their tested data sets.

Chen et. al. in [1] propose a two step building detection scheme. They first implement a region based segmentation scheme on both LiDAR data and aerial imagery and then perform object based classification. They merge regions having similar geometric and spectral properties. Object-based classification is performed by considering characteristics of elevation, spectral information, texture, roughness and shape to detect building regions. The elevation of a given pixel is obtained by subtracting a DTM from a DSM. Then an elevation threshold is set to detect objects above ground. A greenness index is then used to distinguish vegetation from non-vegetation areas (it is not mentioned how the greenness index is determined). A texture analysis from the aerial image based on entropy and homoegeneity is used to separate building from vegetation when objects have similar spectral responses. The roughness of the LiDAR data aims to differentiate vegetation from non-vegetation. Finally a shape attribute which includes size and length to width ratio is used. A minimum area threshold is implemented to remove smaller objects which are most likely not building. The authors report their data set having an average point density of 1.6 pts/m^2 and an ortho-rectified aerial image having a pixel resolution of 10 cm. Their approach was able to detect 79 out of 89 buildings realizing a detection rate of 81%. No correctness is reported. Furthermore, of the results presented, most are relatively large buildings. Finally, the authors report the 10 buildings that were not detected were smaller than 35m^2. No mention of how the DTM and DSM were generated (probably with morphological filters with a constant window size). Because a DTM and DSM were used, the LiDAR data was most likely interpolated.

Hu et. al. in [10] threshold the LiDAR data at a given height for building detection. Objects below a certain height are classified as terrain and objects above a certain height as building or vegetation. The authors then use the color information from the aerial imagery to further discriminate the buildings from vegetation. If a point exists above the previously applied height threshold and is green it is labeled as vegetation. The authors however do not report building level or pixel level completeness or correctness for their algorithm. It is not mentioned how the green colors are extracted for vegetation detection.

Several common limitations/drawbacks are existent in LiDAR and aerial image building detection methods found in the recent literature. All of the aforementioned building detection methods in this section ([1], [10], [30], and [41]) treat the registration of the two data sources (LiDAR and aerial imagery) as a pre-processing technique and proceed to manually register the souces together. The algorithms presented in [30] and [41] also require that the irregularly spaced LiDAR data be rasterized/interpolated to fixed point spacings. The method in [30] requires first and last pulse returns as well as the LiDAR intensity and requires that the end user estimate the amount of vegetation existent in the terrain within 5% of the ground truth.

CHAPTER THREE: BUILDING DETECTION FROM LIDAR DATA

A novel paradigm is presented which employs homotopy trees, a branch of topography, for building detection in Light Detection and Ranging data. The method is developed for irregularly spaced LiDAR data and therefore it can also be applied to rasterized/grid spaced data without any modifications. Using features extracted from either the first and/or last returns (if available) of the LiDAR pulses and the triangulation of the LiDAR data, the proposed method can discriminate ground from non-ground points and subsequently differentiate non-ground as building or non-building points. The only assumption the algorithm makes about the building structures is that they exist as convex hulls that protrude from the ground with a minimal predefined area and height. The following subsections detail the specifics of the proposed algorithm. First, homotopy trees are discussed and then second pseudo homotopy trees are presented. Third, some notation associated with pseudo homotopy trees is introduced. Fourth, the process of how to use homotopy trees to identify the ground level is presented. Finally, some heuristics are formulated for how to differentiate building from non-building.

3.1 Homotopy Trees

For a concise definition on homotopy trees see page 44 of [35]. A brief discussion as they pertain to the proposed algorithm will be outlined next, but first the concept of connectivity must be defined. A set is considered connected if each pair of the set's elements can be joined by a path where all the elements of the path also belong to that set. A triangulated irregular network (TIN) is an approximation of a 3-dimensional depiction of a given collection of points which in this case is a LiDAR point cloud describing terrain sampled by a LiDAR sensor. The TIN itself

is then a bounded set of interconnected triangles with non-overlapping edges (all triangles are completely visible from a top down or nadir perspective). The concept of connectivity is demonstrated in the arbitrarily created TIN shown in Figure 3 where all the light grey triangles belong to one set and the dark grey triangles belong to another set but the light grey set and the dark grey set do not compose a single set as the two sets are not connected to one another.

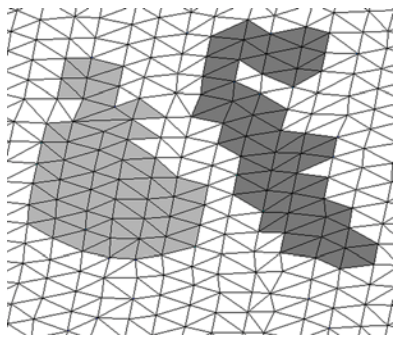


Figure 3 - Example TIN Connected Sets

Define X as a bounded set containing a collection of connected components as shown in Figure 4. Then, the complement of X, denoted as X^c would then contain all the space in which X does not occupy. A homotopy tree of a bounded set X is defined as having a root X_0 corresponding to the unbounded connected component of X^c (white background shown in Figure 4). The first level or first generation of nodes corresponds to the connected components $X_{1,i}$ which belong to X and are adjacent to X_0 (the branches of the tree indicate which nodes or connected components are adjacent to other connected components). The second generation of nodes correspond to the bounded components $X_{2,i}$ which belong to X^c and are adjacent to each $X_{1,i}$. Future generations are added as additional sets are found to be adjacent to already considered sets. Notice that every even (and 0) level corresponds to X^c and all odd levels correspond to X. From this definition, tree structures or homotopy trees can then be constructed to describe the sets. Two sets and their trees are depicted in Figure 4 and Figure 5. In Figure 4,

observe how $X_{1,1}$ is adjacent to $X_{2,1}$, and $X_{1,1}$ exists in one generation older than $X_{2,1}$. The set $X_{1,1}$ is then referred to as the parent of $X_{2,1}$ and $X_{2,1}$ is referred to as the child of $X_{1,1}$.

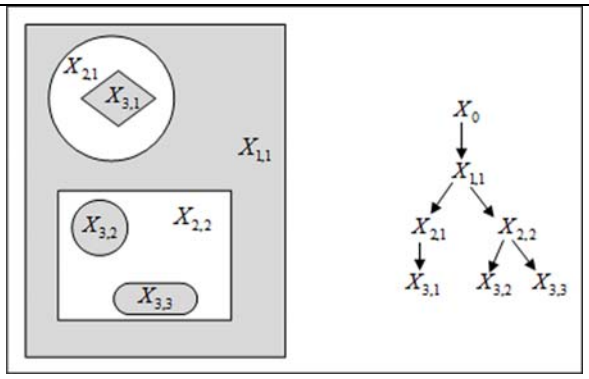


Figure 4 - 1st Homotopy Tree Example

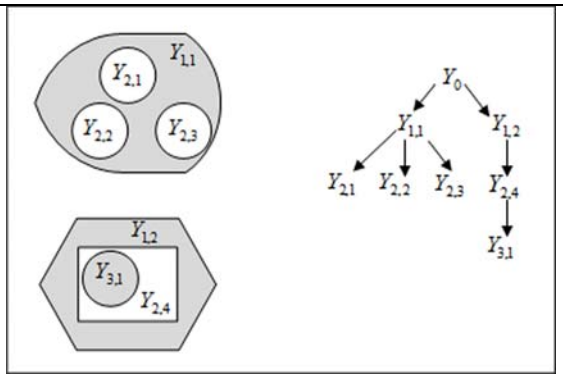


Figure 5 - 2nd Homotopy Tree Example

3.2 <u>Psuedo Homotopy Trees</u>

3.2.1 Wall Triangles, Non-Wall Triangles and Connected Sets

For the purpose of building detection, the formal definition of homotopy trees is slightly modified, where the modified definition is referred to as the pseudo homotopy tree (PHT). Before discussing PHTs, first some definitions about triangles existent in the LiDAR TIN and definitions pertaining to the LiDAR data itself are necessary. Let P be the set of N irregularly distributed points p_i in the raw LiDAR point cloud:

$$P = \left\{ p_i \middle| i \in [1, N] \right\} \tag{1}$$

Those N irregularly distributed points are triangulated via a modified greedy insertion triangulation algorithm [31] into a TIN. If the LiDAR set is too large, it is evenly partitioned into smaller groups so that it is more manageable to work with.

Two types of mutually exclusive triangles are defined: wall triangles (WT) and non-wall triangles (NWT). Consider the blue triangle in Figure 6. A triangle's orientation θ is defined as the angle between the vector n normal to the triangle's plane (formed by its three vertices) and the z or elevation axis (pointing straight up). A WT is defined as a triangle having an orientation θ greater than 45 degrees (θ>45) and a difference Z_{diff}, between its maximum Z_{max} and minimum Z_{min} points in elevation, of at least 1 meter ($Z_{diff} > 1$). A NWT is then defined as a triangle violating either of the two conditions needed to be a wall triangle.

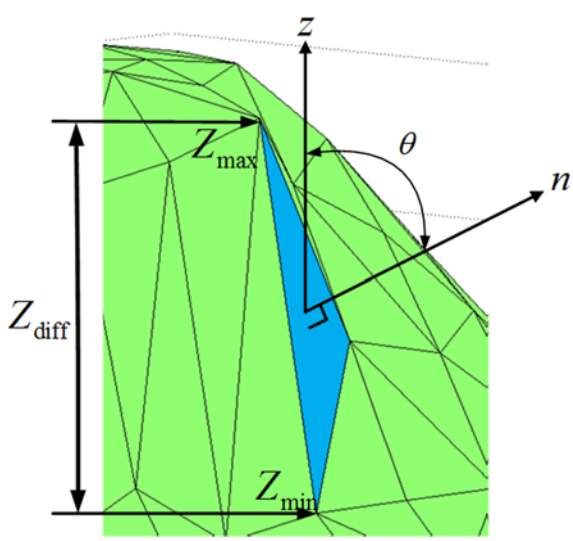


Figure 6 - Wall Triangle Orientation and Elevation Difference

In a LiDAR TIN, a wall connected (WC) set is defined as the set of wall triangles in which are all connected to one another. A non-wall connected (NWC) set is a set of non-wall triangles in which are connected to one another. A ground connected (GC) set is a NWC set that

corresponds to the ground in the depicted terrain. A first generation connected (FGC) set is a WC set whose immediate parent is a GC set.

3.2.2 Constructing the Pseudo Homotopy Tree

In the process of defining a PHT, at first the PHT is assumed to be equivalent to a HT and then at the end their differences are shown. Let S be the set of all Y triangles t_i in a given LiDAR TIN: $S=\{t_i|i\in[1,Y]\}$. Let Z^c be the set of all non-wall triangles in the LiDAR TIN and Z be the set of all wall triangles. Note that because a NWT cannot be a WT and vice versa then $Z\cup Z^c=Y$ and $Z\cap Z^c=\emptyset$.

Let t_0 be the NWT with the lowest elevation average, calculated by taking the average of the elevation component of the triangle's vertexes. Region growing is then applied by connecting all NWTs which are connected to this lowest elevation average triangle t_0. These NWTs then form the NWC set Z_0 which is assumed to be either a trench or a ground plane. Wall triangles encountered during this region growing process are stored in Z_0's WT queue. Then, a single WT is taken from Z_0's WT queue and region growing is applied forming a WC set - $Z_{1,1}$. Note that $Z_{1,1}$ forms the 1st generation and is a child to the Z_0 parent node. If not all of the WTs are used up during this region growing process, then region growing is applied on unused WTs forming more WC sets - $Z_{1,2}$, $Z_{1,3}$, etc. It is possible for several different WC sets to be immediately adjacent to a single NWC set. Therefore, any node in the PHT can have multiple children but all nodes (except the root node) can only have one parent node (conversely no node can have multiple parents). While these WC sets $Z_{1,1}$, $Z_{1,2}$ and $Z_{1,3}$ are being grown, any unused

NWTs that are encountered are stored in their respective queues. Once all the WTs from Z_0's queue are used up in growing WC sets $Z_{1,j}$ then all the NWTs in $Z_{1,j}$'s queues are used as seeds for growing NWC sets $Z_{2,1}$, $Z_{2,2}$, etc. This process, depicted in Figure 7, continues iteratively as the PHT is grown. The root node or the NWC set containing the lowest elevation NWT is always referred to as Z_0. All future nodes are labeled $Z_{i,j}$ where i corresponds to the level or generation of the connected set and j corresponds to the node index of a given node in generation i.

The WC sets are equivalent to set X in section 3.1 and the NWC sets equivalent to X^c. However, because a single ground plane does not necessarily always surround all WC sets (the terrain could be partitioned by an elevated highway or a trench), the concept of a PHT had to be developed. There are only three differences between the PHT and an HT which are as follows. First, the root node or NWC set containing t_0 (recall this is denoted as Z_0) will always be bounded (because there are a finite amount of LiDAR points) unlike the HT which must have an unbounded root node. Second, the root node may not always surround (from a nadir perspective) all of the WC sets. Finally, in an HT X refers to an actual set describing a surface and X^c refers to a set describing an empty space (the set and empty space both being mutually exclusive) whereas in the PHT X and X^c are equivalent to two different types of sets - WC sets and NWC sets, neither of which are empty space but both being mutually exclusive from one another. Despite the subtle differences between the PHT and the HT, the paradigm of representing a surface as a homotopy tree (or for this application as a pseudo homotopy tree) and describing properties of that surface with set theory are very practical for building detection in a LiDAR TIN, as is soon shown in the results.

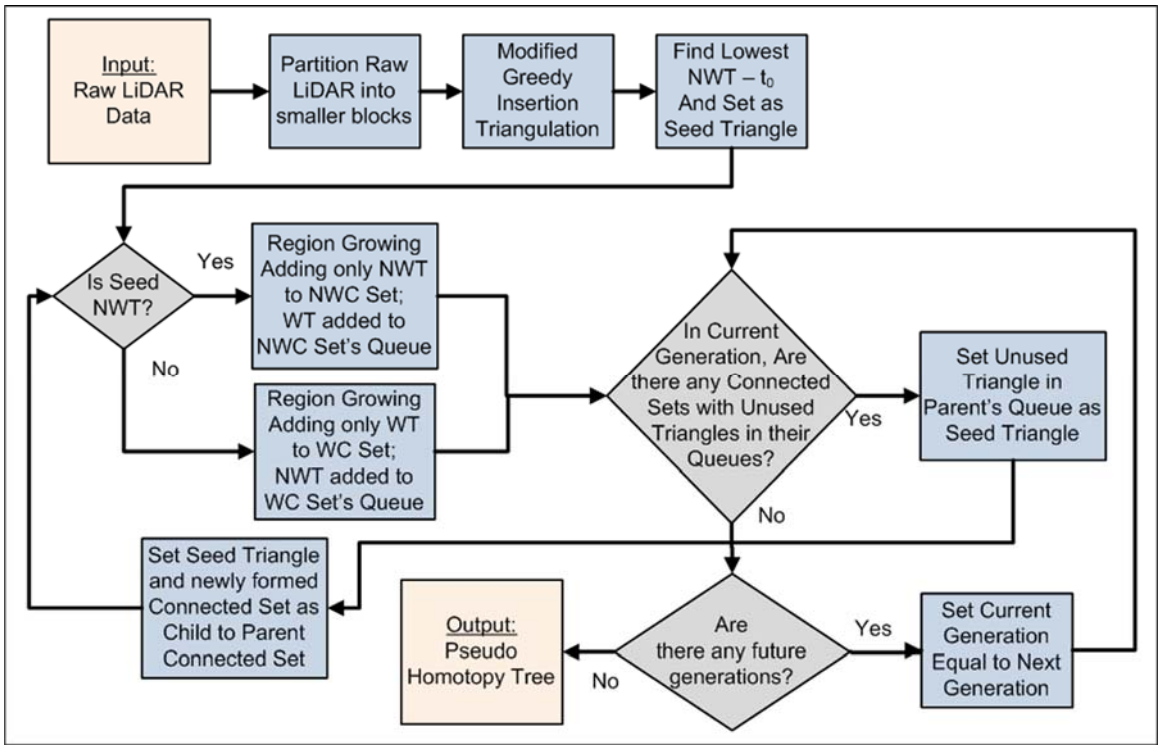


Figure 7 - Pseudo Homotopy Tree Growing Block Diagram

3.2.3 Pseudo Homotopy Tree Notation

Let *G* denote the total number of generations in a given PHT and *J(i)* denote the total number of nodes in each generation *i*. For example, in Figure 5, J(1) = 2, J(2) = 4, and J(3) = 1. Note for all PHT's and all HT's, J(0) = 1 as there can be only one root node. Observe during the PHT construction process, as described in 3.2.2, how all the NWC sets exist at all even generations:

$$Z^C = \bigcup_{i=2\cdot k}^{G} \bigcup_{j}^{J(i)} Z_{i,j} \quad k=0,1,2,\ldots,G/2 \quad j=1,2,\ldots,J(i) \tag{2}$$

The WC sets exist at all odd generations :

$$Z = \bigcup_{i=2\cdot k+1}^{G} \bigcup_{j}^{J(i)} Z_{i,j} \quad k=0,1,2,\ldots,G/2 \quad j=1,2,\ldots,J(i) \tag{3}$$

For connected set - $Z_{i,j}^{n,m}$, let superscripts n denote the number of children node j at generation i possesses and m denote the node index of $Z_{i,j}^{n,m}$'s parent connected set as it exists in the previous generation. Therefore, if a given connected set belongs to generation 3, has a node index of 2, has 4 children, and has a parent whose node index was 1 in the previous generation, then that connected set would be denoted as $Z_{3,2}^{4,1}$. Let the connected set $C_{i,j}^{n,m}$ denote the union of connected set $Z_{i,j}^{n,m}$ and all of connected set $Z_{i,j}^{n,m}$'s children and children's children in all future generations. Recalling Figure 5, let's add the superscripts to make the connected sets Y conform to the notation mentioned in this section, now represented in Figure 8:

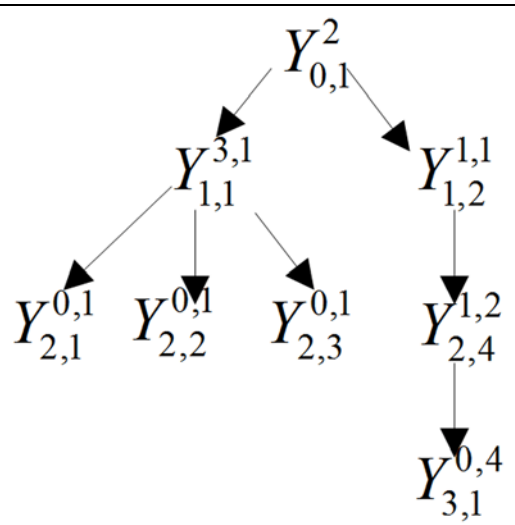


Figure 8 - Homotopy Tree Example with Child and Parent Notation

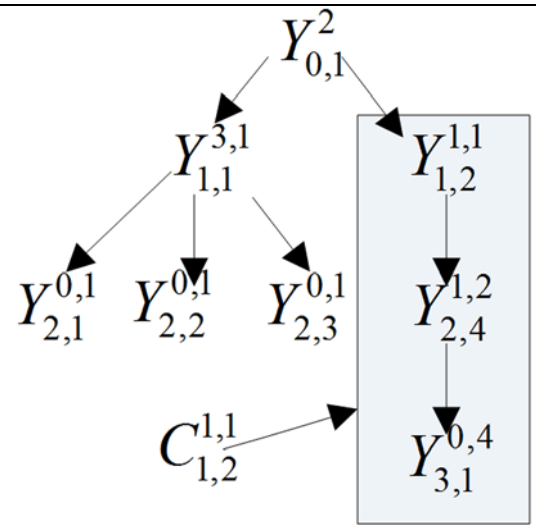


Figure 9 - Homotopy Tree Example with C variable

Note that because the root node has no parent, the parent superscript index is dropped. In Figure 9, the $C_{1,2}^{1,1}$ denotes the union of $Y_{1,2}^{1,1}$ and $Y_{1,2}^{1,1}$'s child $Y_{2,4}^{1,1}$ and $Y_{1,2}^{1,1}$'s children in all future

generations which in this case also includes $Y_{3,1}^{0,4}$, or in other words - $C_{1,2}^{1,1} = Y_{1,2}^{1,1} \cup Y_{2,4}^{1,1} \cup Y_{3,1}^{0,4}$. Finally, the last bit of notation introduced is the S(Z) function which returns the size of connected set Z or in this case the area in m^2 in which connected *Z* spans over the given terrain. Given that the units of measurement (feet/meters/etc.) are provided with the data set, the area each triangle covers from a nadir perspective can be calculated and then all the areas for all the triangles in connected set *Z* can be summed. For example, if connected sets $Y_{1,2}^{1,1}$ covers 3 m^2, $Y_{2,4}^{1,1}$ covers 4 m^2, and $Y_{3,1}^{0,4}$ covers 1 m^2, then $S\left(Y_{1,2}^{1,1}\right) = 3m^2$, $S\left(Y_{2,4}^{1,1}\right) = 4m^2$, $S\left(Y_{3,1}^{0,4}\right) = 1m^2$ and $S\left(C_{1,2}^{1,1}\right) = S\left(Y_{1,2}^{1,1}\right) + S\left(Y_{2,4}^{1,1}\right) + S\left(Y_{3,1}^{0,4}\right) = 3m^2 + 4m^2 + 1m^2 = 8m^2$.

3.3 PHTs for Identifying Ground Level

It is first shown how PHTs can be utilized to differentiate the ground level from everything else. Then in the next section, a collection of heuristics are implemented to differentiate buildings from non-buildings. Consider the following terrain profile shown in Figure 10 (nadir view) and Figure 11(side view).

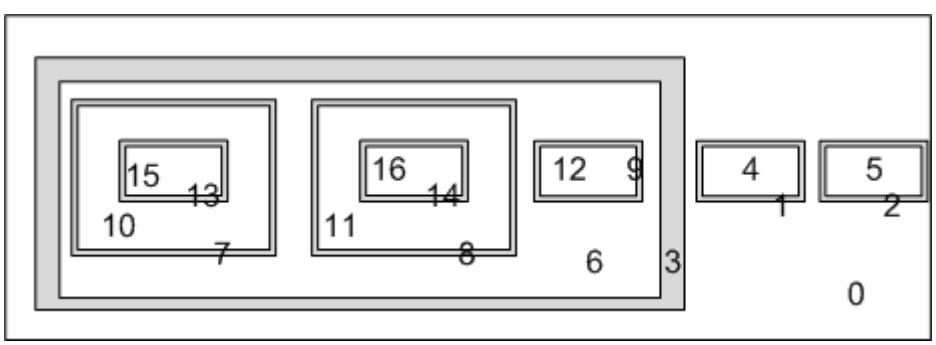


Figure 10 - Top/Nadir view of example terrain

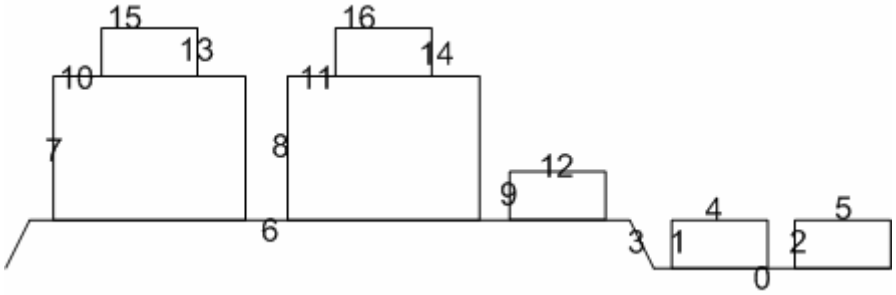


Figure 11 - Side view of example terrain

In the example depicted by the terrain profile, let connected sets 0 and 6 correspond to ground levels and connected sets 2, 3, 7, 8, and 9 correspond to the bases of buildings. A step by step description of the algorithm's execution, on the example depicted in Figure 11, is provided as follows. The PHT algorithm starts out at the lowest NWT which would exist in connected set 0. The algorithm then finds all NWTs existent in that connected set, which are adjacent to one another, thus forming a NWC set. All the while the algorithm would add the WTs in connected sets 1, 2 and 3, which are adjacent to the NWTs in NWC set 0, to the queue for the NWC Set 0. After all the connected, adjacent NWT's are found for the NWC Set 0, then a WC set is formed from one of the WTs in NWC Set 0's WT queue. Let's assume that connected set 2 in Figure 11 was the next WC set formed. Then all the connected, adjacent WTs in that connected set would belong to WC set 2. Note WC set 2 would be the child of NWC set 0. Remember, while the algorithm grows the WC set corresponding to connected set 2, the NWT's existent in connected set 4 of Figure 11 would be added to WC set 2's queue.

Then the algorithm would go back to NWC set 0's queue and continue growing WC sets until NWC set 0's queue was emptied. Let's say then WC sets 1 and 3 are then grown from the WTs from the queue of the NWC set representing connected set 0. After NWC set 0's queue is exhausted, the algorithm proceeds to NWC set 0's children (which are WC sets 1, 2, and 3) and

grows the NWC sets from the NWTs in their (1,2 and 3) queues. Then NWC sets 4, 5 and 6 are grown. Following this, WC sets 7, 8 and 9 are grown while NWT triangles in connected set 10 are added to WC set 7's queue. Also, NWT triangles in connected sets 11 and 12 are added to WC sets 8 and 9's queues respectively. Then NWC set 10, 11 and 12 are grown. Following this, WC sets 13 and 14 are grown. Then from their (13 and 14) queues NWC sets 15 and 16 (respectively) are grown.

After all connected sets are formed; the hierarchical relationship or PHT between those connected sets is then analyzed. For the simplified example in Figure 11, the PHT generated is displayed in Figure 12. Using the notation established in section 3.2.3, the sets are relabeled and the tree redone in Figure 13.

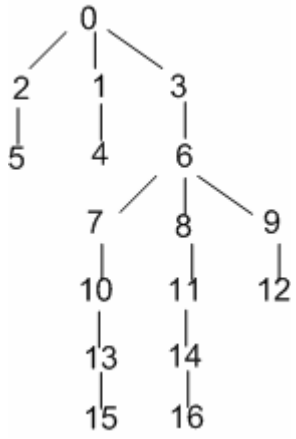


Figure 12 - Pseudo Homotopy Tree for Sample Terrain

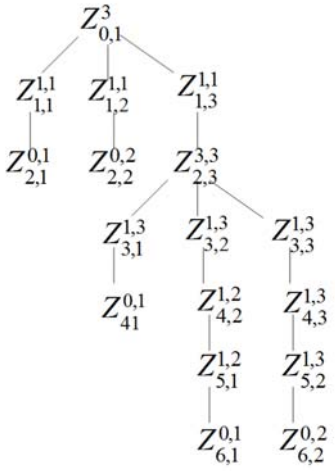


Figure 13 - PHT Labelling for Sample Terrain

Notice how the GC sets (sections 0 and 6) have a larger number of immediate child sets than the other sets. In terms of the data, this means there are more buildings protruding from the ground then there are smaller structures (air conditioning vents, chimneys, etc) protruding from the roofs of buildings. In actual data, it was found that GC sets had a significantly larger number

of immediate child sets than all other sets existent in the PHT. Therefore, the NWC sets with the largest number of child nodes are the GC sets. A simple thresholding technique automatically identified these GC sets. In the simplified example, this establishes sections 0 and 6 as GC sets. Using eq (2) and eq (3), one can see that all the NWC sets are sets 0, 4, 5, 6, 10, 11, 12, 15 and 16 (all even generation nodes) and all the WC sets are 1, 2, 3, 7, 8, 9, 13, and 14 (all odd generation nodes).

3.4 Heuristics for Differentiating Building from Non-Building

Several heuristics are implemented to accurately differentiate building from non-building structures.

- Heuristic I: GC sets will have a significantly larger number of child WC sets than any other NWC sets. Let E be the total number of connected sets in the data set. A NWC set is a GC set if a NWC set has at least 0.05*E children. That means if a single NWC set has at least 5% of the total number of connected sets E as its immediate children, then that NWC set is labeled a GC set. The 5% threshold was determined empirically.

$$GC = \left\{ Z_{i,j}^{n,m} \middle| \left[n > 0.05 \cdot E \right] \quad \wedge \quad \left[i = 0,2,4,6... \right] \right\} \tag{4}$$

In (4), the generation i must be even or 0 as all even (and root node or generation 0) sets are NWC sets and GC sets are by definition only NWC sets. Therefore (4) states that all ground connected sets GC are equal to all NWC sets given those NWC sets $Z_{i,j}^{n,m}$ have children n greater than 0.05*E.

- Heuristic II: There are two heuristics which are employed to identify connected sets in which do not belong to buildings or ground. The first is by examining a FGC set's

children: if a FGC set has no NWC set children, or in other words, if an isolated connected set of only WTs protrudes from a GC set, then that connected set is labeled as non-building. Let *NBC* be the set of all connected sets defined as connected sets not belonging to buildings or ground. Then let *NBC1* correspond to the first way in which connected sets are identified as not belong to buildings or ground and let *NBC2* be the second way where NBC = NBC1∪NBC2. Then *NBC1* is formulated as follows:

$$NBC1 = \left\{ Z_{i,j}^{n,p} \middle| \quad [n=0] \quad \wedge \quad [i=1,3,5,7...] \quad \wedge \quad [p \rightarrow GC] \right\} \tag{5}$$

Equation (5) states that connected sets not belonging to buildings or ground, denoted as *NBC1*, contain wall connected sets (hence the generation index *i* being only odd) who have GC set immediate parents (p→GC) and no children (n = 0). Note that in (5), *p* is the set of all node indices pointing only to GC set nodes. Recall that FGC sets are WC sets (therefore their generation index *i* is odd) and their immediate parent connected set is a GC set. Looking back to the example in Figure 11, NWC sets 1 in Figure 12 (or $Z_{1,1}$ in Figure 13) ,2 ($Z_{1,2}$) ,3 ($Z_{1,3}$), 7 ($Z_{3,1}$), 8 ($Z_{3,2}$) and 9 ($Z_{3,3}$) are FGC sets and their *p* indices for that example are the node indices of the GC sets ($Z_{0,1}$) and ($Z_{2,3}$) in Figure 13 which are 1 and 3: p∈[1,3]. This heuristic simply states that in order for a FGC set (and all of its immediate and future generation children) to be labeled as building, it must have NWC set children. Another way of stating this is that buildings (whose walls are represented as FGC sets) will have roofs on top of them (where those roofs are represented by the FGC sets' NWC set children).

- <u>Heuristic III:</u> The second heuristic utilized to identify non-building connected sets examines the connected set's total area: if the sum of the area of all of the triangles

belonging to a FGC set and all its children sets accumulates to less than 40 m^2, then label those sets as non-building (dense vegetation, cars, trucks, shacks, etc.). This heuristic simply places a minimum area a collection of connected sets must occupy in order to be classified as building and it is formulated as follows:

$$NBC2 = \left\{ C_{i,j}^{n,p} \middle| \ \left[S\left(C_{i,j}^{n,p}\right) < 40m^2 \right] \ \wedge \ \left[i = 1,3,5,7... \right] \ \wedge \ \left[p \rightarrow GC \right] \right\} \tag{6}$$

Equation (6) states that non-building connected sets *NBC2* contain FGC sets and their child connected sets $\left(C_{i,j}^{n,p}\right)$ which have a total of area less than 40 m^2 $\left(S\left(C_{i,j}^{n,p}\right) < 40m^2\right)$. Because there are more smaller patches of vegetation than there are buildings with area less than 40 m^2 and because it is so difficult to reliably detect buildings with area less than 40 m^2, this threshold was set as the minimum area buildings must cover; otherwise buildings, as well as a vast majority of dense vegetation, with an area of smaller than 40 m^2 are classified as non-building.

- <u>Heuristic IV:</u> If a FGC set has at least one NWC set child, at least 40 m^2 coverage contained in the FGC set and all child sets combined, and that FGC set's parent is by definition a GC set, then the FGC set and all child sets contain building points. Let *BC* denote the set of all connected sets depicting buildings.

$$BC = \left\{ C_{i,j}^{n,p} \middle| \ \left[S\left(C_{i,j}^{n,p}\right) \geq 40m^2 \right] \ \wedge \ \left[n > 0 \right] \ \wedge \ \left[i = 1,3,5,7... \right] \ \wedge \ \left[p \rightarrow GC \right] \right\} \tag{7}$$

- <u>Heuristic V:</u> If in a future generation NWC set, a child to a FGC set (where both sets have been labeled as building) has an average elevation of less than 3 meters plus the FGC set's floor average, then that future generation NWC set is considered a GC set. Three meters is chosen as it is assumed that buildings are at least 3 meters tall. The

average elevation of a connected set is the average of all of the elevations of the vertices of all the triangles in a given connected set. Let all WTs in a FGC set that are immediately adjacent to the NWTs in the parent adjacent GC set be considered floor WTs. Then, the floor elevation of a FGC set is the average of the elevations of the lowest two vertices of all floor WTs in the FGC set. Consider Figure 14 and assume the yellow connected set is a FGC and the orange connected set is the FGC set's parent, a GC set. All of the points which would go into calculating the yellow FGC set's floor average are circled in light blue. All triangles having two of its points circled in blue are the floor WTs of that FGC set.

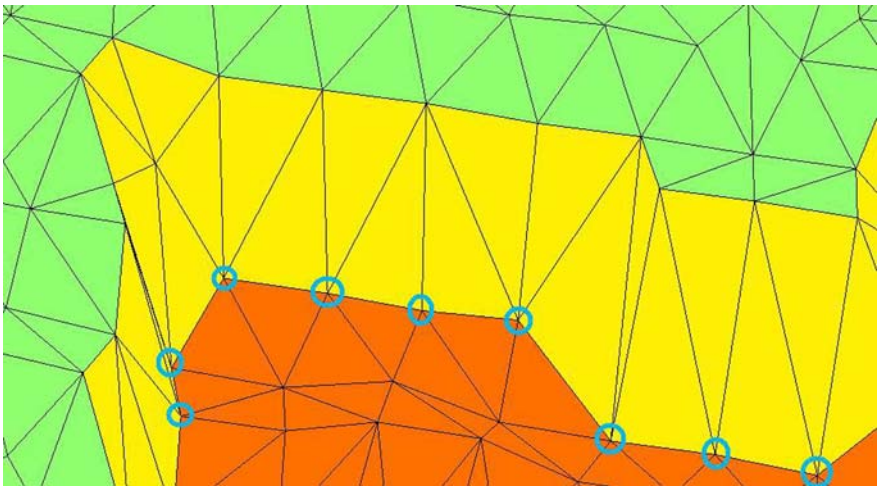


Figure 14 - Example Depicting FGC Set Floor Average

If in a collection of connected sets marked as a building a future NWC set has an average elevation lower than 3 meters plus the FGC set's floor average, then that NWC set is most likely an enclosure, a court yard or small portion of the terrain (not having many child WC sets), which is surrounded by neighboring buildings. This heuristic prevents that NWC set from erroneously being labeled as building by re-labeling that connected set back to ground. An example of such a

scenario is depicted in Figure 15 (LiDAR TIN) and Figure 16 (corresponding aerial image coverage).

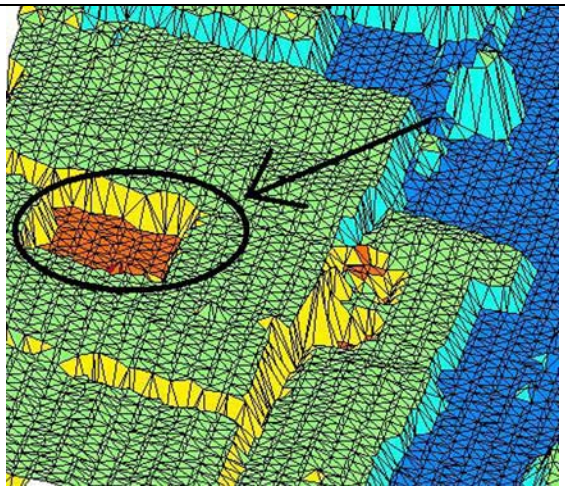


Figure 15 - LiDAR Enclosure Example

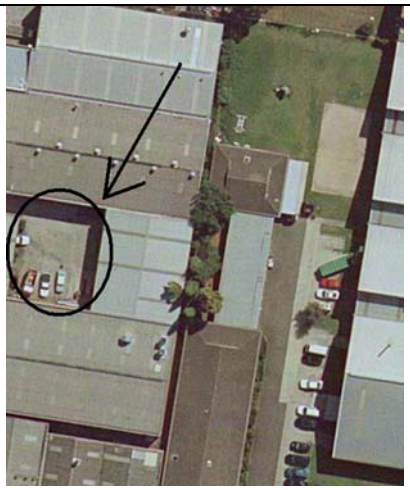


Figure 16 - Aerial Image Enclosure Example

In some rare cases, some buildings actually have ramps leading from the ground to their roofs. Therefore, during region growing when forming the NWC set that is the GC set, that region growing will add all the NWT's on the ground as well as the NWT's on the ramp of the building and finally NWT's on the roof of the building. Most building roof top NWT's are typically isolated from nearby GC set NWT's because those building roof tops are surrounded and bounded by WT's from the building's surrounding WC sets which depict the building's walls. However, if any part of the building exterior has a ramp connecting the ground to the roof, then the roof is erroneously labeled part of the GC set during the region growing / PHT building phase of the algorithm. Figure 17 (LiDAR TIN) and Figure 18 (corresponding aerial image coverage) depict a scenario of a ramp connecting the ground to a building's roof (parking garage). Note that in both pictures, a pink arrow is shown pointing up the ramp. In the LiDAR TIN, the dark blue corresponds to GC set, the light blue to FGC sets, and the light green and orange are future generation connected sets which are children to the light blue FGC sets.

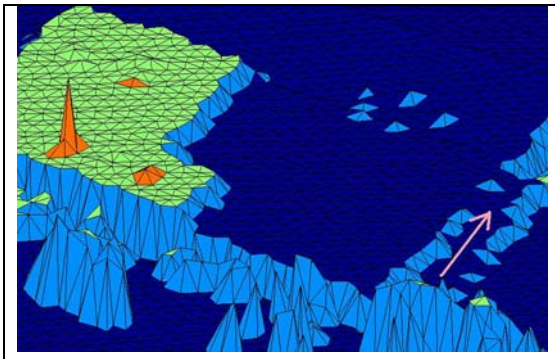


Figure 17 - LiDAR TIN Ramp Example

Figure 18 - Aerial Image Ramp Example

- Heuristic VI: If NWTs adjacent to the top of a FGC set's WTs belong to a GC set, apply region growing and label all adjacent NWTs as building provided those NWTs average elevation exists 3 meters above the FGC set's floor average. What this heuristic effectively does is find NWTs which are adjacent to the tops of WC sets and then iteratively labels those NWTs as building provided the NWTs exist above the WC sets' floor average. If a building has a ramp leading to its roof, the ground connected set will bleed onto the roof. This heuristic then correctly re-labels those triangles back to building. If the region grows to become adjacent to more than 30 different WC sets, then the region is kept marked as ground. If more than 30 WC sets are adjacent to this growing region, then the region is most likely not building but rather a GC set existing on top of a trench.
- Heuristic VII: Exterior building WTs (which will belong to a FGC) must be adjacent to a building NWC set by at most through one adjacent triangle; otherwise the WT will be labeled as ground. If long strands of trees exist adjacent to buildings, the building WC set can bleed outwards to erroneously include these tree strands. This heuristic prevents this from happening.

- Heuristic VIII: This heuristic is executed provided a given data set has both the first and last return LiDAR pulse sets. If triangles (belonging to a FGC) have vertices (or raw points) with first and last return pulse differences of greater than 4.6 meters (Clode 2004), then those triangles are labeled as non-building. Points with a high first and last return pulse difference correspond to edges of walls and vegetation.
- Heuristic IX: It is assumed that t_0, the lowest NWT contained in the given terrain, belongs to either the ground and/or a trench and not a building.
- All of these heuristics, plus the PHT growing algorithm shown in Figure 7, compose the PHT building detection algorithm as depicted in Figure 19:

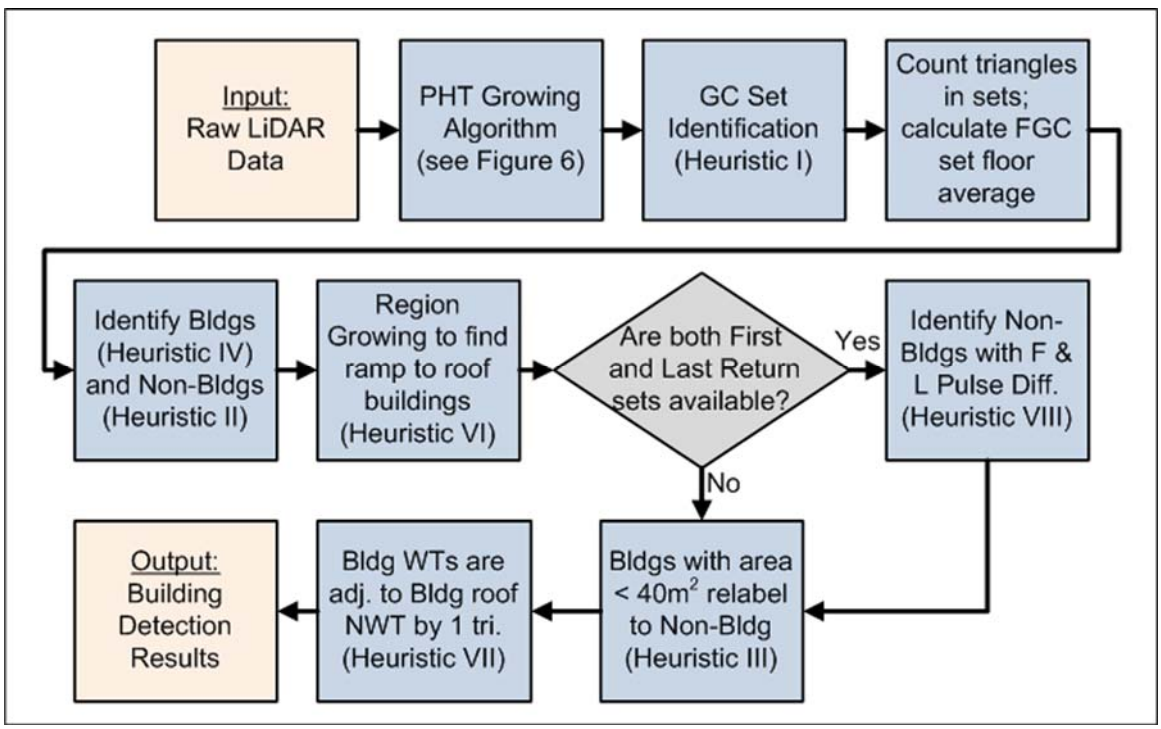


Figure 19 - PHT Building Detection Algorithm Block Diagram

CHAPTER FOUR: BUILDING DETECTION FROM AERIAL IMAGERY

A novel, automatic tertiary classifier is proposed for identifying vegetation, building and non-building objects from a single nadir aerial image. The only assumption the algorithm makes about the building structures is that they have convex rooftop sections. The major processes employed by the algorithm are summarized as follows. First, the input image is segmented by a proposed color segmentation algorithm. Second, vegetation regions are identified and removed. Third, shadows are identified and removed. Fourth, the original color image is segmented again using a texture segmenting algorithm. From these regions, non-building concrete surfaces such as parking lots, roads etc. are identified and removed leaving only building candidate regions remaining. Each of the aforementioned processes are now described in detail in the subsequent subsections.

4.1 Pixel Band Color Segmentation

The vegetation detection algorithm relies on the following proposed color segmentation technique. Let the input aerial image be defined as a matrix ***I***, containing R rows and C columns and let the indices i,j and k address a pixel intensity I(i,j,k) at row i, column j and channel k where $i \in \{1,2,...R\}$, $j \in \{1,2...C\}$ and $k \in \{r,g,b\}$ (for the red, green and blue channels respectively). First, each color channel (red, green and blue) of the input aerial image ***I*** is divided up into 17 subsections each existing as 15 intensity band increments from 0 to 255 (0 to 15, 15 to 30, 30 to 45..., 240 to 255). Pixels which exist in the same color channel, within a single 15 intensity band increment in that color channel, and are connected to one another (in an

8 connected neighborhood) are then assigned a region label. Different connected sets of pixels are given different region labels. Let a single region label existing in the red, blue and green band be defined as l_r, l_g, l_b respectively. Let the set of all region labels in the red, blue and green channels be defined as L_r, L_g, and L_b respectively where $l_r \in L_r$, $l_g \in L_g$, and $l_b \in L_b$. Note - all of the region labels in each color channel are unique, therefore: $L_r \cap L_g = 0$, $L_g \cap L_b = 0$, and $L_r \cap L_b = 0$. Let the S( ) function return the size (in pixels) of a given region. A region image ***P*** is created, which has three channels where the intensities in each channel corresponds to a region label in that channel. Therefore, the region label for a pixel located in i,j in channel k is P(i,j,k).

Typically, even a single roof panel on a small house can be represented by a single region of at least 2.25 m^2 or 100 connected pixels at 15cm pixel resolution. Therefore, regions having less than 100 connected pixels are removed (their region label set to '0' to denote clutter). It should be noted at this point that this color segmentation technique is custom tailored for object detection in aerial imagery. The removed regions are not of value and often contain clutter such as small cars, industrial crates, or other objects which are typically not buildings, roads, pavement or vegetation. Let all clutter regions for a given color band be denoted by Z_k where $k \in \{r,g,b\}$; then, Z_k is formulated as follows:

$$Z_k = \{ l_k \mid S(l_k) < 100 \} \quad k \in \{r,g,b\} \tag{8}$$

Then, each channel in region image ***P*** undergoes a morphological closing operation (image dilation followed by erosion). A consolidated, single channel image ***O*** is then created by searching each pixel P(i,j,k) in each channel and setting the output for that pixel location *O(i,j)*

equal to the largest sized region label S(P(i,j,k)) at location i,j for all channels (∀k). The output is formulated as follows:

$$O(i,j) = \max\Big(S\big(p(i,j,r)\big), S\big(p(i,j,g)\big), S\big(p(i,j,b)\big)\Big) \tag{9}$$

Sometimes a single region in a single channel ***P*** is broken up into multiple smaller, disconnected regions in ***O*** when all the channels of ***P*** are combined and only the largest regions represented in ***O***. Therefore all connected pixels having the same label are relabeled to have unique labels (thus re-labeling two disconnected regions of the previous same label to now having two different labels). In doing this, some smaller regions are then created and therefore all labels are searched and ones having less than 100 pixels in size are again removed. Finally, the consolidated image is processed by a morphological closing operation. Let l_o represent a region in the final processed, consolidated single channel image ***O*** where $l_o \in L_o$ and L_o is the set of all regions in the image ***O***. Note, if pixel location i,j belongs to region l_o then $O(i,j) = l_o$.

Figure 20 (a)

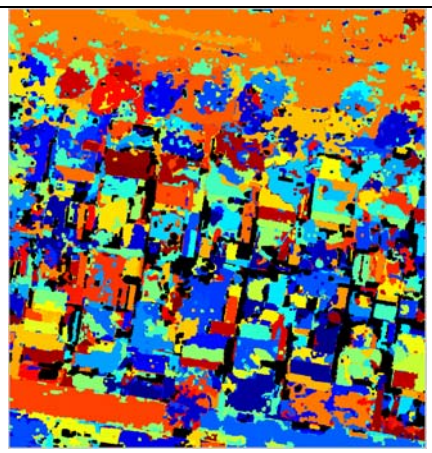


Figure 20 (b)

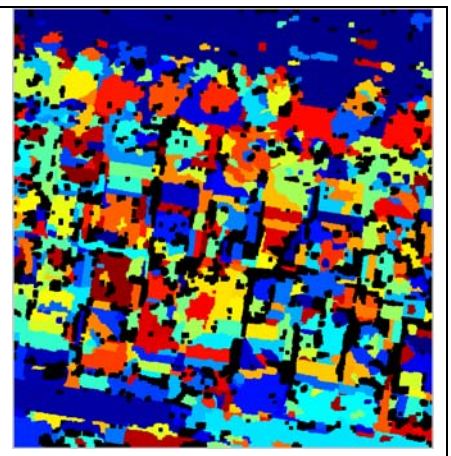


Figure 20 (c)

Figure 20: Input image and consolidated Output image.

(a) Shows the subsection of the original aerial image. **(b)** Shows the largest region labels of the three channels. **(c)** Shows the final output after region labels with less than 100 pixels are removed and the image is processed by a morphological closing operator.

Typically regions of interest, whether they be shadows, vegetation or building are connected with more than a 100 pixels (at 15 cm pixel resolution) and can be best represented by a single region label from one of the color channels. This segmentation approach shows the largest regions of the three bands to be represented in the final output as it is desired to have the fewest, yet most accurate segmented regions in the output. This color segmentation technique has two appealing advantages: (1) there are no subjective parameters to vary during execution (the pixel band width is fixed for different data sets and the small region removal can be recalculated for different resolutions); and (2) the algorithm executes relatively quickly. All of the above processes culminate in the proposed color segmentation technique which is depicted in the block diagram in Figure 21.

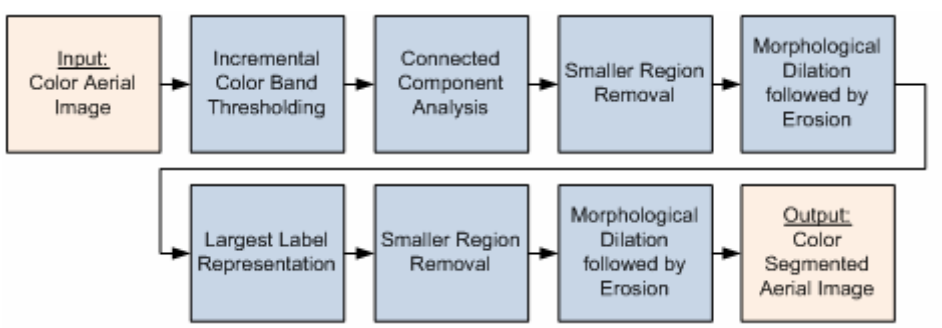


Figure 21: Pixel Band Color Segmentation Block Diagram

4.2 Vegetation Identification

The next phase of the algorithm is to identify and remove vegetation. Vegetation is identified by Otsu thresholding [24] a first-order color invariant [7] and making sure a region, identified via the aforementioned color segmentation algorithm, contains at least 60% identified

vegetation pixels. Color invariants, originally proposed by Gevers and Smeulders in [7] are a set of color models independent of the viewpoint, surface orientation, illumination direction, illumination intensity, and highlights. Similar to Sirmacek et al. in [34], the color information in the aerial image is extracted using the color index proposed by Unsalan and Boyer in [38]. The following color invariant is defined with the green and blue channels of the color aerial image:

$$\psi_g(i,j)=\frac{4}{\pi}\cdot\arctan\left(\frac{I(i,j,g)-I(i,j,b)}{I(i,j,g)+I(i,j,b)}\right)\quad \forall i,\forall j \quad i\in\{1,...,R\}, j\in\{1,...,C\} \tag{10}$$

The above defined color invariant image is then Otsu thresholded and all pixels having values in the image falling above that threshold T_g are marked as vegetation candidates *V(i,j)*. This produces a binary image with white/true pixels corresponding to vegetation candidates and black/off pixels corresponding to non-vegetation candidates:

$$\left[V(i,j)=1\middle|\psi_g(i,j)>T_g\right]\wedge\left[V(i,j)=0\middle|\psi_g(i,j)<=T_g\right]\quad \forall i,\forall j \quad i\in\{1,...,R\}, j\in\{1,...,C\} \tag{11}$$

This method will produce single, isolated pixels in places that do not actually correspond to vegetation. Therefore, the color invariant image is processed by a morphological closing operation (image dilation followed by erosion) followed by an opening operation (image erosion followed by dilation). However, even after this processing, still small groups of pixels will sometimes exist where vegetation does not. However, these small groups of false positives do not compose the majority of the color segmented region in which they exist in. Let the variable l_{vc} be the set of all vegetation candidates V(i,j) = 1 (after morphological opening and closing operators) which exist in the color segmented region l_o.

$$l_{vc}=\left\{V(i,j)\middle|\left[O(i,j)=l_o\right]\wedge\left[V(i,j)=1\right]\right\} \tag{12}$$

Then the total number of vegetation candidate pixels belonging to region l_o would be $S(l_{vc})$ and the percentage of vegetation candidate pixels occupying a given region would be $S(l_{vc})/S(l_o)$. Therefore, a given region l_o and correspondingly all its member pixels are classified as a vegetation region l_v provided that at least 60% of the pixels in the region l_o are classified as vegetation candidates. Let the set of all vegetation regions be denoted as L_v where $l_v \in L_v$.

$$L_v = \left\{ l_o \rightarrow l_v \left| \frac{S(l_{vc})}{S(l_o)} > 0.6 \right. \right\} \tag{13}$$

The steps of the vegetation detection algorithm are depicted in Figure 22.

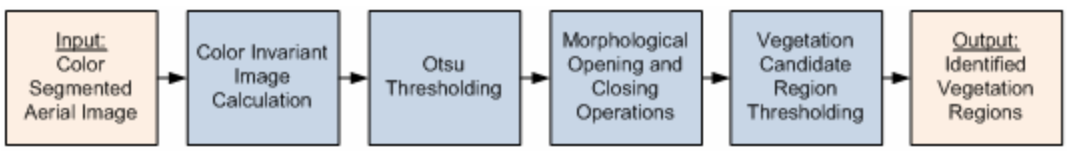


Figure 22: Vegetation Detection Block Diagram

Plots at several stages of the vegetation detection algorithm are shown in Figure 23.

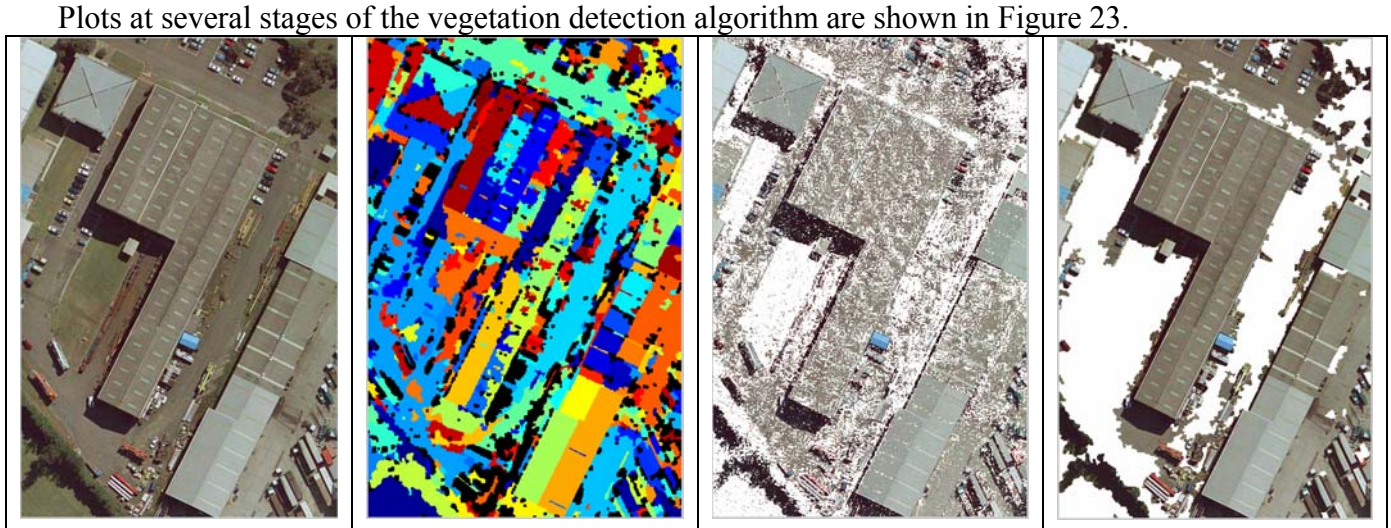


Figure 23 (a) | Figure 23 (b) | Figure 23 (c) | Figure 23 (d)

Figure 23: Output images at various stages of the vegetation detection algorithm.

Subfigure **(a)** shows the original aerial input image, **(b)** shows the final color segmented version of that image, **(c)** shows the output after Otsu thresholding the green color invariant

where white pixels are marked as vegetation candidates; **(d)** shows the final vegetation regions after morphological opening and closing and thresholding the regions for sufficient amount of vegetation candidates.

4.3 Shadow Region Identification

Shadows are identified through the use of a modified color invariant. Sirmacek et. al. used an equation similar to (10) except they took the difference between the blue and green color channels over the sum of those channels to identify shadows. Better success was found with instead using the image constructed by the following equation:

$$\psi_s(i,j) = \frac{4}{\pi} \cdot \arctan\left(\frac{I(i,j,r) - \sqrt{I(i,j,r)^2 + I(i,j,g)^2 + I(i,j,b)^2}}{I(i,j,r) + \sqrt{I(i,j,r)^2 + I(i,j,g)^2 + I(i,j,b)^2}} \right) \quad \forall i, \forall j \quad i \in \{1,\ldots,R\}, j \in \{1,\ldots,C\} \quad (14)$$

This image is then Otsu thresholded and all pixels below the threshold are considered shadow. In Figure 24, Sirmacek et. al's equation is implemented on the middle image (using the difference between the blue and green channels over their sum) and the method from equation (14) is shown on the right where pixels appearing completely white are shadow pixels. In the authors experience, it is rare that a building casts a shadow on another building. Notice that Sirmacek et. al's method winds up labeling pixels on the dark side of the roof as shadows and the method from (14) does not.

		
Figure 24 (a)	Figure 24 (b)	Figure 24 (c)

Figure 24: Shadow detection result comparison.

Original aerial image in 23(a), Sirmacek and Unsalan's method in 23(b) and the proposed method from (14) in 23(c)

4.4 Building/Non-Building Identification and Texture Segmentation

The color segmenting technique in section 4.1 sometimes segments the image with too many segments but the borders of those segments are fairly accurate in outlining the objects they approximate in the image. A texture segmenting technique is used for building identification that has a parameter to control the coarseness of the generated segments. Buildings and non-buildings (other than shadow and vegetation) are detected by measuring the solidity of their regions. The regions are obtained by entropy filtering the input image and then using a watershed segmentation technique on that entropy filtered image.

Matlab's 'entropyfilt' function was used to produce an entropy image. The entropy filter (see Chapter 11 of [8]) produces an entropy image where each pixel is the entropy calculated from a 9x9 neighborhood in the input image I. The formula for the entropy is calculated as follows:

$$e = -\sum_{i=1}^{L} p(z_i) \cdot \log_2\left(p(z_i)\right) \tag{15}$$

Where $p(z_i)$ is the number of occurrences the intensity z_i has in the 9x9 neighborhood. Then, the entropy image I is thresholded (all pixels less than 0.75 times the maximum entropy value in the entropy image are removed). Finally, a watershed segmentation [18] was applied to the distance transformed entropy image. For the purpose of building detection, the 0.75 worked for the desired coarse segmentations and this parameter was not changed when the algorithm was executed across both data sets. This results in the texture segmentation algorithm block diagram depicted in Figure 25.

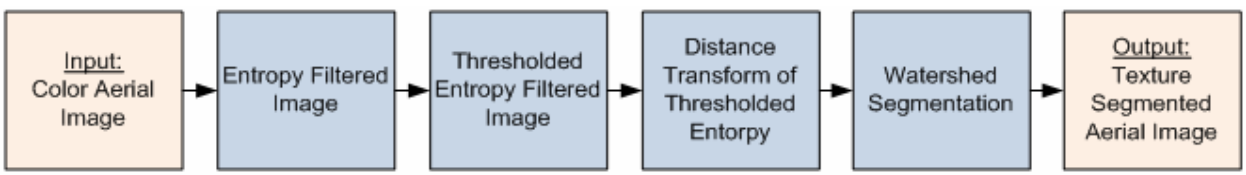


Figure 25: Entropy Filtering / Watershed Texture Segmentation Algorithm Block Diagram

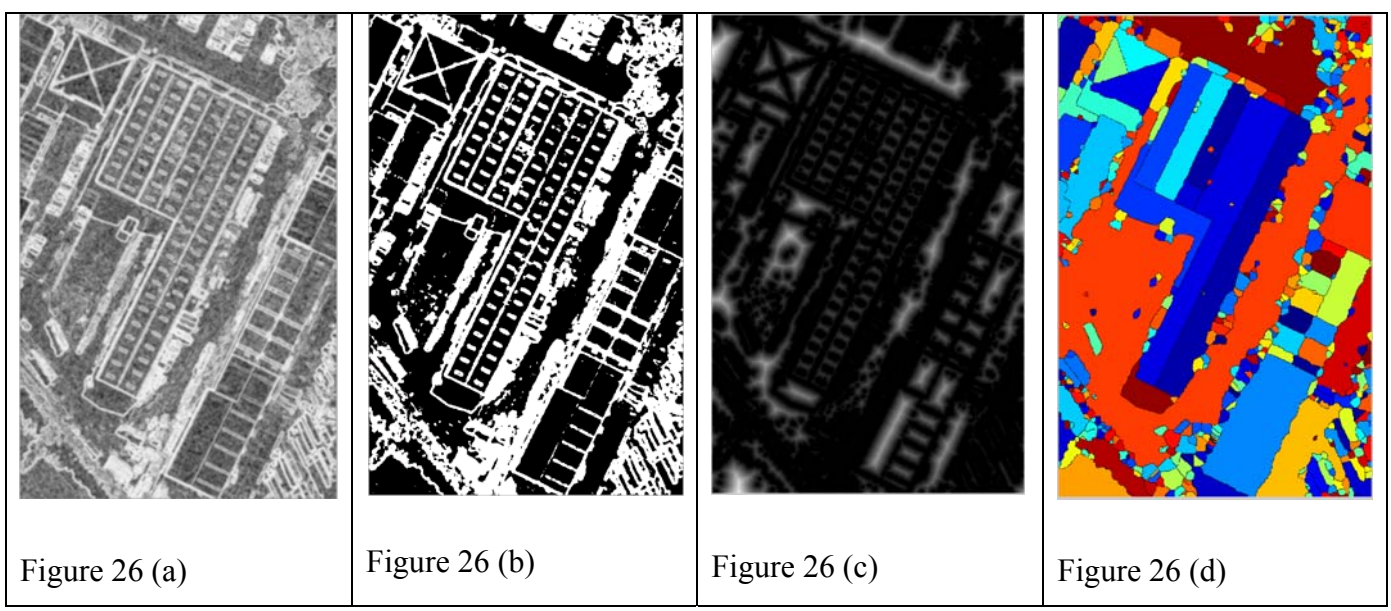


Figure 26: Outputs of the entropy/watershed texture segmentation technique

Subfigure 25(a) is the entropy image, subfigure 25(b) is the thresholded entropy image, subfigure 25(c) is the distance transform of the thresholded entropy image and subfigure 25(d) is the watershed segmentation applied to output 25(c).

A convex hull is defined here as the minimum polygon which completely encompasses a given region. Consider the black region in Figure 27 (a), its area is simply the sum of all the black pixels. The convex hull for that region is then shown in Figure 27 (b) in yellow and the area for that convex hull is in Figure 27 (c) (the sum of all the yellow pixels).

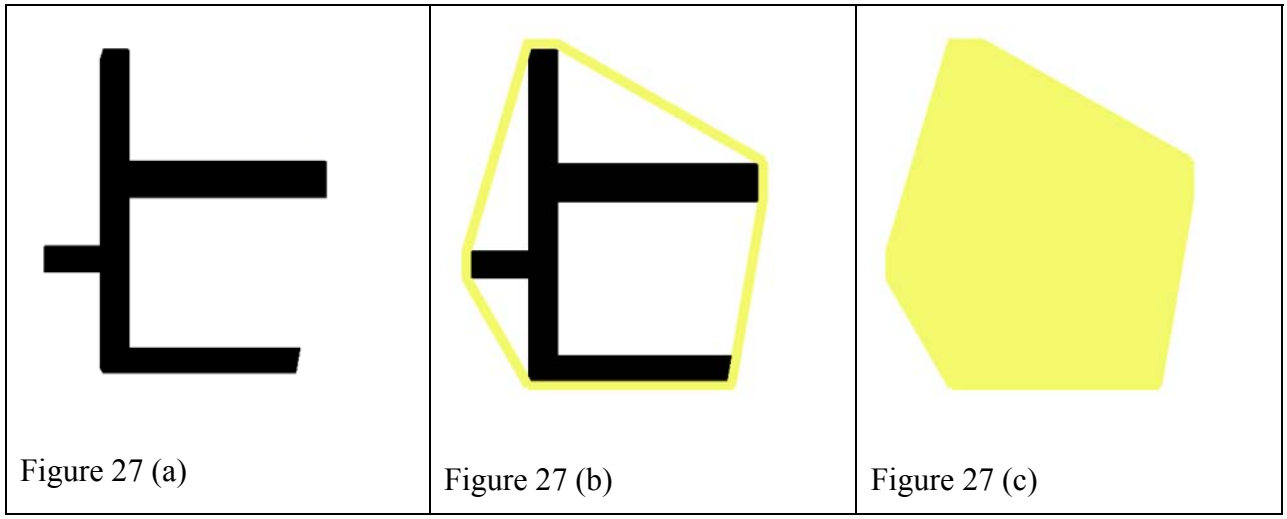


Figure 27 (a) | Figure 27 (b) | Figure 27 (c)

Figure 27: Convex hull example

Subfigure 26(a) shows an arbitrary region, subfigure 26(b) shows a yellow convex hull which bounds the exterior of that region, and figure 26(c) shows the area of that convex hull.

Let a region created by the watershed segmentation technique be denoted as l_w where L_w is the set of all regions created and $l_w \in L_w$. Let the set of all pixels which composes the convex hull (filled in area as shown in Figure 27 (c)) which bounds l_w be denoted as c_w and the size of that convex hull be denoted as $S(c_w)$ (note this size is defined as the area within the convex hull or sum of all yellow pixels in Figure 27 (c)). Let the size of the watershed segmentation region be denoted as $S(l_w)$. Let the solidity for the watershed segmentation region l_w be denoted as D_w. Solidity is then defined as follows.

$$D_r = \frac{S(l_w)}{S(c_w)} \quad D_r \in (0,1] \tag{16}$$

The upper limit of the solidity can be exactly one (the region itself is a convex hull) and the lower limit can get infinitely close to 0. Therefore, the more convex the region exterior is, the higher its solidity. For example, a region that is a triangle, square, pentagon, etc. will have a high solidity as its exterior contour is a convex hull. However a region shaped like the letter 'E' or 'W' will have a low solidity as there will be lots of empty space between the bounding convex hull and the region's exterior contour.

Notice that for even in the largest building in the center of Figure 26, most of the building is broken into regions having a high solidity. However, the largest segment surrounding the center building in Figure 26 (appearing in orange) is not approximated very well by a convex hull and has a low solidity. It was noticed roads and surrounding landscapes are typically approximated by a single region with this texture segmentation approach and often have low solidity values. Therefore all regions having a solidity above 0.7 are labeled as building and everything equal to or below as non-building. Because the entropy is a pseudo measure of texture, this approach implements the assumption that the roof sections of a building can be approximated with convex hull regions. Note that this does not necessarily assume the entire building roof has to have a high solidity, just the various textures or sections composing its roof. However, if a single building exists as an 'L' shape and has a single texture across its entire roof, then this approach would most likely erroneously classify such a building as non-building.

A block diagram showing all of the aforementioned components of the proposed vegetation identification and building detection system are shown in Figure 28.

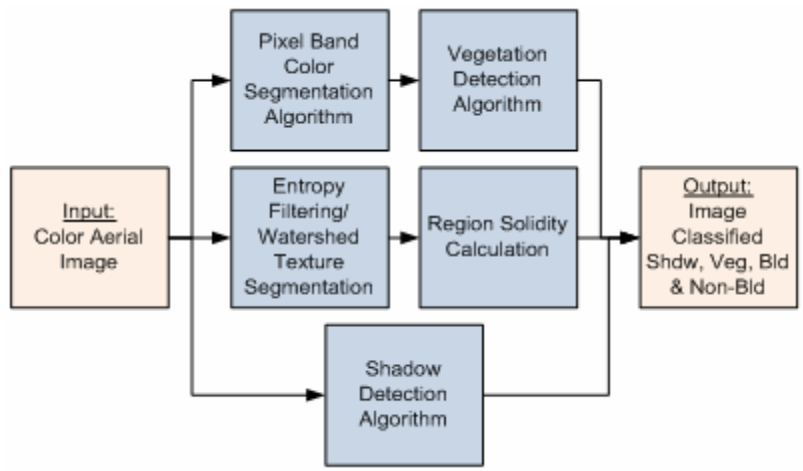


Figure 28: Building Detection and Image Classification Block Diagram

CHAPTER FIVE: REGISTRATION OF LIDAR TO AERIAL IMAGERY FOR IMPROVED CLASSIFICATION

5.1 Registration Introduction and Building Masks

The proposed use of phase correlation for the automatic registration of LiDAR data to a single nadir aerial imagery is implemented. First, buildings existent in the LiDAR data and aerial imagery are detected. Then the LiDAR data is interpolated to fixed point spacings, producing both a pixel to irregular point mapping and a binary building mask. In the pixel to point mapping the irregular points belonging to the interpolated pixels are cached creating a way to reference the irregular points when addressing the interpolated pixel. In the building mask the bright pixels correspond to buildings and dark pixels to everything else. A binary building mask is also produced from buildings detected in a corresponding aerial image. The Fourier transforms and the log polar Fourier transforms of both building binary masks are computed. Phase components are correlated and their peaks reveal the translation, rotation and scaling geometric transformation parameters.

In order for a building detection and/or reconstruction algorithm to make use of features from multiple data sources, those sources must be registered (projected onto one another or describable by a single coordinate system). If both aerial image and LiDAR data exist for a given scene, then it behooves one to take advantage of the information existent in both sources of data for more accurate building detection and/or reconstruction. Several building detection/reconstruction approaches existent in the literature [10], [1], [43], [30], and [41] treat the registration of the two data sources as a pre-processing technique and proceed to manually

register the images together. With the plethora of image registration methods published [47], several of which are automated, it is possible to instead automatically carry out this task.

The proposed registration scheme is implemented in the following steps: (1) detect buildings in the LiDAR data; (2) interpolate the LiDAR data to fixed point spacings; (3) detect buildings in the aerial imagery; (4) register the detected buildings in the LiDAR data to the aerial image. All of the above steps are illustrated in the block diagram representation in Figure 29:

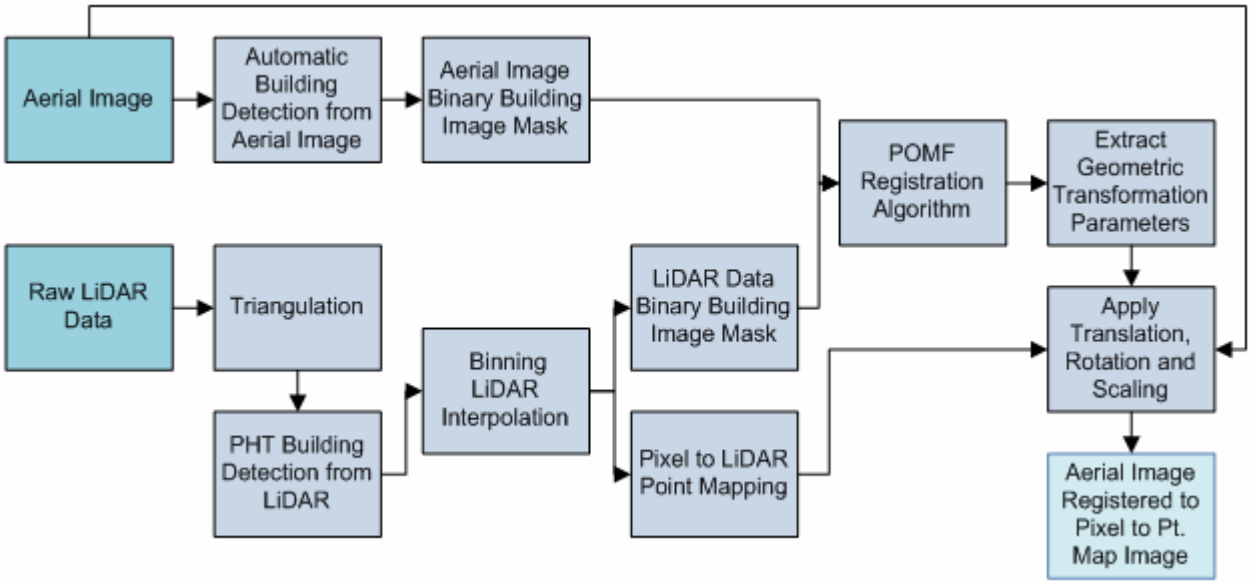


Figure 29: Registration Algorithm Block Diagram

5.2 LiDAR Interpolation

The phase correlation algorithm extracts the geometric transformation parameters from the Fourier transform space. Therefore, in order to use the algorithm, a two-dimensional Discrete Fourier transform of both the aerial image and the LiDAR data must be taken. The irregular LiDAR data must therefore be interpolated to fixed point spacing. There are several documented disadvantages to interpolating the data to fixed point spacing: ambiguities are

introduced when multiple irregular points are reduced to a single, rasterized point which inaccurately attempts to describe both discrete ground and non-ground points [40]. The registration algorithm will be used to relate information extracted from the aerial image to the irregular LiDAR data to enhance the image classification and building detection accuracy.

Cho et. al. in [3] proposed a Pseudo Grid approach where a grid was overlaid on top of irregular LiDAR data. The points within a given grid space were interpolated and thus each grid made up a pixel in what became a range image. With this approach, the authors were able to keep track of the irregular points used for interpolation of a given rasterized point. Conceptually, a grid is overlaid on top of the LiDAR points, as shown in Figure 30.

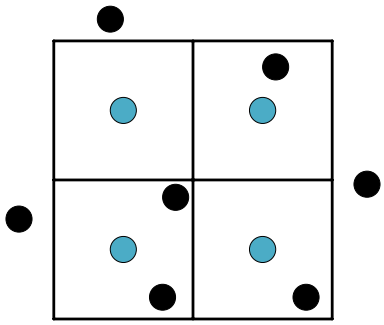


Figure 30: A top down 2D view of the conceptual grid laid upon the LiDAR

The light dot is a grid cell center, the dark dot is an irregular LiDAR point, the solid line is a grid cell border.

Each raw point, bounded by a given grid cell, is stored in that cell's data structure. Furthermore, the center of the square grid cell is also stored in that cell's data structure. From this interpolation scheme, the irregular data is therefore closely tied to the interpolated data which will be related to the aerial imagery. The information extracted from the aerial imagery can therefore be applied to the irregular LiDAR.

5.3 Phase Correlation Registration

It is impossible to develop a registration technique which will be optimal for all types of geometric transformations for all sources of images. Instead, the method must be optimized for the specific, required task. The two sources of data, an aerial image and LiDAR, are data captured from two different sensors. Furthermore, the overhead aerial image and LiDAR data are assumed to only differ via rotation, translation and scaling geometric transformations. By correlating the phases of the 2D Fourier transforms and phases of the log polar 2D Fourier transforms of the target and reference images, parameters for translation, rotation and scaling geometric transformations can be extracted [27].

The phase correlation method is an automatic, area based image registration algorithm; meaning that the algorithm operates on image intensity instead of control points or features such as corners, lines, etc. The aerial image intensity is obviously significantly different from the interpolated LiDAR range image intensity. Because the two data are captured from two different sources, the technique cannot be applied without some preprocessing. The PHT building detection from LiDAR method identifies which points belong to buildings and which do not. The interpolation method, described in section 5.2, produces two outputs - a pixel to irregular point mapping and a binary building image. In the binary building image, the bright pixels correspond to building and dark pixels correspond to everything else. Then a binary building image from the output of the automatic building detection from single, nadir aerial image approach is produced, again where bright pixels correspond to buildings and dark pixels to everything else. The phases of the Fourier transforms of the two binary images and the phases of the log polar Fourier transforms of the binary images are then correlated and the geometric

transformation parameters describing how the two binary images (as well as range and aerial images) differ from one another are then extracted. The peak of the inverse of the phase correlation of the log polar Fourier transforms of the binary images reveals the scaling and rotation parameters while the phase correlation of the Fourier transform of the binary images reveals the translation parameters.

Consider the reference image r and the transformed image s differing by translation (x_0,y_0):

$$r(x,y) = s(x - x_0, y - y_0) \tag{17}$$

The phases of the Fourier transforms of both images (r and s) are as follows:

$$R^*(u,v) = F\left\{r\left(x,y\right)\right\} = \left|R^*(u,v)\right| \cdot e^{j\phi_r(u,v)} \Rightarrow e^{-j\phi_r(u,v)} = \frac{R^*(u,v)}{\left|R^*(u,v)\right|} \quad (18)$$

$$S(u,v) = F\left\{s\left(x,y\right)\right\} = \left|S(u,v)\right| \cdot e^{j\phi_s(u,v)} \Rightarrow e^{j\phi_s(u,v)} = \frac{S(u,v)}{\left|S(u,v)\right|} \quad (18) \tag{18}$$

Multiplication of the phases then yields:

$$\frac{S(u,v) \cdot R^*(u,v)}{\left|S(u,v) \cdot R^*(u,v)\right|} = e^{j\phi_s(u,v)} \cdot e^{-j\phi_r(u,v)} = e^{j\cdot\left[\phi_s(u,v) - \phi_r(u,v)\right]} \tag{19}$$

The above is simplified and its inverse Fourier transform taken:

$$e^{j\cdot\left[\phi_s(u,v) - \phi_r(u,v)\right]} = e^{j\cdot\left(u\cdot x_0 + v\cdot y_0\right)} \Rightarrow F^{-1}\left\{e^{j\cdot\left(u\cdot x_0 + v\cdot y_0\right)}\right\} = \delta\left(x_0, y_0\right) \tag{20}$$

The above Fourier Shift Property (both the forward and reverse versions) have been proved in appendix C.

Now consider a reference image r and a transformed image s differing by rotation θ_0 and translation (x_0,y_0):

$$r(x,y) = s(x \cdot \cos\theta_0 + y \cdot \sin\theta_0 - x_0, -x \cdot \sin\theta_0 + y \cdot \cos\theta_0 - y_0) \tag{21}$$

Taking the Fourier Transform of (21) yieds:

$$R(u,v) = e^{j\cdot(u\cdot x_0 + v\cdot y_0)} \cdot S(u \cdot \cos\theta_0 + v \cdot \sin\theta_0, -u \cdot \sin\theta_0 + v \cdot \cos\theta_0) \tag{22}$$

The magnitude in (48) is represented with polar coordinates:

$$\left|R(\rho,\theta)\right| = \left|S(\rho,\theta-\theta_0)\right| \tag{23}$$

Where:

$$\rho = \sqrt{x^2 + y^2} \quad \theta = \tan^{-1}\left(y/x\right) \tag{24}$$

Observe that in (49) the rotation is now represented as a linear shift and can be extracted with the Fourier shift property.

Now consider images r and s differing by rotation θ_0, translation (x_0,y_0) and scaling a:

$$r(x,y) = s\left[\frac{1}{a}\cdot\left(x \cdot \cos\theta_0 + y \cdot \sin\theta_0\right) - x_0, \frac{1}{a}\cdot\left(-x \cdot \sin\theta_0 + y \cdot \cos\theta_0\right) - y_0\right] \tag{25}$$

Taking the Fourier transform of (50):

$$R(u,v) = e^{j\cdot(u\cdot x_0 + v\cdot y_0)} \cdot S\left[\frac{1}{a}\cdot\left(u \cdot \cos\theta_0 + v \cdot \sin\theta_0\right), \frac{1}{a}\cdot\left(-u \cdot \sin\theta_0 + v \cdot \cos\theta_0\right)\right] \tag{26}$$

Representing the Fourier Transform with polar coordinates:

$$\left|R(\rho,\theta)\right| = \left|S\left(\rho/a, \theta-\theta_0\right)\right| \tag{27}$$

Where:

$$\rho = \sqrt{\left(\frac{x}{a}\right)^2 + \left(\frac{y}{a}\right)^2} = \frac{1}{a}\cdot\sqrt{x^2+y^2} \quad \theta = \tan^{-1}\left[\left(y/a\right)/\left(x/a\right)\right] = \tan^{-1}\left(y/x\right) \tag{28}$$

Taking the log of the ρ-axis in (27):

$$\left|R(\log(\rho),\theta)\right| = \left|S\left[\log(p)-\log(a),\theta-\theta_0\right]\right| \tag{29}$$

Observe in (29) that both the rotation and scaling have been reduced to linear translations. Again, using the Fourier shift property, the scaling and rotation parameters can be extracted.

The automatic phase correlation registration algorithm, described by equations (17) - (29), are implemented as shown in the block diagram in Figure 31.

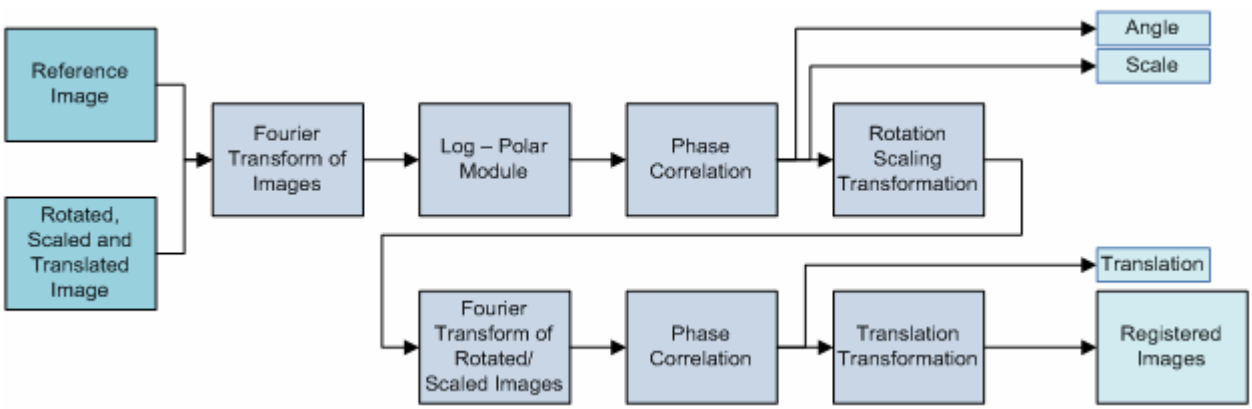


Figure 31: Automatic Phase Correlation Image Registration Algorithm

Phase correlation is limited for scaling translations from about 50% to 200% [2]. If the target and reference image (or for this application case the two building masks) differ by scaling outside of the aforementioned range, then the phase correlation registration technique will fail. Pixel spacing in the aerial image is typically known a-priori and provided in the data set's meta data. The LiDAR point density, denoted now as p_d, can be calculated, if it is not also known a-priori. By knowing both the pixel spacing and LiDAR point density, then the aerial image can be brought within +/- 10% the same scaling as the interpolated LiDAR data. The LiDAR building mask is interpolated to an R_l by C_l image where R_l and C_l are calculated as follows.

$$R_l = \frac{y_{\max} - y_{\min}}{\eta}, C_l = \frac{x_{\max} - x_{\min}}{\eta} \tag{30}$$

Where eta is the grid cell spacing which will have approximately one LiDAR point per grid cell if calculated as proposed by [19].

$$\eta = \frac{1}{\sqrt{p_d}} \tag{31}$$

The LiDAR building mask is brought within +/- 10% of the same scaling as the aerial image building mask because the exact scaling is difficult to compute due to the fact that the LiDAR data point spacing is sometimes not completely uniform. When the plane flies a given path, the LiDAR data procured on that path is referred to as a strip. Sometimes the strips will overlap creating denser point densities in some areas than others.

5.4 Combining LiDAR and Aerial Image Classifiers

After the aerial imagery has been registered to the LiDAR data, it is then possible to use features from both sources of data for improved classification. The building detection from LiDAR algorithm sometimes has trouble mislabeling grass patches surrounded by vegetation or steep hills as building. The proposed vegetation detection from aerial imagery approach however wound up being highly correct when labeling pixels as vegetation. Therefore, the vegetation classification from the aerial image is used to remove false positives from the building detection from LiDAR data.

In order for the vegetation classification from the aerial image to be used with the LiDAR data, it has to be registered to the LiDAR data (i.e. the image has to be translated, rotated and scaled). The output of the building detection from aerial imagery algorithm contains a

classification image where the pixel intensities correspond to different classes of objects as classified by the algorithm. When undergoing the geometric transformations of scaling and rotation, sometimes those intensities can be slightly altered from their original values when the image is re-sampled after being scaled and/or rotated. Therefore, before using the vegetation classification image, it is filtered via a morphological opening and then closing operation to remove noise introduced by the aforementioned interpolation.

The vegetation classification from aerial image approach is not perfect and some buildings are labeled as vegetation as a result. When combining the vegetation detection approach with the classifications made by the building detection from LiDAR data algorithm, it is therefore desired to avoid if possible having the vegetation approach mislabel buildings as vegetation. Therefore the combination of the sources should increase building detection from LiDAR algorithm's correctness by correctly labeling vegetation that the LiDAR algorithm mistakenly labeled as building. Furthermore, most trees are already identified in the non-building class. However the ground class from the building detection from LiDAR algorithm ambiguously contains both pavement as well as grass vegetation. In order to better classify the image, it is desired to have the ground be exclusively pavement and dirt and the vegetation consist of grass. In order to intelligently combine these approaches, the following feature from the LiDAR data has been used. Consider a non-wall connected set (NWC) in the LiDAR data. At the outer edge of that NWC set will exist pairs of NWT's which are adjacent to WT's belonging to an adjacent wall connected set (WC). Then define the three points which make up that WT as pt_L, pt_M, and pt_H where pt_H is the highest point in elevation, pt_L is the lowest and pt_M's elevation is less than pt_H's elevation and greater than pt_L's elevation. Two scenarios then arise: the NWT in the NWC set can either exist at the bottom of the adjacent WT or at the top. A 'low

side' adjacency is defined as when a NWT is adjacent to the triangle side of the WT which contains pt_L and pt_M of the WT as depicted in Figure 32. A high side adjacency is then defined as when a NWT is adjacent to the triangle side of the WT which contains pt_H and pt_M as depicted in Figure 33.

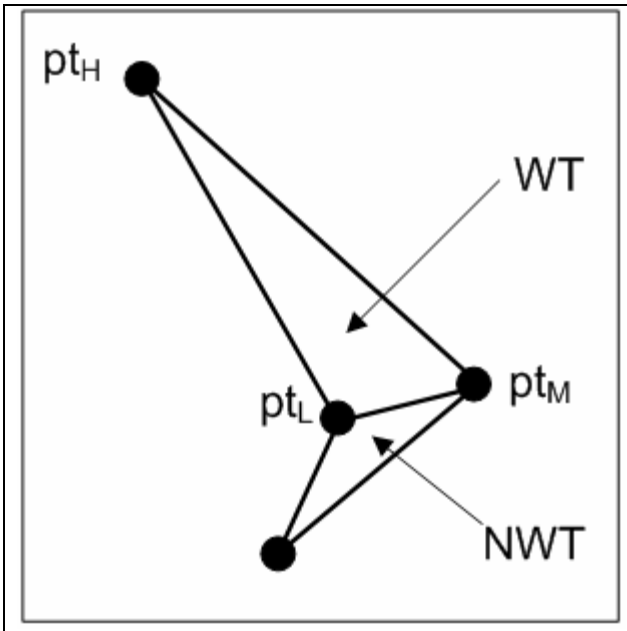


Figure 32: Low Side Adjacency

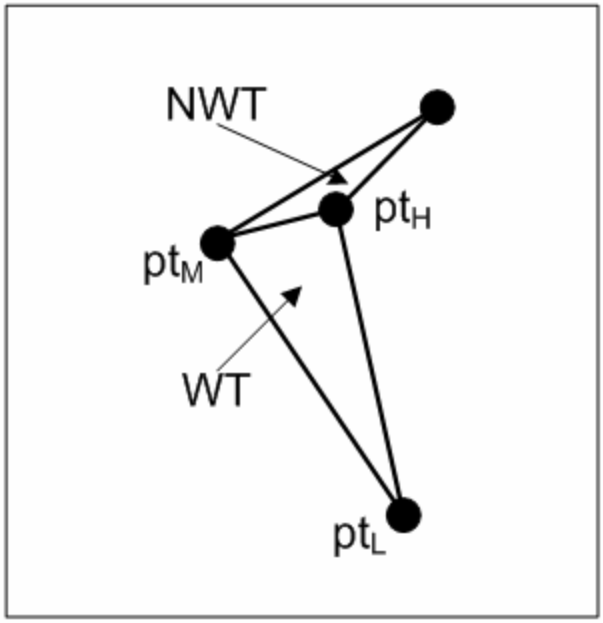


Figure 33: High Side Adjacency

Then, points belonging to a NWC set were relabeled if identified as vegetation by the vegetation detection from aerial imagery algorithm and if their low side adjacencies in the triangulated LiDAR data made up at least 20% of the total adjacencies. This empirically determined parameter was held constant across the algorithm's execution on both data sets. Tops of buildings will actually have a high number of high side adjacencies as the NWC set describing a building's roof literally exists above most of the wall triangles which describe that building's side. On the other hand, grass and ground will exist at the bottom of neighboring connected sets and have a higher percentage of low side adjacencies. Combining the features in this proposed manner yielded a slight increase in the algorithm's per point correctness while resulting in only a

minute decrease in the completeness. Furthermore, in the per point accuracy plots it was noticed large portions of false positive vegetation grass patches were removed and correctly labeled as non-building. Finally, a new class was incorporated exclusively identifying vegetation grass.

CHAPTER SIX: DATA SETS AND EVALUATION

6.1 Fairfield Data Set Description

The proposed algorithm has been tested on two different data sets. The first data set, depicting terrain from Fairfield, Australia, will be referred to as "the Fairfield data set". This data set includes both LiDAR data, depicted in Figure 35, as well as aerial imagery, depicted in Figure 34. The LiDAR data was procured using an Optech ALTM 3025 laser scanner. The ALTM3025 system has a pulse rate of 25,000 points per second, is capable of recording first and last return pulse information as well as returned pulse intensity and has a vertical accuracy of 15 cm. The point density is one point per 1.2 m^2. The Fairfield data set depicts 2x2km^2 area, covering a suburban area in the southwest whereas the northeast mostly contains industrial buildings. The RGB aerial color image, accompanying the LiDAR data, has 15 cm resolution. The RGB aerial color image covers a slightly smaller portion of the terrain than the LiDAR data. All of the buildings in the Fairfield data set have been manually identified and marked as black polygons in Figure 36.

Figure 34 - Fairfield Data Set Aerial Image

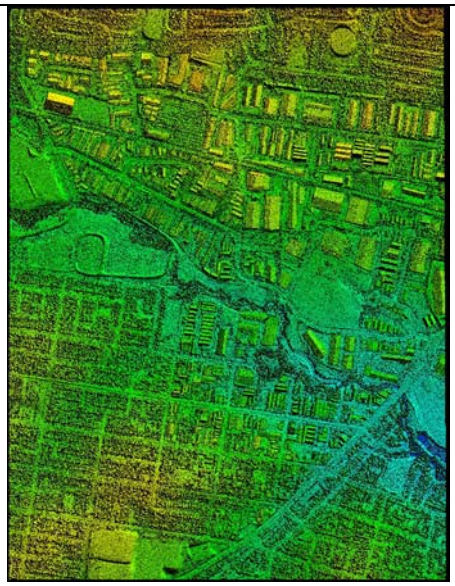


Figure 35 - Fairfield Data Set LiDAR

Figure 36 - Buildings Manually Identified in the Fairfield Data Set

Figure 37 - Buildings Manually Identified in the Anchorage Data Set

6.2 Anchorage Data Set Description

The second data set, depicting terrain from Anchorage, Alaska, will be referred to as "the Anchorage data set". The data set includes both LiDAR data, depicted in Figure 39, as well as aerial imagery, depicted in Figure 38. Unlike the Fairfield data set, this data set's LiDAR only consists of first return pulses and no last return pulses or returned intensity. Also, again unlike the Fairfield, this data set's aerial imagery completely coincides with the LiDAR coverage; whereas the Fairfield's LiDAR coverage contained more area than the aerial imagery. The aerial image and LiDAR coverage depict a 1.67x1.67km^2 area, covering a mostly suburban area in the south half of the data set and commercial buildings existing in the north half. The LiDAR data set, as provided by Aerometric, has been interpolated on a grid at one point per 1.5 m^2. The LiDAR data has a vertical accuracy of 20 cm. The accompanying RGB aerial color image has a 15 cm pixel resolution and a planned horizontal accuracy of 60 cm. The buildings in the

Anchorage data set have been manually identified and marked off as black polygons in Figure 37.

Figure 38 - Anchorage Data Set Aerial Image

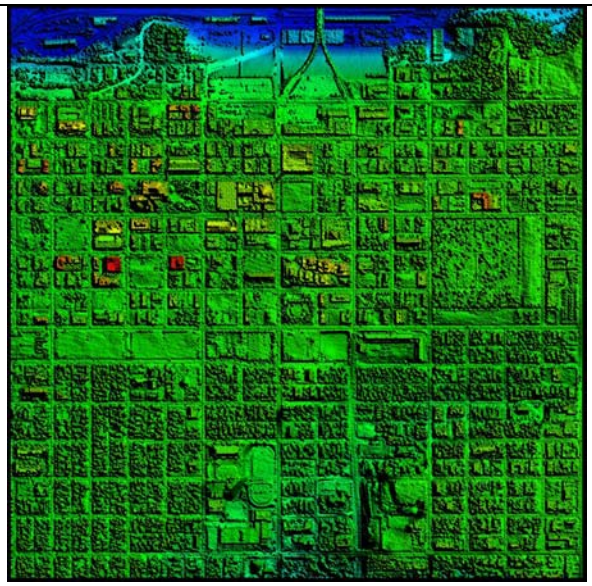


Figure 39 - Anchorage Data Set LiDAR

6.3 Similarities and Differences in Data Sets

In both data sets, the LiDAR data and the aerial image depict the terrain from a nadir/top down perspective and the accompanying aerial images have been ortho rectified. It is worth noting that in both data sets, the LiDAR data were captured at a later time than the aerial images and that certain buildings which were under construction may exist in the LiDAR data but not in the aerial images.

The Fairfield data set has different building and terrain characteristics when compared to the Anchorage data set . The industrial portion of the Fairfield data set is not by any means equivalent to the commercial portion of the Anchorage data set. The buildings in the industrial

portion of the Fairfield data set, for the most part, lie relatively low and have a large base, whereas some of the commercial buildings in the Anchorage data set rise fairly high. In the Anchorage data set there are two bridges (one to the north and one to the south) and a steep hill (in the northwest). In the Fairfield data set, there are no bridges but there is a trench in the middle of the data set filled with dense forestry. There are also a great deal of sheds of varying sizes and some pools near the residential homes in the southern part of the Fairfield data set. There were no pools in the Anchorage data set and fewer homes have sheds. Because the sun is lower to the ground when the Anchorage data set was captured, the shadows in the Anchorage are longer than the Fairfield data set. Furthermore, the Anchorage data set to the North East contains a densely populated urban area filled with a great deal of concrete and little vegetation.

In order to evaluate the accuracy of the proposed algorithm, buildings in the LiDAR data were manually identified and used as the ground truth. With the help of the Quick Terrain Modeler software, where the full version available for trial period can be downloaded at [26], building points in both data sets were manually extracted. The Quick Terrain Modeler software is capable of working with the raw, irregular LiDAR data, no interpolation was necessary. Because it was necessary to look at a point's approximate location and context in the aerial imagery as well as the LiDAR data in order to accurately determine if the point did indeed belong to a given building, only points with aerial image coverage were extracted. In the Fairfield data set 2,189 buildings were identified and for the Anchorage data set 1302 buildings were identified. Points belonging to different buildings were given different, unique labels. This set of manually labeled points will be referred to as the reference set. The algorithm also marks all points belonging to different buildings with individual building labels. From here on out this will be referred to as the automatic set. Due to several forms of noise in the LiDAR data, the

reference set may not be 100% accurate. Note two different reference sets were developed for both the aerial imagery and the LiDAR data for each data set.

6.4 Per Point Completeness and Correctness

For evaluating the automatically extracted buildings, using a reference data set, two performance metrics are pertinent - the completeness and the correctness of the results [9]. Completeness and correctness are defined to exist on a point level and on a building level. On a point level, the completeness represents the percentage of points belonging to buildings that were correctly detected by the building detection from LiDAR data algorithm and the correctness represents the percentage of points labeled by the algorithm as building that actually correspond to building points. Completeness and correctness, on the point level, are also used in [16], [29], [30], and [41].

Consider the following three conditions for points in the LiDAR data:

(i) Point p_i has aerial image coverage

(ii) The same point p_i is identified as belonging to a building in the reference set

(iii) The same point p_i is identified as belonging to a building in the automatic set

Let TPp be the binary set which identifies all true positive points where TPp $=\{tpp_i \rightarrow [0,1] | i \in [1,N]\}$. If a point p_i satisfies conditions (i), (ii) and (iii), then tpp_i takes on a value of 1, indicating point p_i has been successfully detected by the algorithm; otherwise tpp_i is 0. Let FPp be the binary set which identifies of all false positive points where FPp $=\{fpp_i \rightarrow [0,1] | i \in [1,N]\}$. If a point pi satisfies conditions (i) and (iii) and fails (ii), then fpp_i takes

on a value of 1, indicating that the algorithm has labeled the point p_i as belonging to a building when in fact that point does not belong to a building; otherwise fpp_i is 0. Finally, let FNp be the binary set that identifies all false negative points where FNp $=\{fnp_i \rightarrow [0,1] | i \in [1,N]\}$. If a point satisfies conditions (i) and (ii) and fails (iii), then fnp_i takes on a value of 1, indicating that the algorithm has not labeled point p_i as belonging to a building when in fact that point does belong to a building; otherwise fnp_i is 0.

The point-wise level completeness Comppt is defined as the sum of all true positive points divided by the sum of all true positive points and false negative points:

$$Comppt = \frac{\sum_{i=1}^{N} tpp_i}{\sum_{i=1}^{N} tpp_i + \sum_{i=1}^{N} fnp_i} \tag{32}$$

Similarly, the point level correctness Corrpt is defined as the sum of all true positive points divided by the sum of all true positive points and false positive points:

$$Corrpt = \frac{\sum_{i=1}^{N} tpp_i}{\sum_{i=1}^{N} tpp_i + \sum_{i=1}^{N} fpp_i} \tag{33}$$

The completeness and correctness for LiDAR points shown in (32) and (33) can easily be extended to the completeness and correctness for pixels to verify the accuracy of buildings detected from the aerial image. Notice that the true positive, false positive and false negative points must all satisfy condition (i) and then what determines if a point is a true positive, false positive or false negative depends on whether that point satisfies conditions (ii) or (iii). In the aerial image all pixels have aerial image coverage, therefore the analysis remains the same except points are pixels and there are only two conditions instead of three. On a pixel level, the

completeness represents the percentage of pixels belonging to buildings that were correctly detected by the building detection from aerial imagery algorithm and the correctness represents the percentage of pixels labeled by the algorithm as building that actually correspond to building pixels. Let p_i now correspond to the pixel location in a given image: where $i \in [1,N]$ and $N = R \cdot C$

where R is the rows in the image and C is the columns.

The two conditions determining whether a pixel is a true positive, false positive or false negative are then defined as follows.

(i) Pixel p_i is identified as belonging to a building in the reference set

(ii) Pixel p_i is identified as belonging to a building in the automatic set

The completeness and correctness therefore for the pixels in the aerial image is the same as (32) and (33) except what determines a true positive, false positive and false negative is only based on the above two conditions (unlike three conditions for the LiDAR data).

6.5 Building Level Completeness and Correctness

In addition to point/pixel level correctness and point/pixel completion, it is also relevant to look at building level correctness and building level completion. On a building level, the completeness represents the percentage of buildings that were correctly detected by the algorithm and the correctness represents the percentage of buildings that the algorithm identified that actually were in fact buildings that existed in the reference set. The building level completeness and correctness metrics are defined the same way in which they were developed by Rottensteiner et. al. in [29] (and was also used in [30]). Note - the derivation for building level completeness

and correctness is the same for both buildings identified in the LiDAR data automatic and reference sets and buildings identified in the aerial image automatic and reference sets.

Using the same notation in [29], the following quantities relating the reference set labels to the automatic set labels are formulated. For the following discussion, let a region denote a set of LiDAR points or aerial image pixels in either the reference or automatic set that all have the same building label in that set. The set of all regions in the reference set are denoted as L_a and in the automatic set are denoted as L_r. For each pair of regions $l_a \in L_a$ in the automatic set and $l_r \in L_r$ in the reference set, the overlap ratios are calculated: $p_{a \cap r} = n_{a \cap r}/n_a$ and $p_{r \cap a} = n_{a \cap r}/n_r$. Note that $n_{a \cap r}$ is the number of common points/pixels assigned to l_a in the automatic set and to l_r in the reference set; n_a is the total number of points/pixels assigned to region l_a in the automatic set; and n_r is the total number of points/pixels assigned to l_r in the reference set. If the building detection algorithms and the reference sets identify the same points/pixels as belonging to buildings, all overlap percentages for corresponding regions pointing to the same building would be 100% and there would be exactly only one region in the automatic set $l_a \in L_a$ corresponding to each region in the reference set $l_r \in L_r$ and vice versa for each building depicted by both sets. However, because multiple regions in the automatic set partly correspond to a single region in the reference set and vice versa, overlap percentages are calculated to match corresponding regions in the two sets. The function 'overlap' which classifies whether or not regions legitimately overlap is defined as follows:

$$overlap(l_i, l_j) = \begin{cases} true & \forall i,j \,|\, p_{ij} > 50\% \\ false & \forall i,j \,|\, p_{ij} < 50\% \end{cases} \tag{34}$$

Let $i \in \{r,a\}$ and $j \in \{r,a\}$ and $i \neq j$. Note that overlap(l_r, l_a) does not always equal overlap(l_a, l_r), the function is not necessarily symmetric. Overlap ratios for all pairs of regions are calculated and if pairs have regions having both overlap(l_a, l_r) = false and overlap(l_r, l_a) = false, the overlap for that pair is interpreted as spurious and is no longer considered. For a single region $l_a \in L_a$, a subset is obtained $L_{ar} \subset L_r$ where this subset contains all regions in the reference set L_r that correspond to a single region l_a in the automatic set:

$$L_{ar} = \left\{ l_r \in L_r \,\middle|\, \left[overlap(l_r, l_a) = true \right] \vee \left[overlap(l_a, l_r) = true \right] \right\} \tag{35}$$

Similarly, for each region in the reference set $l_r \in L_r$, the subset $L_{ra} \subset L_a$ contains all regions from the automatic set L_a, corresponding to a single region in the reference set l_r:

$$L_{ra} = \left\{ l_a \in L_a \,\middle|\, \left[overlap(l_r, l_a) = true \right] \vee \left[overlap(l_a, l_r) = true \right] \right\} \tag{36}$$

Another way to interpret L_{ra} is it is the set of regions in the automatic data set into which the single region in the reference data l_r is split. The amount of coverage a given region l_a has in the automatic set is denoted by d_a:

$$d_a = \frac{\sum_{l_r \in L_{ar}} n_{a \cap r}}{n_a} \tag{37}$$

For all of the regions l_r in the reference set which correspond to a single region l_a, denoted by $l_r \in L_{ar}$, the number of common pixels those regions l_r have with l_a is found and is

divided by the total number of points/pixels the region l_a, has which is denoted by n_a. If this ratio, denoted as d_a is greater than 50% for region l_a then the building l_a in the automatic set is considered correct. The amount of coverage a given region l_r has in the reference set is denoted by d_r:

$$d_r = \frac{\sum_{l_a \in L_{ra}} n_{a \cap r}}{n_r} \tag{38}$$

For all of the regions l_a in the automatic set which correspond to a single region l_r in the reference set, denoted by $l_a \in L_{ra}$, the number of common points/pixels those regions l_a have with l_r is found and is divided by the total number of points/pixels in the region l_r which is denoted by n_r. If this ratio d_r is greater than 50% for region l_r then the building l_r in the reference set is considered detected.

Let F denote the total number of buildings identified in the automatic set. Let D denote the total number of buildings identified in the reference set. Therefore l_a and l_r respectively exist in the following ranges: $l_a \in \{1,...,F\}$ and . $l_r \in \{1,...,D\}$. Then let TPb1 be the set identifying buildings as detected where, for building r, $tpb1_r$ takes on a value of 1 if $d_r > 50\%$ and 0 if $d_r < 50\%$.

$$TPb1 = \left\{ tpb1_r \rightarrow [0,1] \middle| r \in [1,D] \right\} \tag{39}$$

Note then the set identifying buildings as not detected is the complement of TPb1.

$$FNb = 1 - TPb1 = TPb1^c = \left\{ fnb_r \rightarrow [0,1] \middle| r \in [1,D] \right\} \tag{40}$$

Let TPb2 be the set identifying buildings as correct where, for building a, $tpbl_a$ takes on a value of 1 if $d_a > 50\%$ and 0 if $d_a < 50\%$.

$$TPb2 = \left\{ tpb2_a \rightarrow [0,1] \middle| a \in [1,F] \right\} \tag{41}$$

Note then the set identifying buildings as incorrect is the complement of TPb2

$$FPb = 1 - TPb2 = TPb2^c = \left\{ fpb_a \rightarrow [0,1] \middle| a \in [1,F] \right\} \tag{42}$$

The completeness, calculated at a building level, is then formulated as follows:

$$Compb = \frac{\sum_{r=1}^{D} tpb1_r}{\sum_{r=1}^{D} tpb1_r + \sum_{r=1}^{D} fnb_r} \tag{43}$$

The correctness, calculated at a building level, is then formulated as follows:

$$Corrp = \frac{\sum_{a=1}^{F} tpb2_a}{\sum_{a=1}^{F} tpb2_a + \sum_{a=1}^{F} fpb_a} \tag{44}$$

CHAPTER SEVEN: RESULTS

7.1 Per Point Completeness and Correctness and Building Level Completeness and Correctness for Building Detection from LiDAR Data Algorithm

The point level completeness and point level correctness for the Fairfield data set is 0.928 and 0.802 respectively. This means that 92.8% of the points belonging to buildings, according to the reference set, were correctly detected by the algorithm in the automatic set and that 80.2% of the points labeled as building by the algorithm in the automatic set were actually also labeled building in the reference set. The point level completeness and point level correctness for the Anchorage data set is 0.893 and 0.746 respectively. Again, 89.3% of the points belong to buildings were correctly detected by the algorithm and 74.6% of the points labeled as building by the algorithm actually corresponded to building points.

Figure 40 and Figure 41 present point level results for the Fairfield and Anchorage data sets respectively. In these figures, the red color corresponds to correctly detected points, the orange color corresponds to false positives (points the algorithm classified as belonging to buildings but were in fact identified as not belonging to buildings in the reference set) and the light green color corresponds to false negatives (points that were identified as belonging to buildings in the reference set but classified as not belonging buildings by the algorithm). The blue in these figures corresponds to points not identified as building by both the algorithm and the reference set.

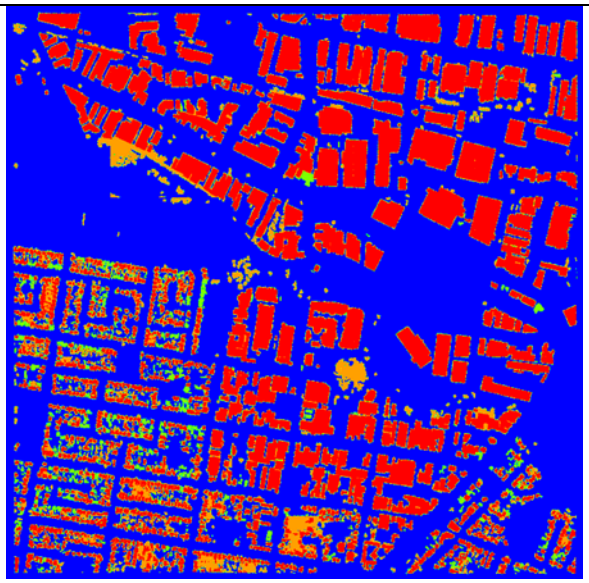


Figure 40 - LiDAR Building Point Accuracy in the Fairfield data set

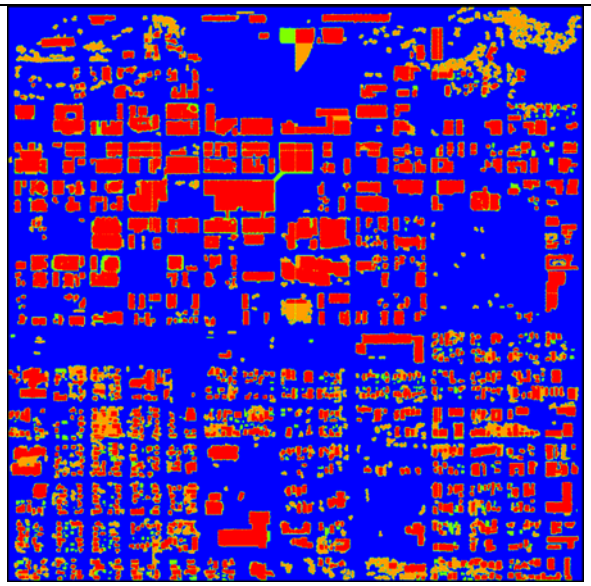


Figure 41 - LiDAR Building Point Accuracy in the Anchorage data set

Observe that in Figure 40 and Figure 41, the PHT algorithm detects nearly all of the industrial buildings and the majority of the residential buildings for both of the data sets.

Figure 42 and Figure 43 present classification results of LiDAR points as building, non-building and ground for the Fairfield and Anchorage data sets respectively. The red points correspond to what the PHT algorithm classified as building, the green points as ground and the orange points as non-building.

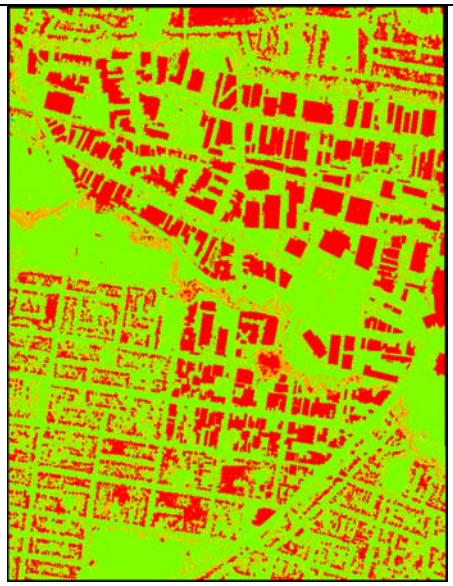


Figure 42 - Point Classification for the Fairfield LiDAR data

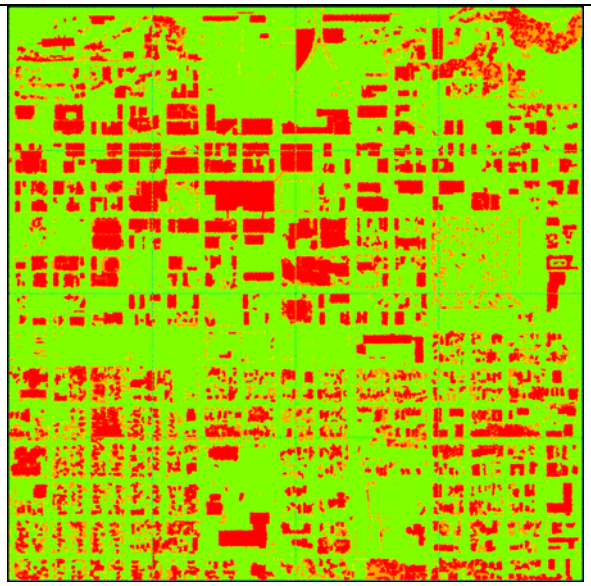


Figure 43 - Point Classification for the Anchorage LiDAR data

Observe in Figure 42 and Figure 43 that the algorithm does a superb job identifying the ground and is able to mostly, correctly classify vegetation for both data sets. There are some cases however where ground between densely built up residential buildings is erroneously labeled as building. A good portion of the vegetation in the trench in the Fairfield data set is correctly classified as vegetation. Almost all of the vegetation in the park that is in the east of the Anchorage data set is correctly classified. There are however cases where densely built up vegetation is sometimes erroneously classified as building. While the algorithm only requires LiDAR data for its classifications, a trade off to this is densely built up vegetation will sometimes appear very similar to building.

The building level completeness and correctness results, as a function of various building areas, for the Fairfield data set are shown in Figure 44. Figure 45 presents cumulative completeness and correctness which is simply the algorithm's completeness and correctness for all buildings having an area greater than the value shown on the x-axis.

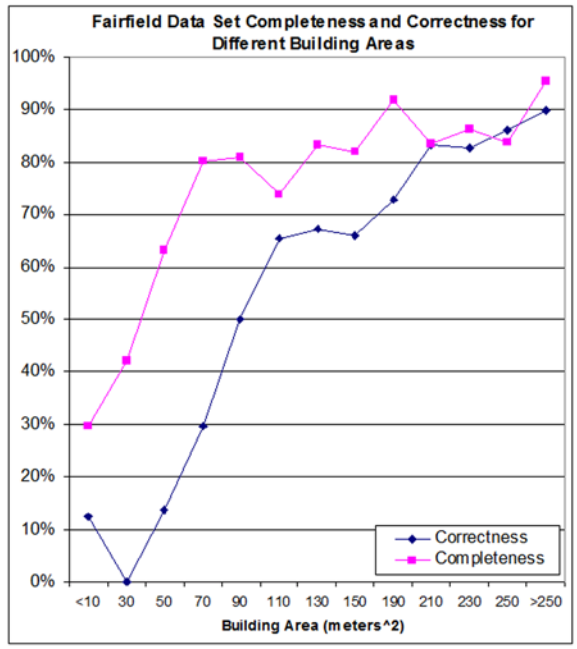


Figure 44 - Completeness and Correctness for the Fairfield Data Set

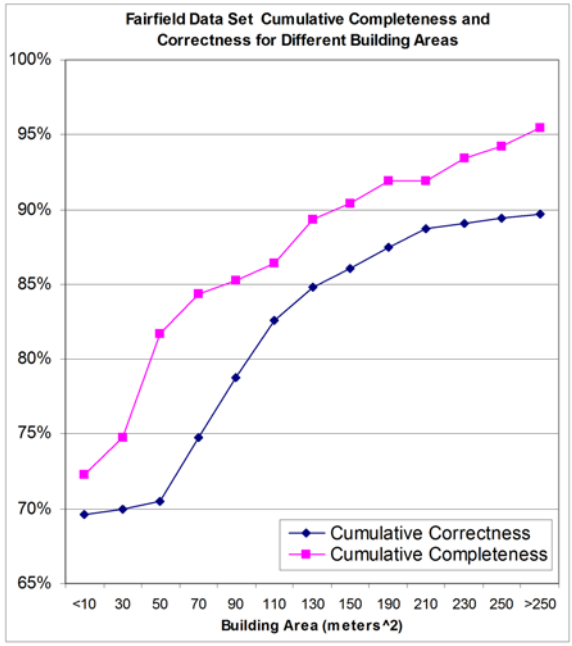


Figure 45 - Cumulative Completeness and Correctness for the Fairfield Data Set

For example, note that for a building area of 150 m^2 for the Fairfield data set, the algorithm has a cumulative completeness and cumulative correctness of approximately 90% and 86% respectively. This simply means that the algorithm's completeness and correctness for all buildings having an area of 150 m^2 and greater is 90% and 86% (respectively). Note that both the completeness and correctness for the Fairfield data set for 230m^2 is 83% and the cumulative completeness and correctness for 230m^2 is 93.0% and 90.7% respectively. The National Association of Home Builders estimated that in 2006 the average area of a new home in the United States was approximately 226m^2 [21]. Therefore, not only can the algorithm reliably detect buildings in the Fairfield data set that have an area equivalent to the average size home in the U.S. but it can also reliably detect buildings larger than that size as well.

The histogram for building sizes for the Fairfield data set is shown in Figure 46 and for the Anchorage data set in Figure 47. The heights of the individual bars represent the total

number of buildings manually extracted for the building area listed on the x-axis. The red part of the bar is proportional to the amount of those buildings which were correctly detected by the algorithm for the building area listed on the x-axis. The yellow part of the bar is proportional to the amount of buildings which the algorithm failed to correctly detect. For example, for >250 m^2, there was 312 buildings identified via manual extraction. Of those 312 buildings, 298 were correctly detected with only 14 false negatives; hence the yellow portion of the bar is small. However, for buildings having an area greater than 10 and less than 30 m^2, 359 buildings were identified in the reference set with only 151 being correctly detected and 208 buildings were not detected; hence the yellow portion of the bar is bigger than the red portion.

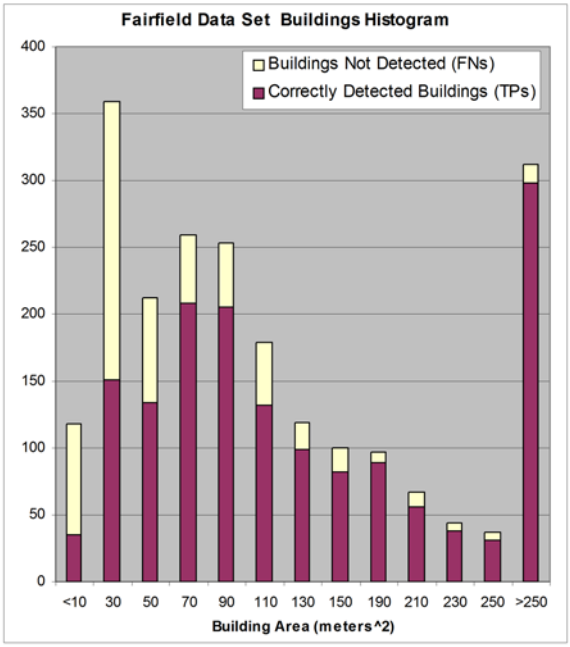


Figure 46 - Fairfield Data Set Building Histogram

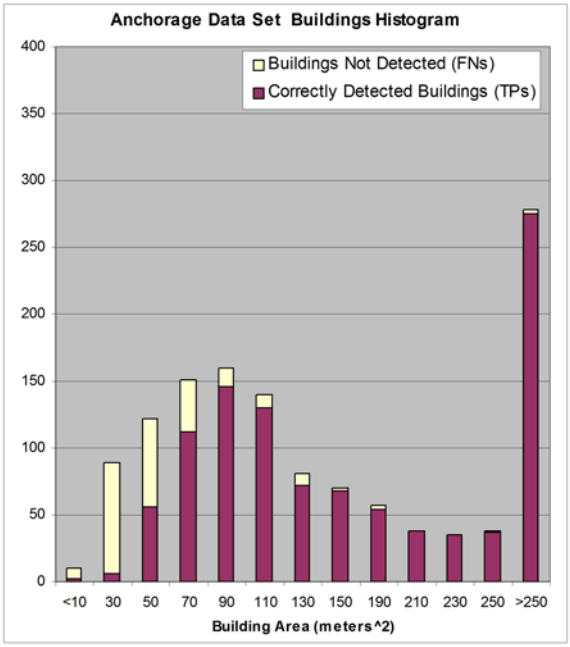


Figure 47 - Anchorage Data Set Building Histogram

The completeness and correctness for the Anchorage data set is shown in Figure 48. The cumulative completeness and cumulative correctness for the Anchorage data set is shown in Figure 49.

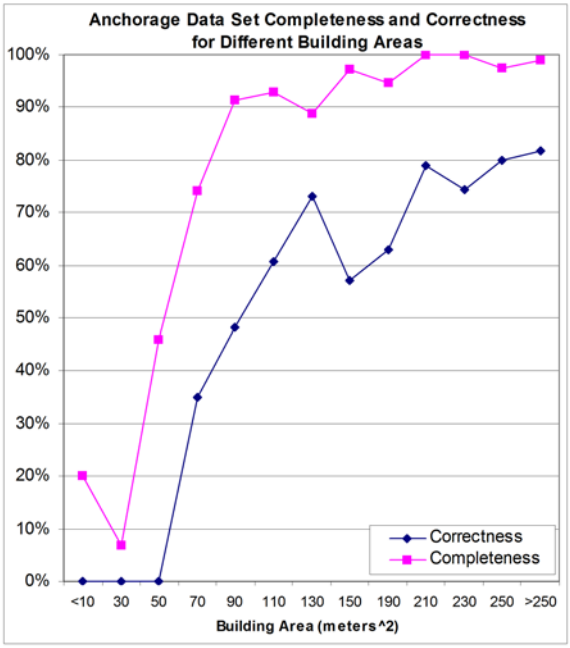


Figure 48 - Anchorage Data Set Completeness and Correctness

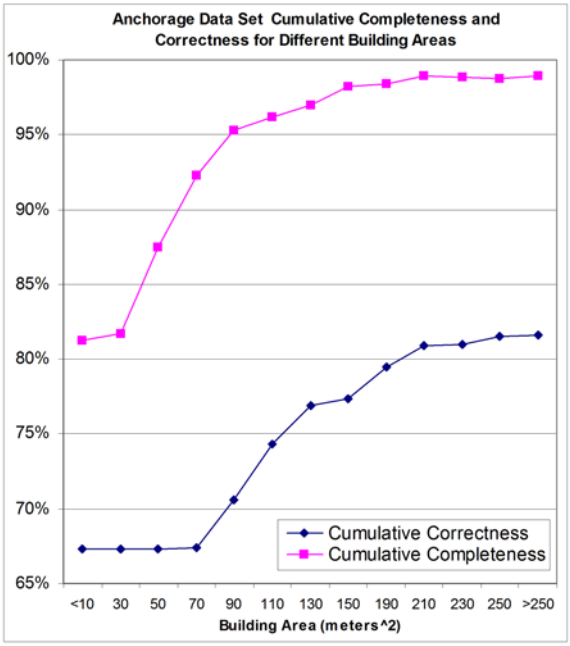


Figure 49 - Anchorage Data Set Cumulative Completeness and Correctness

Observe in Figure 48 that at 210 m^2 and greater for the Anchorage data set, the completeness realized is always greater than 95% and the correctness greater than 74%. Then for Figure 49 the cumulative completeness and correctness for 210 m^2 and greater is always better than 95% and 80% respectively. Considering 210 m^2 is close to the size of a house, these percentages are pretty good as the algorithm is reliably detecting the majority of the houses with reasonable accuracy and then larger buildings with even better accuracy.

For both data sets completeness was better than correctness on both a point level and building level. Furthermore, as seen in the cumulative graphs, as the building area increases, so does the completeness and the correctness. The reason for this is as the building area increases, more points exist to depict and represent that building in the LiDAR data. For both the Fairfield data set and the Anchorage data set, the algorithm had trouble reliably detecting buildings having a building area of less than 190 m^2. This can be seen in Figure 50 and Figure 51 which depict

zoomed in areas of the accuracy plots in Figure 40 and Figure 41. These zoomed in areas contain smaller buildings for the Fairfield and Anchorage data sets respectively. Recall, red points are correctly detected points, orange points are false positives and light green points are false negatives. However, buildings having an area between 190 and 210 m^2, the completeness and correctness for the Fairfield data set are was 92% and 73% respectively and for the Anchorage data set completeness and correctness for the same building are range were 95% and 63%. These results are acceptable as most other methods in the literature are also having trouble with detecting smaller buildings.

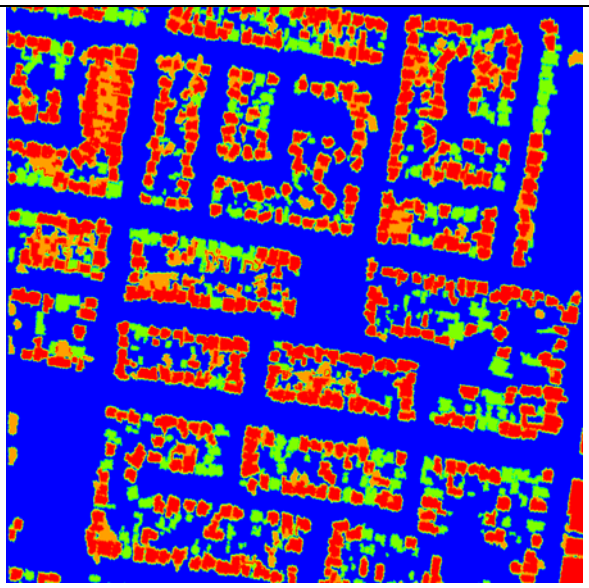


Figure 50 - Cropped Fairfield Accuracy Plot

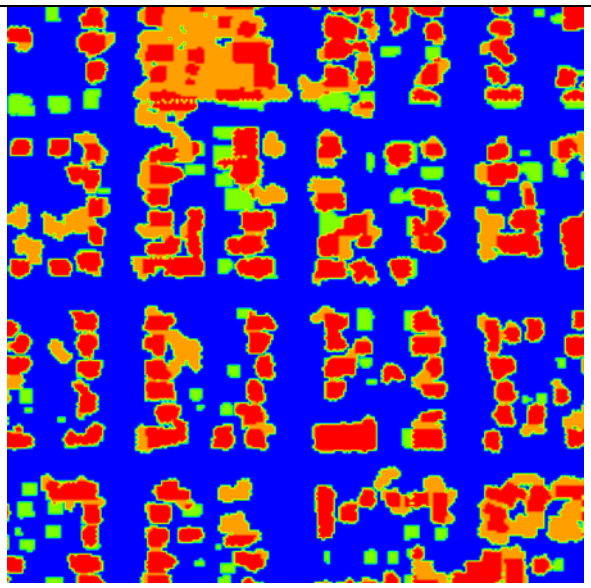


Figure 51 - Cropped Anchorage Accuracy Plot

7.2 Per Pixel Completeness and Correctness and Building Level Completeness and Correctness for Building Detection from Aerial Image Algorithm

The pixel level completeness and pixel level correctness for the Fairfield data set is 0.8258 and 0.6163 respectively. This means that 82.6% of the pixels belonging to buildings, according to the reference set, were correctly detected by the algorithm in the automatic set and that 61.6% of the pixels labeled as building by the algorithm in the automatic set were actually also labeled building in the reference set. The pixel level completeness and pixel level correctness for the Anchorage data set is 0.7486 and 0.4154 respectively. Again, 74.9% of the pixels belong to buildings were correctly detected by the algorithm and 41.5% of the pixels labeled as building by the algorithm actually corresponded to building pixels.

Figure 52 and Figure 53 present pixel level results for the Fairfield and Anchorage data sets respectively. In these figures, the blue color corresponds to correctly detected points, the light green color corresponds to false positives (points the algorithm classified as belonging to buildings in the automatic set but were in fact identified as not belonging to buildings in the reference set) and the red color corresponds to false negatives (points that were identified as belonging to buildings in the reference set but classified as not belonging buildings in the automatic set by the algorithm). The white in these figures corresponds to points not identified as building by both the algorithm in the automatic set and the manual extraction in the reference set.

Observe that in Figure 52 and Figure 53 almost all of the false positive pixels correspond to either roads, concrete or dirt. Some of the roads have been correctly classified as non-building because their solidity is too low when represented by a single texture segment. The majority of

false negatives are in the smaller buildings or are due to buildings casting shadows on other buildings. The algorithm does an excellent job not mistaking vegetation for building.

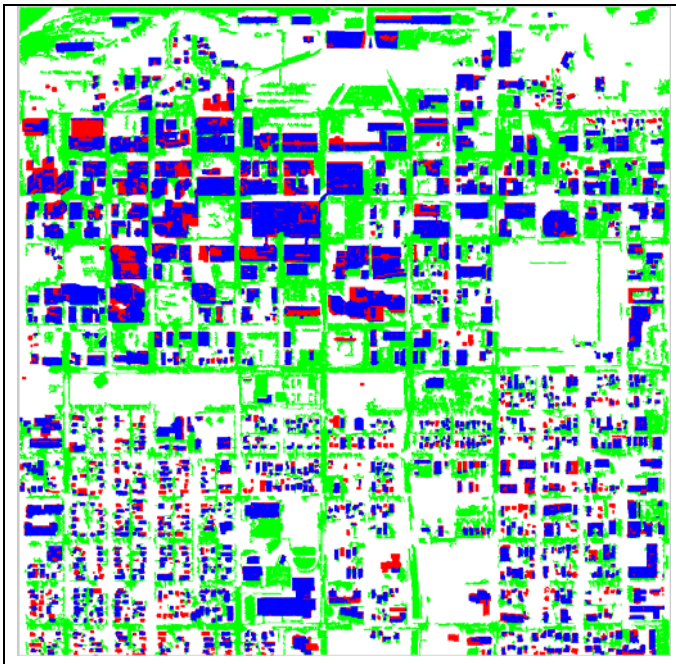


Figure 52: Building Pixel Accuracy for the Anchorage Data Set

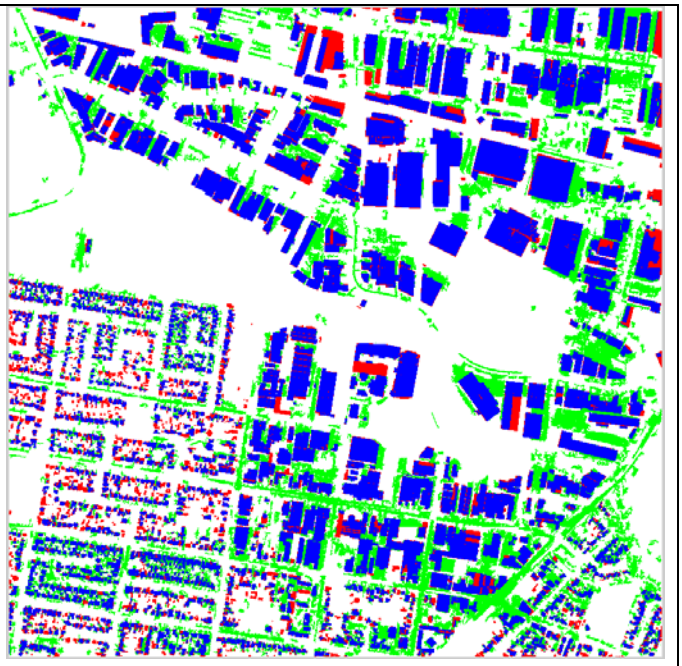


Figure 53: Building Pixel Accuracy for the Fairfield Data Set

Figure 54 and Figure 55 present classification results of the pixels as building, non-building, shadow and vegetation for the Fairfield and Anchorage data sets respectively. The red pixels correspond to what the proposed algorithm classified as building, the green pixels as vegetation, the blue pixels as shadow and the white pixels as non-building. Unfortunately, the dark side of the trees are often classified as shadow, which is correct however the trees typically overcast vegetation. Most of the non-building correctly corresponds to concrete surfaces that do not belong to building but to roads or parking lots. Again, observe the algorithm does an excellent job identifying the sun side part of the dense forestry and the rest of the vegetation existent in both data sets.

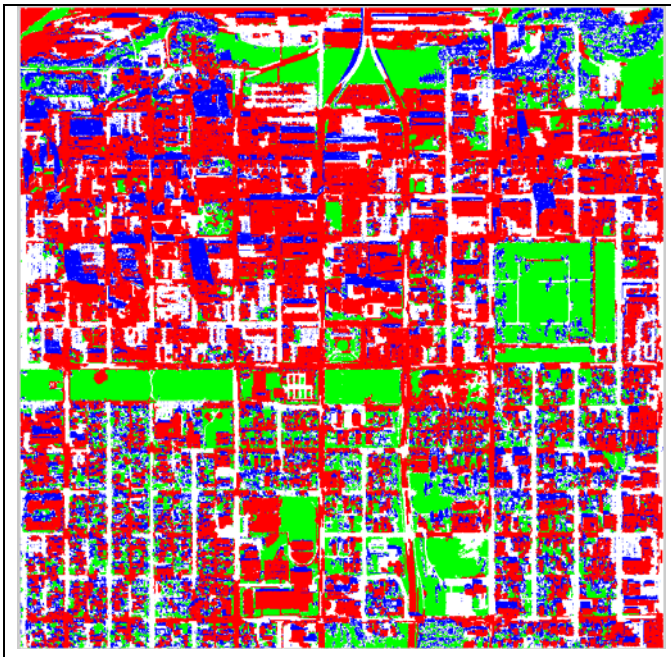


Figure 54: Pixel Classification for Anchorage Data Set

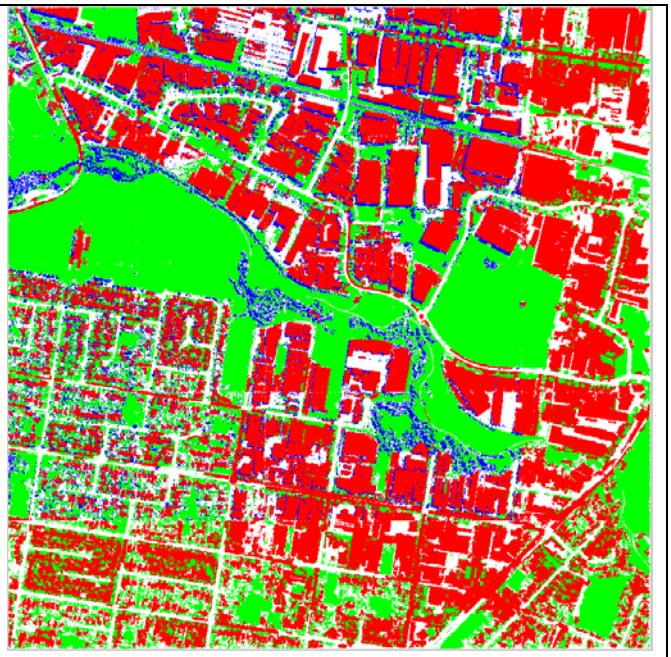


Figure 55: Pixel Classification for Fairfield Data Set

The building level completeness and correctness results, as a function of various building areas, for the Anchorage data set are shown in Figure 56. Figure 57 presents cumulative completeness and correctness which is simply the algorithm's completeness and correctness for all buildings having an area greater than the value shown on the x-axis.

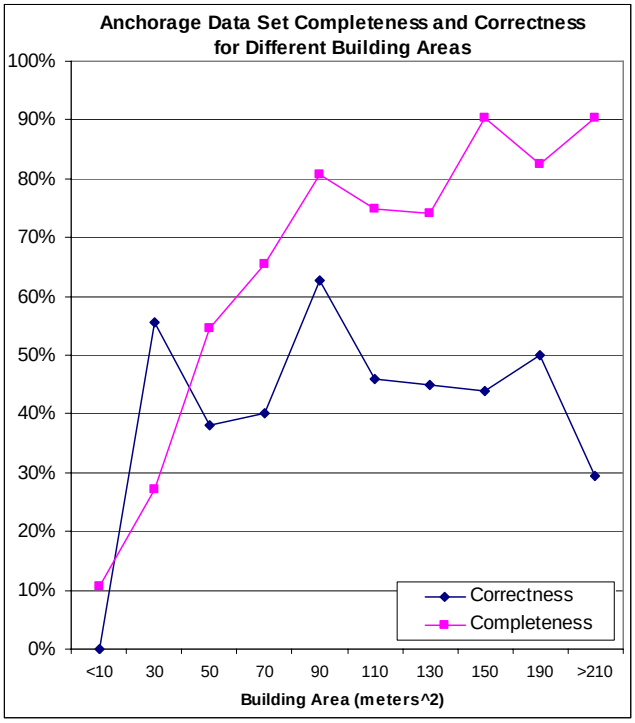


Figure 56: Completeness and Correctness for the Anchorage Data Set

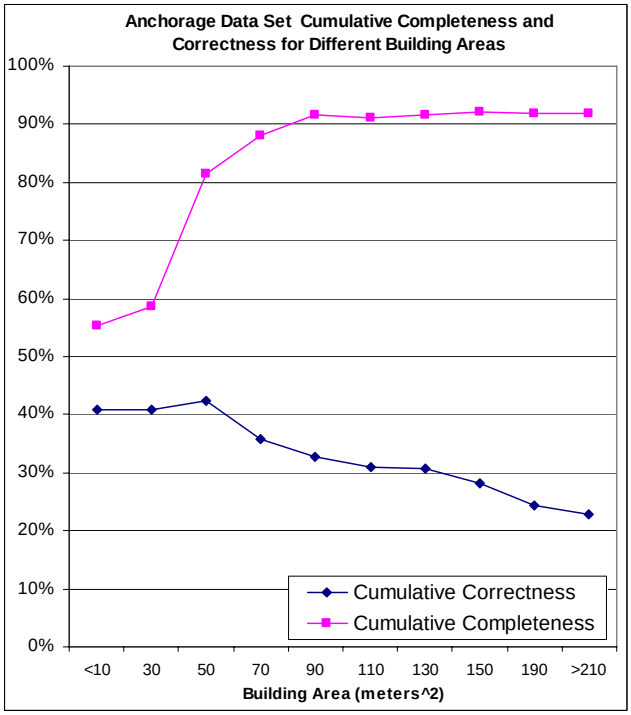


Figure 57: Cumulative Completeness and Correctness for the Anchorage Data Set

For example, note that for a building area of 150 m^2 for the Fairfield data set, the algorithm has a cumulative completeness and cumulative correctness of approximately 90% and 33% respectively. This simply means that the algorithm's completeness and correctness for all buildings having an area of 150 m^2 and greater is 90% and 33% (respectively). The range was plotted from between 10 m^2 to 210 m^2 as after 210 m^2 the plot looks pretty much the same. The algorithm detects 75% of the buildings larger than 90 m^2 but at the same time its correctness suffers at the expense of false positives. The reason for the mediocre correctness is the algorithm is completely automated: no parameters change during the execution across both data sets and furthermore there are no training phases. Also, no assumptions are implemented to tailor to a specific building size or shape. As it will soon be shown there are a variety of different building

sizes in the data sets. Finally, the mediocre correctness is also because the automatic algorithm is detecting these buildings from only a single nadir aerial image.

The completeness and correctness for the Fairfield data set is shown in Figure 58. The cumulative completeness and cumulative correctness for the Fairfield data set is shown in Figure 59. The completeness for the Fairfield data set in comparison to the Anchorage are about the same yet the correctness for the Fairfield is better than that of the Anchorage. The reason for this is the Anchorage has significantly more areas of non-building concrete (streets, parking lots, tennis courts, etc.) than the Fairfield data set.

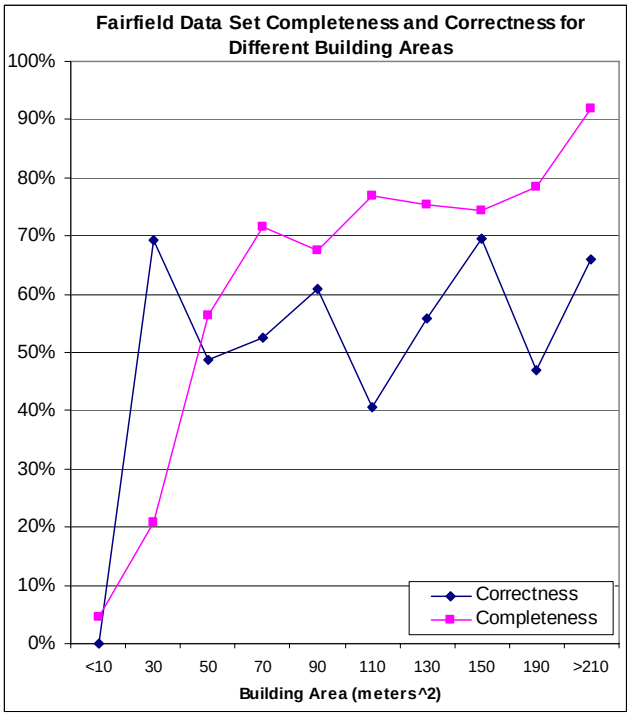


Figure 58: Completeness and Correctness for the Fairfield Data Set

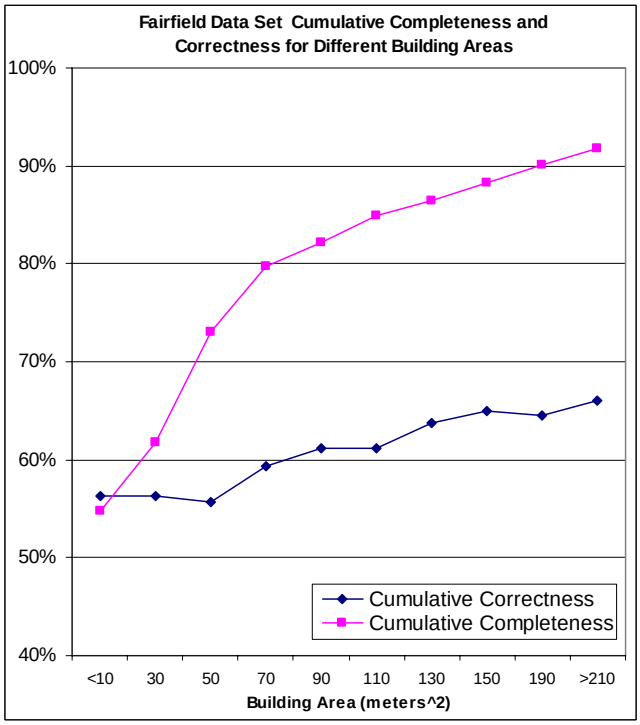


Figure 59: Cumulative Completeness and Correctness for the Fairfield Data Set

The histogram for building sizes for the Anchorage data set is shown in Figure 60 and for the Fairfield data set in Figure 61. The heights of the individual bars represent the total number

of buildings manually extracted for the building area listed on the x-axis. The red part of the bar is proportional to the amount of those buildings which were correctly detected by the algorithm for the building area listed on the x-axis. The yellow part of the bar is proportional to the amount of buildings which the algorithm failed to correctly detect. For example in Figure 61, for >210 m^2, there was 61 buildings identified via manual extraction. Of those 61 buildings, 56 were correctly detected with only 5 false negatives, hence the yellow portion of the bar is small. However, for buildings having an area greater than 10 and less than 30 m^2, 708 buildings were identified in the reference set with only 311 being correctly detected and 397 buildings were not detected; hence the yellow portion of the bar is bigger than the red portion. The building histograms in Figure 60 and Figure 61 demonstrate the variety of building sizes existent in both data sets.

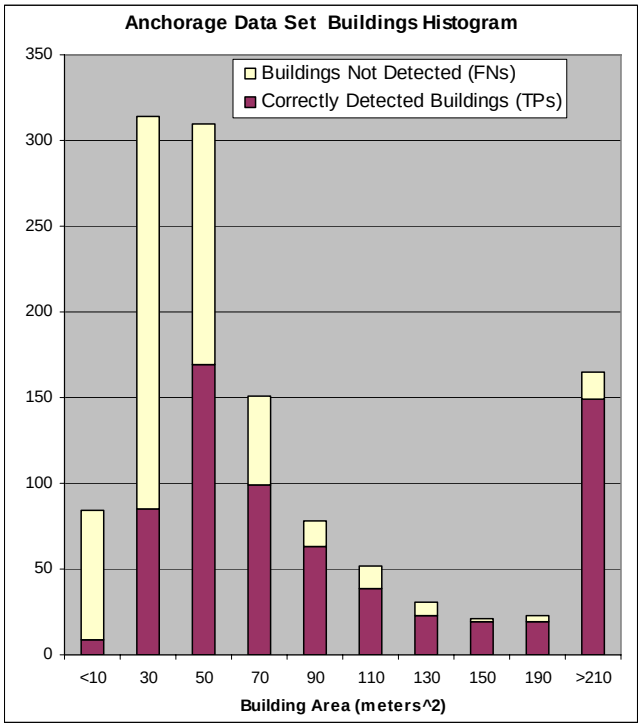


Figure 60: Anchorage Data Set Histogram

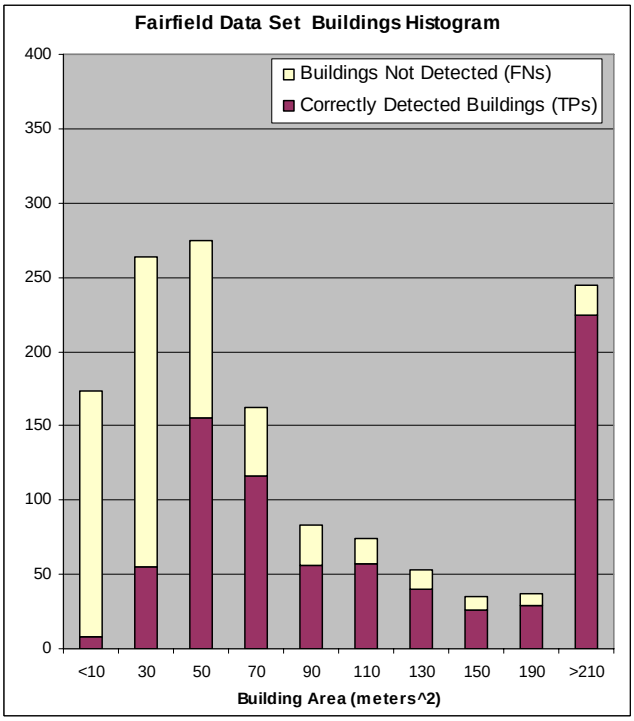


Figure 61: Fairfield Data Set Histogram

7.3 Results for Registration

Recall from section 5.3 the phase correlation registration algorithm requires the binary building mask of the LiDAR data and the binary building mask of the aerial image where a binary building mask is an image having white pixels where the algorithm detected building and black pixels for everything else. The binary building mask for the buildings automatically detected from the LiDAR data for the Fairfield data set is presented in Figure 62. The binary building mask for the buildings automatically detected from the aerial image for the Fairfield data set is presented in Figure 63.

Figure 62: Fairfield Data Set LiDAR Building Mask

Figure 63: Fairfield Data Set Aerial Image Building Mask

The binary building mask for the buildings automatically detected from the LiDAR data for the Anchorage data set is presented in Figure 64. The binary building mask for the buildings

automatically detected from the aerial image for the Anchorage data set is presented in Figure 65.

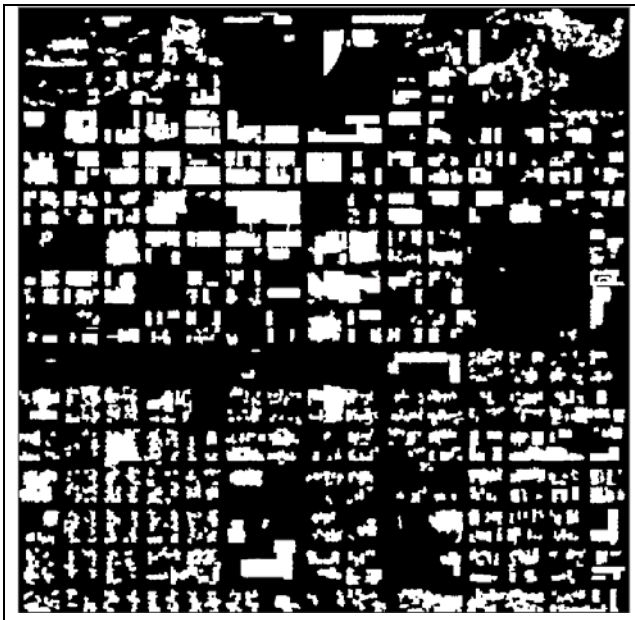


Figure 64: Anchorage Data Set LiDAR Building Mask

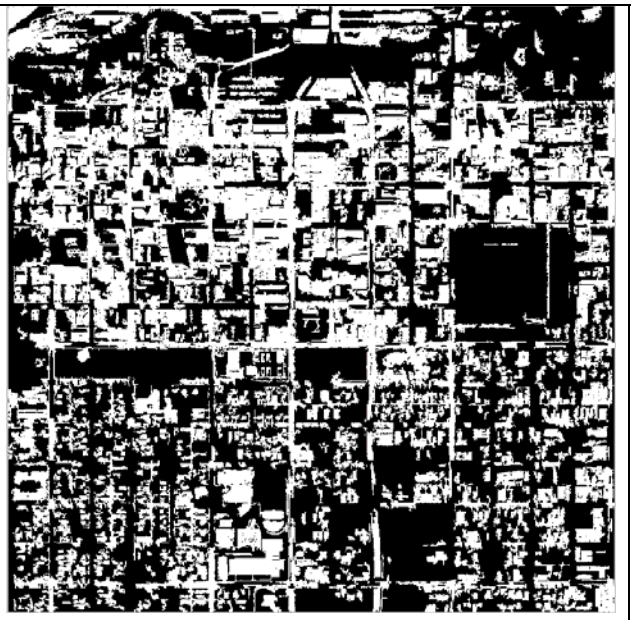


Figure 65: Anchorage Data Set Aerial Image Building Mask

To show the data sets registered onto of one another, the following is done. First, the binary LiDAR building mask is multiplied by the interpolated LiDAR data. This results in black pixels for everything but detected buildings which still retain their elevation values. Then the aerial imagery is registered to the LiDAR data and everywhere in the LiDAR data there exists a detected building the aerial imagery pixel intensity is replaced with that detected building pixel's interpolated LiDAR elevation data. This then produces an aerial image with only LiDAR data shown where the buildings exist in the LiDAR data. If the two mediums are properly registered, only building pixels in the aerial imagery will be replaced by interpolated LiDAR values. The LiDAR data registered to the aerial image for the Fairfield data set is shown in Figure 66. The LiDAR data registered to the aerial image for the Anchorage data set is shown in Figure 67.

Figure 66: Fairfield Data Set LiDAR data registered to Aerial Imagery

Figure 67: Anchorage Data Set LiDAR data registered to Aerial Imagery

One can see from above, the data sets mostly if not completely overlap. The Anchorage data set was already interpolated, so when the aerial image was resized with the meta data to match the point density of the LiDAR data, there was no significant scale change that had to be handled by the registration algorithm. Due to the Fairfield data set having irregular point spacing, even after downscaling the aerial image with the meta data, there was still a 3% scaling difference which the registration algorithm had to adjust the data so that they would properly align together.

Due to the data sets already mostly overlapping and with the meta-data resizing the aerial image so that there were little scaling differences, further testing was implemented to benchmark the robustness of the proposed method. To further test the registration algorithm, artificial rotations and scalings were introduced. Furthermore, parts of the data set were cropped in the following fashion so that there was not complete overlap. The overlap parameter β is defined as

the percentage of overlap of both dimensions as depicted in Figure 68. The two data sets were approximately square and therefore R and C were about equal to one another. Let the overlap area then be denoted as A and is calculated as follows: $A = 1 - 2 \cdot \beta$. Let the dotted square be one image and the solid square be another image where both images are to be registered by the proposed registration algorithm. As β increases, the overlap between the sources decreases where when β is 1 there is no overlap and when β is 0 the images completely overlap: $\beta \in [0,1]$.

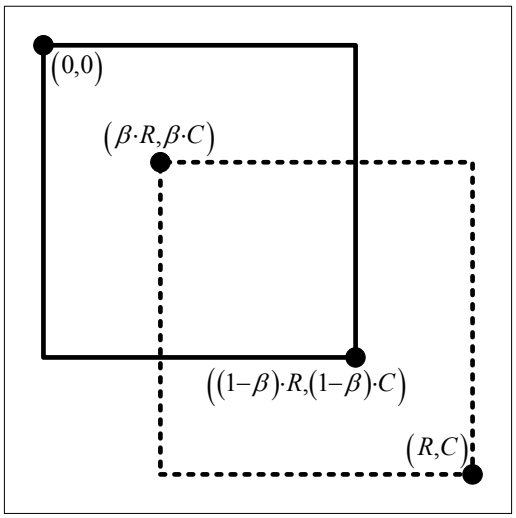


Figure 68: North West Overlap Image Cropping Parameter for Registration Algorithm

For the Fairfield and Anchorage data set, the registration algorithm was tested to determine the minimum percentage of area overlap such that the algorithm would still correctly register the sources of data together. In finding this minimum overlap, no artificial rotation and scaling transformations were introduced, yet. The minimum overlap required for successful registration for the Fairfield data set was 82% (β = 0.09) and for 66% (β = 0.17) for the Anchorage data set when the LiDAR data overlapped with the North West portion of the aerial imagery. The registration of the data sets with this overlap is depicted in Figure 69 and Figure

70. The black borders are existent because there is not complete overlap, i.e. there is LiDAR data that doesn't have aerial image coverage and vice versa.

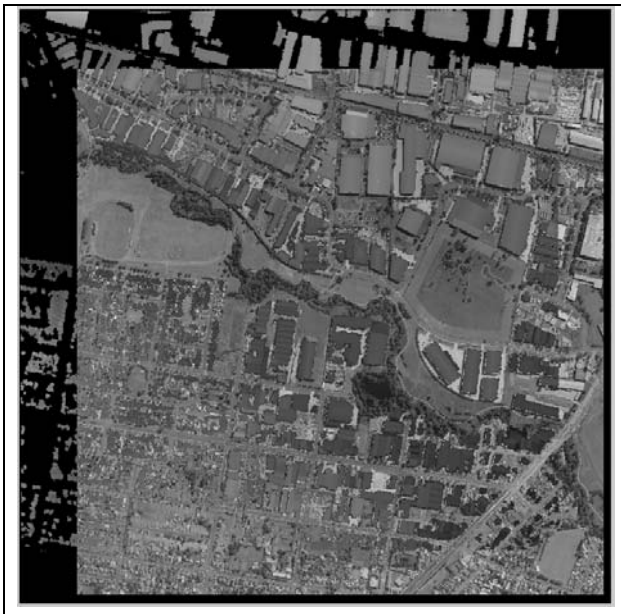


Figure 69: Fairfield Data Set Registration with 82% NW Overlap

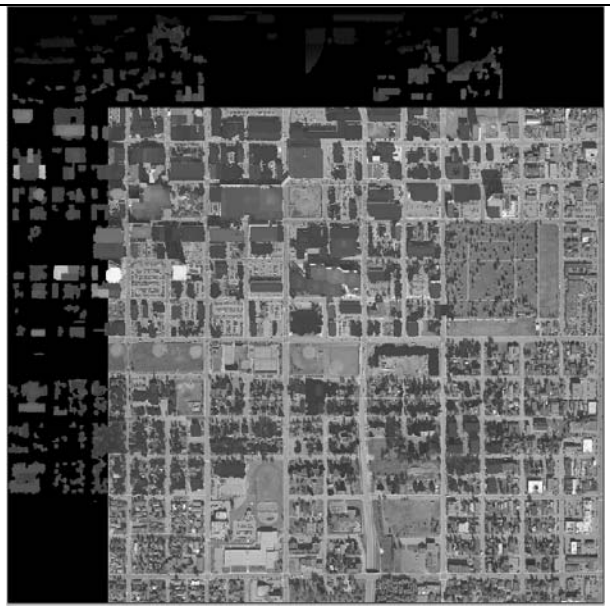


Figure 70: Anchorage Data Set Registration with 66% NW Overlap

When the LiDAR data was overlapping with the South West portion of the aerial imagery, the minimum overlap for the Fairfield and Anchorage data sets were 78% and 62% respectively as depicted in Figure 71 and Figure 72.

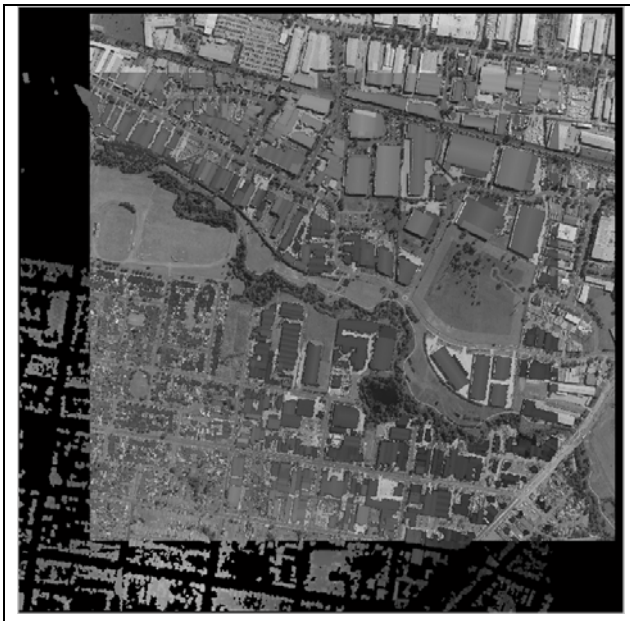


Figure 71: Fairfield Data Set Registration with 78% SW Overlap

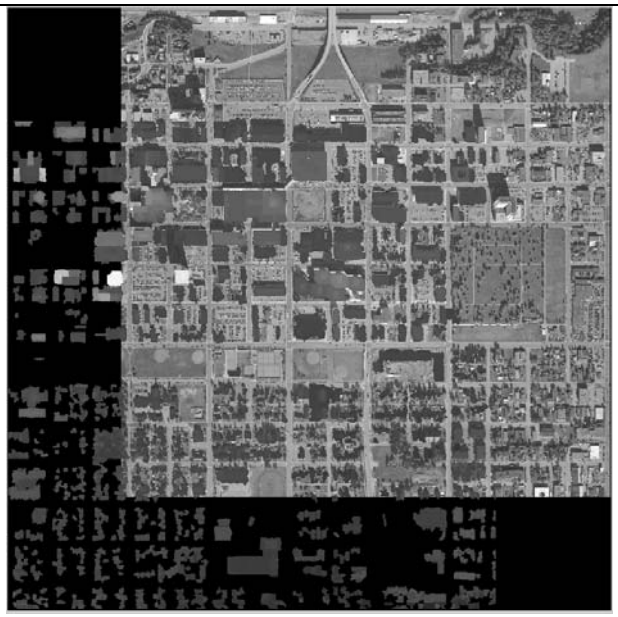


Figure 72: Anchorage Data Set Registration with 62% SW Overlap

When the LiDAR data was overlapping with the North East portion of the aerial imagery, the minimum overlap for the Fairfield and Anchorage data sets were 76% and 68% respectively as depicted in Figure 73and Figure 74.

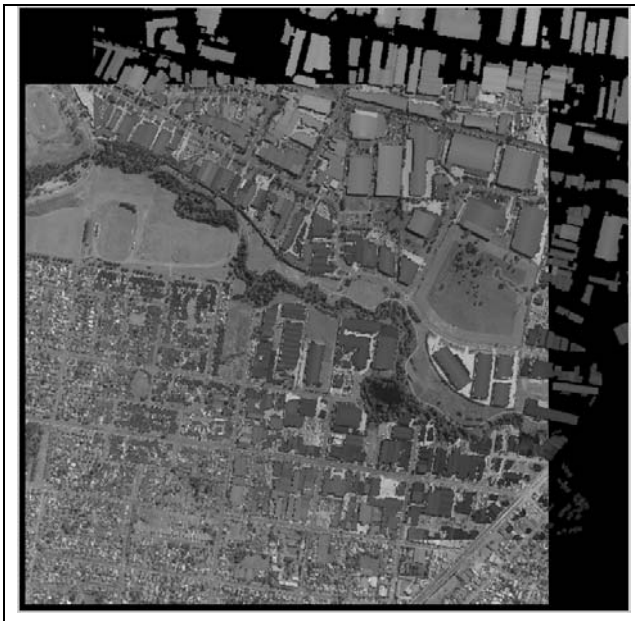


Figure 73: Fairfield Data Set Registration with 76% NE Overlap

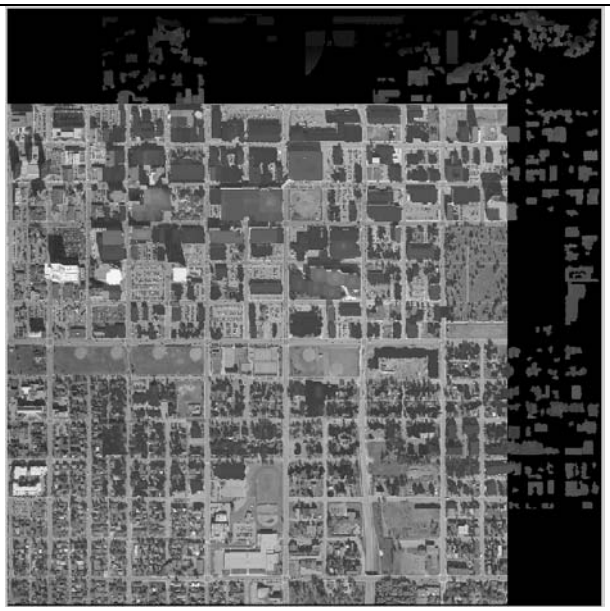


Figure 74: Anchorage Data Set Registration with 68% NE Overlap

When the LiDAR data was overlapping with the South East portion of the aerial imagery, the minimum overlap for the Fairfield and Anchorage data sets were 74% and 64% respectively as depicted in Figure 75 and Figure 76.

Figure 75: Fairfield Data Set Registration with 74% SE Overlap

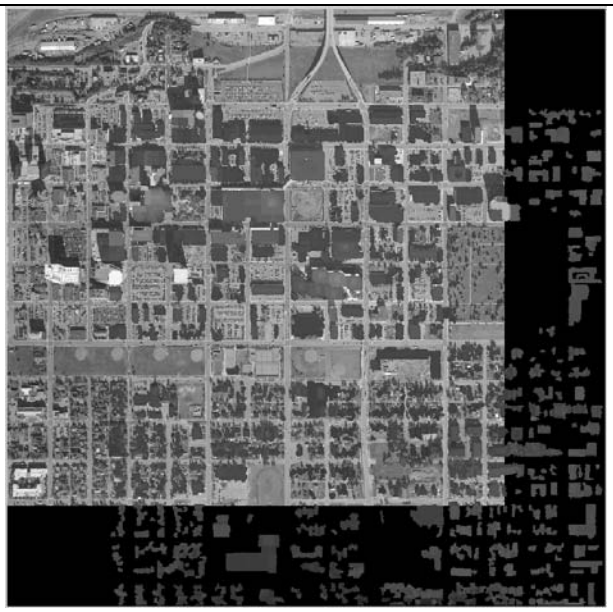


Figure 76: Anchorage Data Set Registration with 64% NE Overlap

Next, the LiDAR data overlapping the North West portion of the aerial imagery at 90% was taken and the scaling and rotation were artificially varied. The North West overlap was chosen arbitrarily. The following table shows all of the tested rotations and scalings, the algorithm's estimation of the rotations and scalings applied, and the percent error between the actual and estimated. For all of the tests listed in the table, the algorithm successfully registered the two sources of data.

	Scaling	Est Scale	Error	Rotation	Est Rot	Error
Fairfield	0.9	0.9038	0.42%	320	319.78	0.07%
	0.9	0.9068	0.76%	340	339.84	0.05%
	0.9	0.9020	0.22%	20	19.97	0.16%
	0.9	0.9023	0.26%	40	40.04	0.10%
	1.1	1.1018	0.16%	320	319.78	0.07%
	1.1	1.0998	0.02%	340	340.04	0.01%
	1.1	1.0998	0.02%	20	20.11	0.57%
	1.1	1.0972	0.25%	40	40.08	0.20%
Anchorage	0.9	0.8982	0.20%	350	350.05	0.01%
	0.9	0.8972	0.31%	355	354.80	0.06%
	0.9	0.8972	0.31%	5	5.20	4.05%
	0.9	0.8982	0.20%	10	10.29	2.86%
	1.1	1.0976	0.22%	350	350.22	0.06%
	1.1	1.0924	0.69%	355	354.76	0.07%
	1.1	1.1009	0.08%	5	5.24	4.77%
	1.1	1.0976	0.22%	10	9.78	2.19%

Table 1: Tested Rotation, Translation and Scaling and Corresponding Errors for Registration

7.4 Vegetation Classifier Results: Proposed Color Segmentation and Color Invariants vs. NDVI Thresholding

The reason it is mentioned that the Fairfield data set contains the returned LiDAR intensity is that it is possible to use the returned LiDAR intensity and one of the channels of the aerial image to compute a pseudo normalized difference vegetation index (NDVI) and threshold that NDVI image to identify vegetation. Rottensteiner et. al. in [30] propose using the pseudo NDVI image calculate it as shown in (45) where L(i,j) is the returned LiDAR intensity at pixel (i,j). Note that in order to use the NDVI image, the LiDAR data must be interpolated and registered to the aerial imagery so that L(i,j) and I(i,j,k) correspond to the same location in the depicted scene.

$$N(i,j)=\frac{L(i,j)-I(i,j,r)}{L(i,j)+I(i,j,r)} \tag{45}$$

When light makes contact with objects, certain wavelengths of the spectrum of the light are absorbed and others reflected. The pigment in plant leaves, chlorophyll, strongly absorbs visible light (ranging from 0.4 to 0.7 micrometers) for use in photosynthesis. The cell structure of the leaves however strongly reflects near-infrared light (from 0.7 to 1.1 micrometers). Therefore, if significantly more reflected radiation exists in near infrared wavelengths than invisible wavelengths for a given location, then the vegetation in that pixel is probably dense and may contain some type of forest. If the difference is relatively small in the intensity of visible and near infrared wavelengths reflected, then the vegetation is sparse and may consist of grassland, tundra or desert. The calculations for NDVI for a given pixel range from -1 (corresponding to water) to +1 (corresponding highest possible density of green leaves) and zero means no vegetation [22].

With the manually developed reference set, it is then known which pixels belong to buildings and which do not. The pseudo NDVI image was then thresholded with 40 different thresholds to determine where vegetation exists. The thresholds ranged from -0.5 to 0.5 in increments of 0.025. The pseudo correctness for the red, blue and green channel based NDVI images was calculated for all of those thresholds and then the accuracy was plotted. The pseudo correctness is calculated as follows:

$$pCorr = \frac{pTP}{pTP + FP} \tag{46}$$

Where *pTP* is Pseudo True positive and *FP* is a false positive. If the thresholded NDVI calls a pixel vegetation and that corresponding pixel is not labeled as belonging to a building in the reference set, then the pixel is classified as a pseudo true positive. The reason this is referred to as a 'pseudo true positive' is because what is not building could be sidewalk, cement, etc in

addition to vegetation. A reference set was only developed for identifying buildings, not vegetation.

If the thresholded NDVI classifies a pixel as vegetation and that corresponding pixel is labeled as belonging to a building in the reference set, then the pixel is classified as a false positive. A given pixel cannot belong to vegetation and building at the same time because the two classes are mutually exclusive by definition (vegetation does not contain building and vice versa).

Because a reference set was not developed which identified the vegetation ground truth, false negatives cannot be calculated. A false negative being pixels the thresholded NDVI labels as not belonging to vegetation when in fact they do belong to vegetation. Instead only false positives can accurately be determined. Now although in [30] Rottensteiner et. al. use the red channel, in earlier works [28], they report using the green channel of the aerial imagery. Figure 77 shows the pseudo correctness plotted for the red, green and blue thresholded NDVI images for all thresholds tested. Note in Figure 77, the red line corresponds to the NDVI constructed from the returned LiDAR intensity and the red channel of the corresponding color aerial imagery (similarly for the green and blue channels for the green and blue lines). Figure 78 shows the percentage of pixels classified as vegetation by the pseudo NDVI approach.

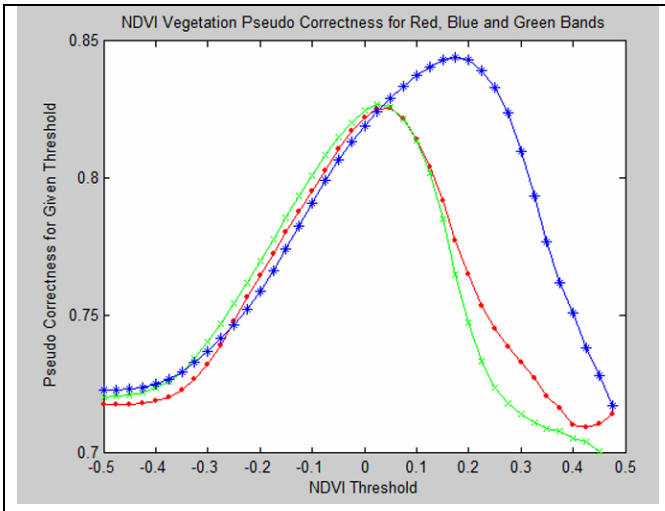


Figure 77: Pseudo Correctness for Red, Blue and Green Channel Based NDVI Vegetation Identification

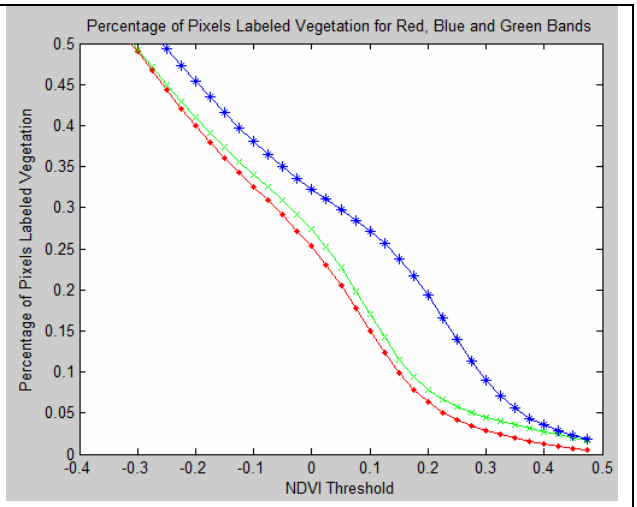


Figure 78: Percentage of Pixels labeled Vegetation by Pseudo Correctness for Red, Blue and Green Channel Based

Figure 79 shows the blue band based NDVI and Figure 80 shows which pixels were classified as vegetation (white pixels) for the blue band based NDVI classifier when the threshold was set to 0.175 for the 84.387%.

Figure 79: Blue Band NDVI Image for Fairfield Data Set

Figure 80: Vegetation Identification using NDVI thresholding Method

Notice there is a parameter associated with the pseudo NDVI approach and that not only does it require LiDAR data and Aerial Imagery but also it requires the registration of the two different sources as well.

The proposed color invariant vegetation detection approach was tested on the Fairfield Data Set and the Anchorage Data Set and it realized a pseudo correctness of 97.25% and 96.91% respectively! Note that the proposed approach only requires the aerial imagery and is automatic - no parameter adjustment. A comparison of the best NDVI results and the proposed color invariant method are presented in Table 2 where the proposed method is listed as 'Clr Inv Veg' and the best results from thresholding the different channel NDVI images are listed as 'NDVI Red', 'NDVI Blue' and 'NDVI Green'.

Method	Correctness	Coverage	Data Set
NDVI Red	82.55%	20.56%	Fairfield
NDVI Green	82.71%	25.23%	Fairfield
NDVI Blue	84.39%	21.67%	Fairfield
Clr Inv Veg	97.25%	30.05%	Fairfield
Clr Inv Veg	96.91%	17.88%	Anchorage

Table 2: Correctness and Coverage Comparison

The detected vegetation pixels using the proposed color invariant vegetation detection method are shown for the Fairfield data set in Figure 81 and for the Anchorage data set in Figure 82 where the detected vegetation pixels are marked in solid white.

Figure 81: Color Invariant Vegetation Detection for Fairfield Data Set

Figure 82: Color Invariant Vegetation Detection for Anchorage Data Set

7.5 Results for Improved Building Detection when considering both LiDAR Data and Aerial Imagery

Combining the vegetation detection algorithm with the LiDAR algorithm helped removed patches of vegetation that were erroneously being labeled as building. In combining the algorithms, the per pixel completeness and correctness for the Fairfield data set is 92.5% and 81.9% and 88.9% and 75.4% for the Anchorage data set. This is a -0.37% difference in completeness and a +1.72% in correctness for the Fairfield data set and a -0.43% difference in completeness and a +0.78% difference in correctness. In both cases, the increase in correctness outweighs the decrease in completeness. Per pixel accuracy plots were created for both data sets after combining the classifiers. For the Anchorage data set, 129,417 points were classified as grass vegetation (10.9% of the total number of points) and of those points 4,470 were originally building (0.4% of the total) and then relabeled to grass vegetation. For the Fairfield data set, 440,523 points were classified as grass vegetation (15.5% of the total number of points) and of those points 15,934 were originally buildings (0.56% of the total) which were then relabeled to grass vegetation. In the Fairfield LiDAR data automatic set, the number of incorrect regions (i.e. false positive regions labeled as building that did in fact not correspond to building) greater than $250m^2$ was 327. In the automatic set where the vegetation detection from the aerial imagery was applied, that number dropped to 316. In the Anchorage LiDAR data automatic set, the number of incorrect regions greater than $250m^2$ was 365. In the automatic set where the vegetation detection from the aerial imagery was applied, that number dropped to 363.

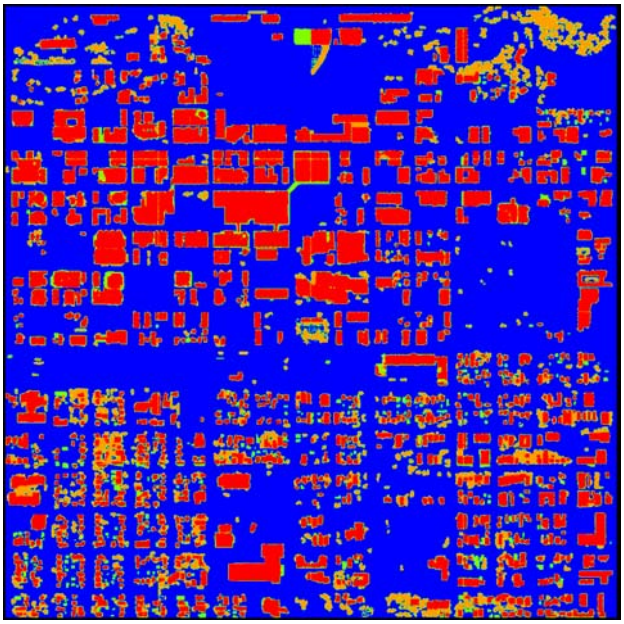


Figure 83: Per Pixel Accuracy for the Anchorage Data Set

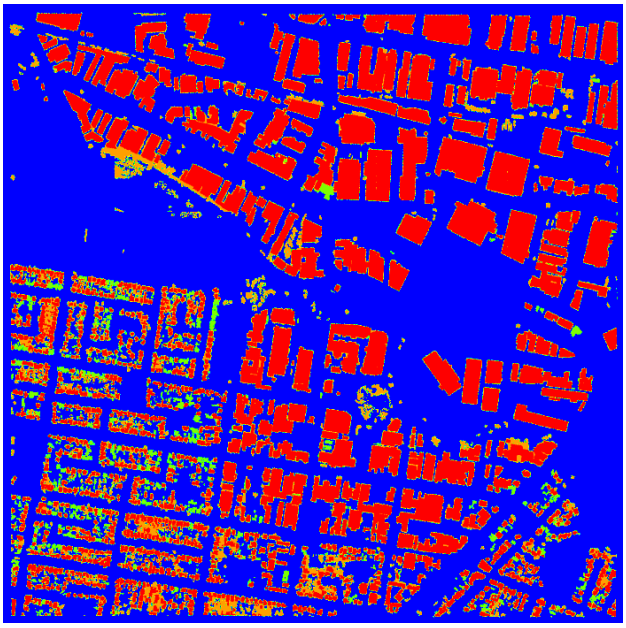


Figure 84: Per Pixel Accuracy for the Fairfield Data Set

Several vegetation patches, erroneously labeled as building by the building detection from LiDAR approach, have been removed by combining that algorithm with the vegetation detection approach. Cropped, zoomed in areas before the vegetation detection approach and after are shown for the Anchorage data set in Figure 85 and Figure 86 respectively.

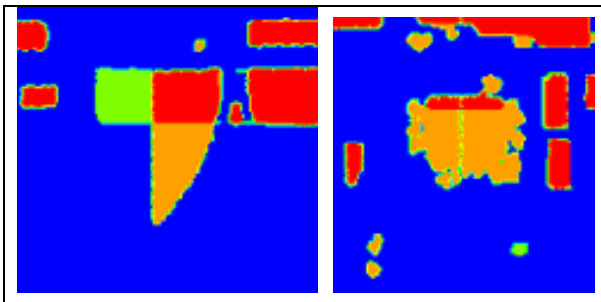


Figure 85: Anchorage LiDAR Before Vegetation Detection

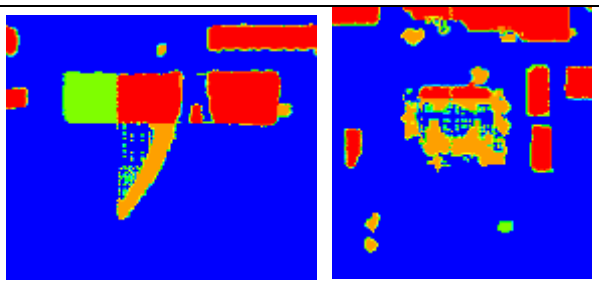


Figure 86: Anchorage LiDAR After Vegetation Detection

Cropped, zoomed in areas before and after the vegetation detection approach for the Fairfield data set are shown in Figure 87 and Figure 88 respectively.

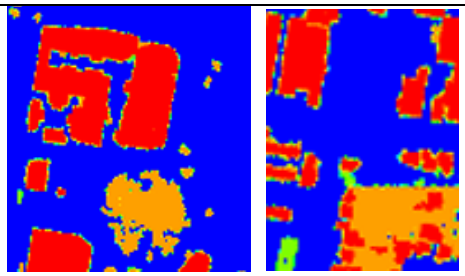


Figure 87: Fairfield LiDAR Before Vegetation Detection

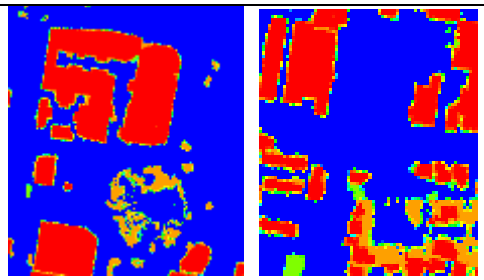


Figure 88: Fairfield LiDAR After Vegetation Detection

The vegetation detection approach, when applied to the LiDAR classification, further improves the correctness by removing these patches erroneously labeled as building.

Using the vegetation detection algorithm in combination with the LiDAR algorithm, the LiDAR data is now classified into 4 classes: buildings, non-building (mostly trees), vegetation (grass) and ground (dirt and pavement). The classification results for using features from both the LiDAR data and aerial imagery for the Fairfield data set are presented in Figure 90 and for the Anchorage data set in Figure 89. The orange points correspond to buildings, the yellow to non-ground, the green to ground and the red to vegetation. Note, for the Fairfield data set, only LiDAR points with aerial image coverage could be labeled as vegetation. This is the reason why the vegetation classifications only exist at the center of Figure 90 and why the vegetation classification stops abruptly along straight lines near the edges of Figure 90.

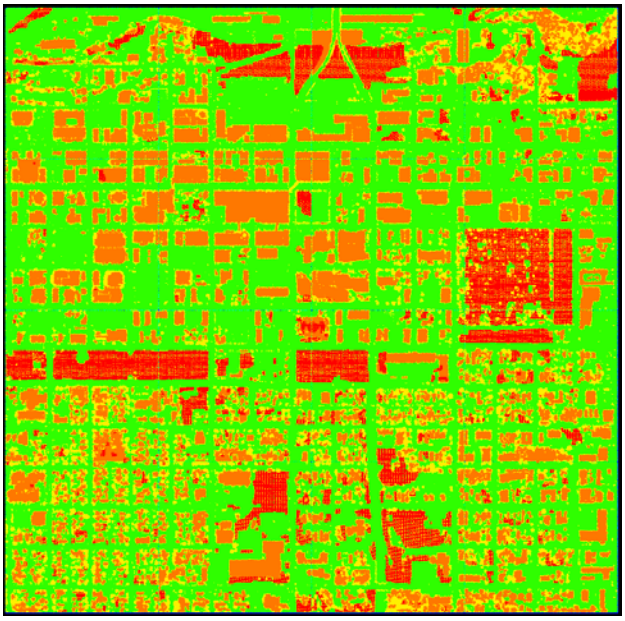


Figure 89: Per Pixel Classification for the Anchorage Data Set

Figure 90: Per Pixel Classification for the Fairfield Data Set

The building completeness and correctness for the Fairfield data set is presented in Figure 91 and for the Anchorage data set in Figure 93. Overall, a slight decrease in completeness and a larger increase in correctness were observed as a result of combining the vegetation detection. The same performance was also noticed for the cumulative completeness and correctness presented in Figure 92 for the Fairfield data set and Figure 94 for the Anchorage data set.

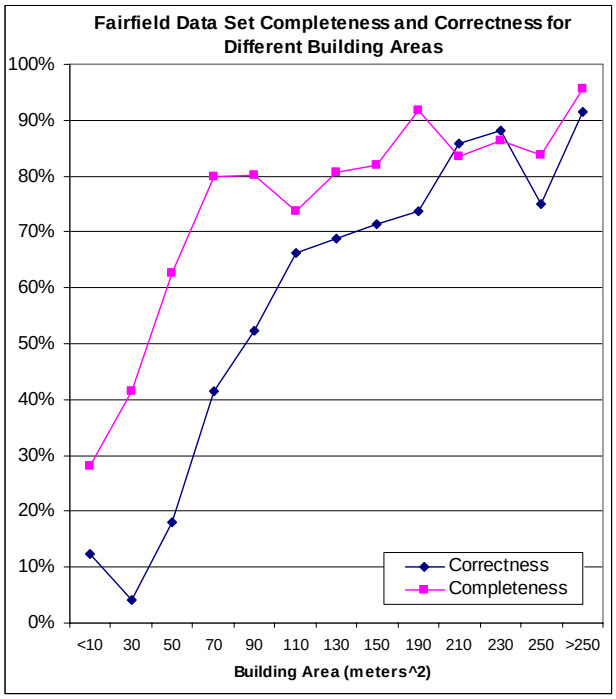


Figure 91: Building Completeness and Correctness for the Fairfield Data Set

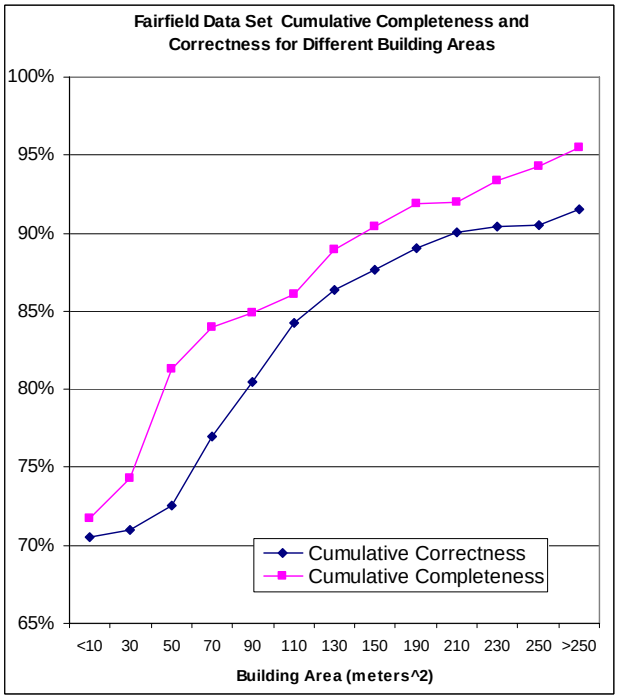


Figure 92: Cumulative Completeness and Correctness for the Fairfield Data Set

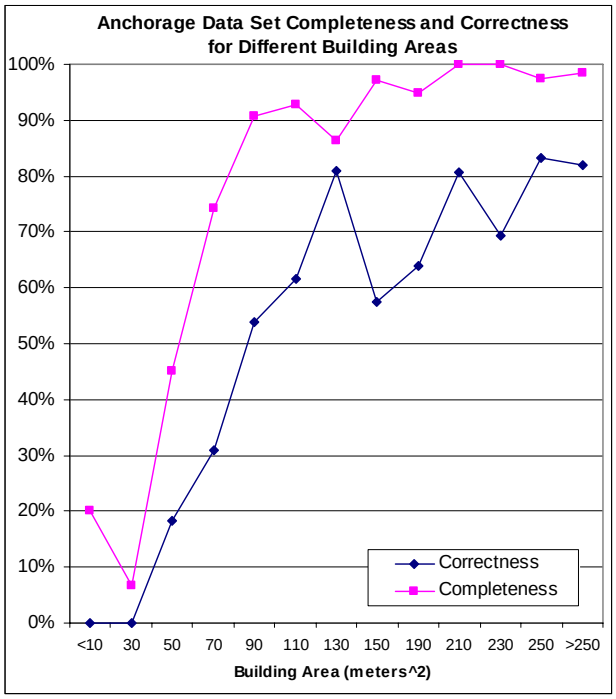


Figure 93: Anchorage Data Set Building Completeness and Correctness

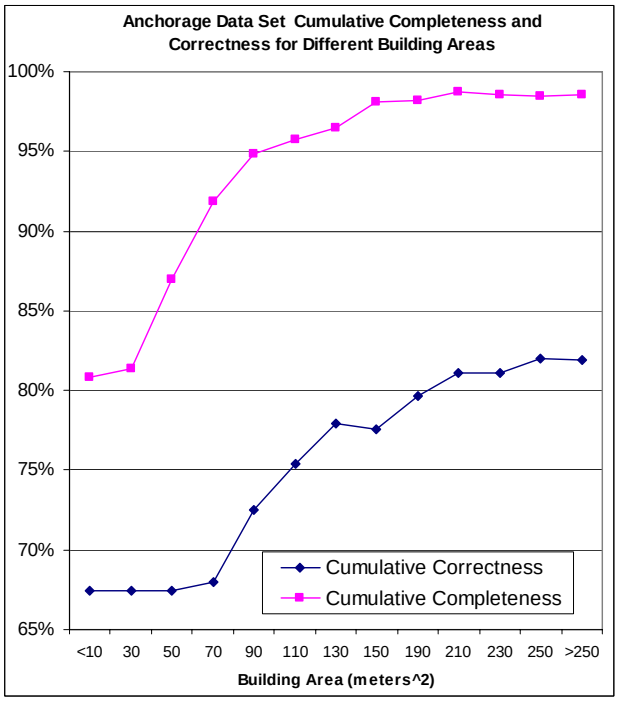


Figure 94: Anchorage Data Set Cumulative Completeness and Correctness

7.6 Comparison to Other Methods for Building Detection from Aerial Imagery

Several approaches in the literature only benchmark their algorithm on less than 250 buildings. The only approach found tested on more than 250 buildings had a 63.60% building correctness. All of the approaches surveyed either didn't report their pixel or their building level correctness; only in this research were both reported. In Table 3 a comparison is provided of all the approaches surveyed. The characteristics listed for each approach are 'Pix Comp', 'Pix Corr', 'Bld Comp', 'Bld Corr' and '# Bldings' which corresponds to pixel level completeness, pixel level correctness, building completeness, building correctness and the number of buildings the results were based upon, respectively. The approaches compared are Lefevre and Webber [13], Muller and Zaum [20], Persson and Sandvall [25], Sirmacek and Unsalan [34], Liu and Prinet [14].

Also included in that comparison is the proposed approach for all buildings, then for buildings having an area of 50m^2 and greater, and finally for buildings having an area of 210m^2 and greater, denoted by 'Shorter all', 'Shorter 50' and 'Shorter 210' respectively. In the cells where there are 'X's, the data was not available or not reported. The results for the proposed approach for all buildings in both data sets at the pixel level completeness are better than Muller, Persson's and Lefevre's. In both the Fairfield and Anchorage data sets, detection rates for buildings smaller than 50m^2 is pretty low. Both data sets have a great deal of homes which have small sheds. The National Association of Home Builders estimated that in 2006 the average area of a new home in the United States was approximately 232m^2 [21]. That is the motivation for including in the table for comparison the mean building completeness and correctness for buildings larger than 50m^2 and then for buildings larger than 210m^2. Notice that for larger than 210 m^2 (which mostly included some large houses and then industrial and commercial buildings) had a building level completeness that was better than two out of the three other approaches surveyed which reported building completeness. .

Approach	Pix Comp	Pix Corr	Bld Com	Bld Corr	#Bldings
Lefevre	63.6%	79.4%	X	X	X
Muller	77.3%	79.5%	X	X	240
Persson	53.0%	93.0%	82.0%	X	17
Sirmacek	X	X	86.6%	X	177
Liu	X	X	94.5%	83.4%	277
Shorter all	78.7%	51.6%	55.4%	48.2%	2643
Shorter 50	X	X	77.3%	64.4%	1414
Shorter 210	X	X	91.8%	44.5%	306

Table 3: Completeness/Correctness at Pixel and Building Level for various Approaches

7.7 Comparison to Other Methods for Building Detection from LiDAR and for Building Detection from LiDAR and Aerial Imagery

Franz Rottensteiner et. al. also used the completeness and correctness performance metrics to evaluate their 'Demster-Shafer' building detection algorithm on the Fairfield data set in [29] and more recently in [30]. They report that the completeness and correctness of their algorithm at the point level is 87% and 91% respectively (in comparison to the proposed algorithm having 92.46% and 81.9% respectively). They report that their algorithm can *reliably* detect buildings with area greater than 110m^2 having both completeness and correctness at the building level greater than 95%. Their algorithm could *mostly* detect buildings having an area greater than 50 m^2 with a completeness of approximately 68%. Buildings smaller than 30m^2 were *not detectable*. For the Fairfield data set, the proposed algorithm can *reliably* detect buildings with an area greater than to 250m^2, having both a completeness and correctness at the building level greater than or equal to 90%. The proposed algorithm can *mostly* detect buildings having an area greater than 90m^2 with a completeness of approximately 81%. Buildings having an area less than or equal to 70 m^2 were *not detectable*. These results are shown in Table 4 where the proposed algorithm is listed as 'PHT' and Rottensteiner et. al's as 'Dempster'.

Algorithm	Reliably	Mostly	Not Detectable
Dempster	> 110 m^2	< 110 m^2 and > 50 m^2	< 30 m^2
PHT	> 250 m^2	< 250 m2 and > 90 m^2	< 70 m^2

Table 4: Comparing PHT vs. Dempster algorithm ability to reliably detect buildings as a function of the building area

Although Rottensteiner et. al's method [30] realizes slightly better building level completeness and correctness for smaller building areas (less than 250m^2), the proposed algorithm performs quite well considering it requires no user intervention, no parameter tuning

for different data sets, and is automatic. Recall from the literature review in section 2.1 that the algorithm in [30] requires a training phase for the user to estimate how much vegetation covers the terrain and that their algorithm will only perform optimally provided that the vegetation coverage estimate the user makes is within 5% of the ground truth. Furthermore, their algorithm makes use of a normalized difference vegetation index (NDVI) image which can only be created provided that the aerial imagery is registered to the LiDAR data. Rottensteiner et. al. [30] report that the inclusion of the NDVI increases the correctness by up to 20% for small to medium-sized buildings. Currently, the automatic registration of aerial imagery to LiDAR data is by far from a trivial process and has most likely been done manually not only in [30] but also in [41]. Also, recall (again as mentioned earlier in section 2.1) that [30] and [16] interpolate the irregularly spaced LiDAR to rasterized/gridded spacings.

Vosselman et. al. in [41]propose a method that also requires registration of the LiDAR to aerial imagery. Using a different data set, they report 85% completeness and 90% correctness at the point level. They conclude that the inclusion of the NDVI image increased their algorithm's accuracy by as much as 3%. Vosselman et. al. in [41] did not however report building level completeness and correctness

Matikainen et. al. in [16] use only LiDAR data for building change detection (when compared to a map) and reports detecting 90% of the building pixels in the map. They evaluate a building level true positive for completeness to have a minimum 70% overlap between a building region in the reference data set and building regions in the automatic set. Also, for a building level true positive for correctness, building regions must have a minimum of 70% overlap between a building region in the automatic data set and building regions in the reference data set. Their method achieves a building level completeness and correctness of 91% and 84%

(respectively) for buildings larger than 200m^2 and for buildings less than 200m^2 42.1% and 23.3% (respectively). The proposed algorithm was re-evaluated per the aforementioned rules for completeness and correctness as defined in [16] which are different than those defined in [29]. The proposed approach realizes a completeness and correctness of 91% and 66% for buildings larger than 190m^2 in the Fairfield data set and 95% and 69% for buildings larger than 190m^2 in the Anchorage data set. For buildings smaller than 190m^2, the proposed approach realizes a completeness and correctness of 63% and 24% for the Fairfield data set and 65% and 55% for the Anchorage data set. Table 5 presents a comparison of the building level completeness (abbreviated as Compl) and correctness (abbreviated as Corr) Matikainen et. al. achieved for their algorithm (displayed as Matikainen) tested on their data set versus the proposed algorithm (displayed as PHT) which was tested on both the Anchorage Data Set (abbreviated as ANC) and the Fairfield Data Set (abbreviated as FF).

Algorithm	>200 m^2	<200 m^2
PHT FF Compl	91%	63%
PHT FF Corr	66%	24%
PHT ANC Compl	95%	65%
PHT ANC Corr	69%	55%
Matikainen Compl	91%	42%
Matikainen Corr	84%	23%

Table 5: Comparing Completeness and Correctness for Buildings having areas greater than and less than 200m^2 for Matikainen's method vs. the proposed method

In [16] the data set the authors tested their algorithm on had only 57 buildings that had an area of less than 200m^2 and 202 buildings had an area greater than 200m^2. Whereas in the Fairfield and Anchorage data sets, there were 1696 and 571 buildings that had areas less than 200m^2 (respectively) and 460 and 374 buildings (respectively) that had areas greater than 200m^2. The proposed approach was tested on two data sets both having significantly more buildings in

the aforementioned ranges than the data set tested in [16]. It should be noted that the data set the authors tested their algorithm on in [16] had a 2-3 points per m^2 whereas the Anchorage data set had only 1 point per 1.5 m^2 (approximately .667 pts per m^2) and the Fairfield data set had only 1 point per 1.2m^2 (approximately .8 points per m^2). The increased point density is a significant factor. Rottensteiner et al in [30] report their completeness and correctness decrease when they decrease the effective point density of their DSM. For buildings having a area of 190m^2 in [30], the completeness for a 1 point per 2m^2 resolution is 93% and for a 1 point per 3m^2 resolution is 73%.

CHAPTER EIGHT: SUMMARY AND CONCLUSIONS

A method for automatically detecting buildings from either aerial imagery or LiDAR data, or both if available, has been proposed, implemented and tested across two real data sets. The proposed approach is flexible because it can utilize either LiDAR or aerial imagery and then improve its accuracy in the event both sources are available. If the LiDAR data is available, the algorithm will triangulate the data and then use the proposed detection methodology, based on the concept of pseudo homotopy trees (developed from topology and set theory), to classify which LiDAR points belong to building, ground or vegetation. If the aerial image data is available, the proposed algorithm uses color invariants to identify shadow and vegetation pixels and then watershed segmentation coupled with a solidity metric to identify building/non-building pixels. If both sources of data are available, the implementation will then automatically detect buildings in both data, register them together and then further improve its classification of the scene depicted by the data. The registration is accomplished by phase correlation of binary building feature masks constructed from the two sources of data. The proposed algorithm is automatic in the sense that it does not require any user intervention, training phases, or parameter adjustment during its execution. The assumptions made are not overly restricting or limiting, thus enabling the approach's applicability to a large variety different building structures. The following sections summarize the contributions realized from this research effort.

8.1 Contributions

In order to overcome limitations in the aerial imagery and LiDAR data, features have to be extracted from the individual sources. To make use of features from both LiDAR and aerial

imagery, the sources have to be registered together. Several approaches in the literature do this manually yet still claim their approach is automatic. It is assumed both sources of data significantly overlap and both contain buildings. Buildings detected from both sources are then used as a common feature in which to align the data sets on top of one another or to describe both data sets with a single coordinate system. Approaches for automatically detecting buildings in the LiDAR data and buildings in a single nadir aerial image are proposed and implemented. An automatic registration approach is proposed and implemented for registering the two sources of data together. Finally, with the sources of data aligned and features extracted from both, an improved classification of the data is provided by the proposed system.

8.1.1 Building Detection from LiDAR

A new method for building detection from irregularly spaced LiDAR is proposed. The proposed method is based on a variation of Homotopy trees which was termed as Pseudo Homotopy Trees. The method has a number of advantages which make it attractive for unsupervised building detection from LiDAR data. One important feature of the algorithm is that it is unsupervised/automatic. By this it is meant there are no training phases, the algorithm does not rely on user input/manual or parameter adjustment. Additionally, the algorithm requires only the raw LiDAR point cloud and can utilize either the first return or the last return pulses or both if available for improved accuracy. Because the method is developed for the irregularly spaced, raw point cloud, it can also work with rasterized/interpolated data without any modifications or preprocessing. The only assumptions employed in the algorithm for detecting building structures is that they occupy a minimum area of 40 m^2 in a nadir view and at least one

of the building's walls abruptly protrudes from the ground with a minimum height of 3m. The algorithm not only classifies collections of connected sets as buildings or non-buildings but it also classifies connected sets which belong to the terrain's ground level. Experiments with real data and comparisons with other proposed methods have shown that the proposed algorithm has competitive performance, despite being completely unsupervised.

The proposed algorithm has some weaknesses too. Buildings are expected to exist as abrupt, concave protrusions from ground planes. In unlikely cases where the orientation of all of the exterior walls of a building slope less than 45 degrees from the vertical, then the proposed method will fail to detect those buildings. Another weakness is that the algorithm simply extracts the buildings just as point clouds, no intelligent measures are made in approximating the building exterior or building contour.

8.1.2 Building Detection from Aerial Imagery

A new method for object classification from a single nadir aerial image is presented. The proposed method uses a novel color quantization technique coupled with a color invariant scheme to identify vegetation. A novel shadow detection procedure is proposed. The distance transform coupled with a thresholded entropy filtered image and watershed segmentation was used to realize texture segmentation. Then the use of solidity was proposed as a metric to identify building regions from the texture segmentation technique. The proposed method was implemented using Matlab and executed the algorithm on a Intel Core 2 Duo (3.0 GHz) machine. The Anchorage data set is a 10896x10896 image and the Fairfield data set is a 13340x 13340

pixel image. It took the algorithm 55 minutes to complete execution on the Anchorage data set and 90 minutes on the Fairfield data set.

The method has several advantages which make it attractive for unsupervised building detection from aerial imagery. One important feature is the algorithm is automatic. In saying it is automatic it is meant that there are no training phases, the algorithm does not rely on user input nor parameter adjustment. Another advantage is the only assumptions employed by the algorithm in regards to building structures are that no shadows are cast upon the tops of buildings and that the building textured rooftop segments exist as convex hulls. The limited assumptions made about buildings enable the proposed algorithm to detect a variety of buildings exhibiting different spectral and structural characteristics. The algorithm is a tertiary classifier capable of classifying objects as non-building and shadow (which shadow assumed to be non-building), building, and vegetation. The method only requires a single nadir aerial image for its input. Experiments with real data and comparisons with other proposed methods have shown that the proposed algorithm has competitive performance, despite being completely unsupervised while not implementing overly restricting assumptions about building structures (such as buildings having only red roof tops or having rectangular exteriors).

The proposed method has some weaknesses as well. Due to the fact that the method is automatic, works off of only a single nadir aerial image, and does not make overly restricting assumptions about building structures, its correctness measure is mediocre. However, it should be noted its difficult to comparatively rate the correctness measure, both on the pixel and building level as all methods surveyed in this paper only reported one or the other but not both.

8.1.3 Automatic Registration of LiDAR Data to Aerial Imagery

A new method for the automatic registration of LiDAR data to a single nadir aerial image is proposed. The method employs the area based phase correlation for registration of the two different sources of information. Because the two images are captured by two different sources, the proposed pre-processing is implemented. Buildings are automatically detected in both images and then binary building masks are created and their phases correlated to automatically extract the parameters of the geometric transformations to align the sources so that they can be described by a single coordinate system. The only assumptions made by this registration approach are that the two sources of data both significantly overlap in their coverage of the same scene and only differ via translation, rotation and scaling. The advantages of the registration method include it being automatic and being able to handle a certain percentage of buildings not being detected in either source. The phase correlation approach automatically extracts the parameters corresponding to the maximum correlation for the two binary building masks. So even if the masks are not exactly the same (as in some false positives or false negatives exist in one mask that are not in the other), the approach can still work. The trade off to this is there are several conditions that can factor into the algorithm not converging on the correct geometric transformation parameters resulting in an inaccurate registration. A sharper correlation peak or simply put a better chance that the registration algorithm will converge on the correct parameters happens when there is more overlapping coverage between the two images; the rotation, translation and scalings between the two images is smaller; and the binary images being correlated are closely alike. A broader correlation peak is produced or the chance is lowered that the registration algorithm will correctly converge when there is less overlap between the binary

images, when the rotation, translation and scalings between the two images are large and the binary images look very dissimilar.

8.1.4 Vegetation Identification from Aerial Imagery

The proposed vegetation detection realizes a 97% correctness on accurately identifying non-building pixels (as building and vegetation are mutually exclusive classes). It only requires the use of a single, nadir aerial image and is completely automatic. It is shown the proposed vegetation identification technique out-performs manually thresholding the pseudo NDVI image in terms of correctness.

8.1.5 Improved Classification

Combining features from the aerial image and LiDAR data enabled the proposed implementation to correctly label vegetation which was being mislabeled as building by the building detection from LiDAR algorithm. These large false positives (vegetation grass patches) being removed caused the algorithm's per point correctness to slightly increase. Furthermore, combining the features enabled the algorithm to now identify an additional class of objects in the image, grass vegetation. In the building detection from LiDAR approach, the ground class encompassed grass, pavement, and dirt. Now, with the approach using both LiDAR data and aerial imagery, an image is classified into building, non-building (mostly trees), grass vegetation and ground (which only includes pavement and dirt).

8.1.6 System Implementation

The proposed unsupervised building detection from LiDAR and aerial image implementation has been developed with an emphasis on automation and flexibility to the end user. The implementation goes to great lengths with providing convenience to the user by incorporating the following. First, if LiDAR data is provided, then buildings are automatically detected from that LiDAR data. If that LiDAR data has both first and last returns, then the algorithm will make use of both returns to improve its accuracy. If only first or last returns are provided, the algorithm will still be able to detect buildings. No preprocessing is necessary if the LiDAR data is interpolated or irregular. If the user has aerial imagery, buildings are automatically detected from the aerial imagery. If the user has multiple data sets, buildings can be detected from those multiple data sets without any parameter adjustment as the algorithm works on different data sets. If the user has both LiDAR data and aerial imagery, the proposed approach will automatically detect buildings in both and automatically register those images together. It is shown that the two sources of data do not necessarily have to completely overlap and can differ via translation, rotation and scaling geometric transformations. Finally, when features from both sources of data are used, classification is further improved by identifying grass vegetation in the LiDAR data.

8.2 Future Work

There are several directions in which one could take to further this research. As mentioned in section 1.3, there are a variety of different noise sources which corrupt the procurement process of the LiDAR data. It would be interesting to investigate a method for

removing that noise, both in the sensors and accounting for the fact that the atmosphere distorts the path of the laser beam emitted from the LiDAR sensor. Furthermore, it would be desirable to know how much can be gained in terms of accuracy for building identification as well as reconstruction as a result of removing that noise.

Lines in the LiDAR data are not as easily extracted or identified as they are in the aerial imagery due to sensor resolution. It would therefore be advantageous to use the aerial imagery to construct polygonal representations of the identified building's exterior so that the buildings are instead represented as shapes instead of a collection of points. Furthermore, it may be possible that reconstruction with using features from both sources of data may be more accurate than attempting to reconstruct buildings from just the LiDAR data.

With the buildings having been identified manually in both the LiDAR data and the aerial imagery, it is now possible to use some of those manually identified buildings as training features for an area based, supervised building detection approach. With two data sets, one could test the algorithm with a small percentage of buildings from one data set and then observe its performance on the other. It would be desirable to have a supervised, semi-automatic algorithm which would only have to be trained once and then capable of identifying buildings across several data sets without having to be retrained. However, innovation would be necessary in coming up with data set invariant features in which to train the supervised classifier as well as picking the appropriate classifier for this application.

One possible avenue to explore to increase the registration algorithm's accuracy is correlating the magnitude spectrum as well as the phase spectrum. Furthermore, it may be possible to gain performance increase by instead of correlating the entire building mask to instead evenly partition the masks and correlate those partitions. Finally, it is possible up-

sampling the LiDAR data to a certain degree as well as down-sampling the aerial imagery less, resulting in higher resolution building masks may also help the registration algorithm's ability to converge on the correct registration parameters. Furthermore, instead of correlating building masks, it may be easier to correlate the detected ground from the LiDAR data with the vegetation and non-building from the aerial imagery. Or it may also be easier correlating other features such as the distance transform of detected lines in both sources.

Finally, there are a plethora of other sources existent in remote sensing other than just aerial imagery and LiDAR data taken from a nadir perspective. It would be interesting to explore the possibility of also combining stereo pairs of aerial images, LiDAR captured from the ground, and multi-spectral images.

APPENDIX A:
PHT GENERATION PLOTS AND HEURISTIC EFFECTIVENESS

One of the advantages of the PHT algorithm is the ability to produce generation plots showing how the connected sets are related to one another as a function of their generation. Figure 95 depicts an aerial image portion of the Anchorage data set containing a several commercial buildings and a park.

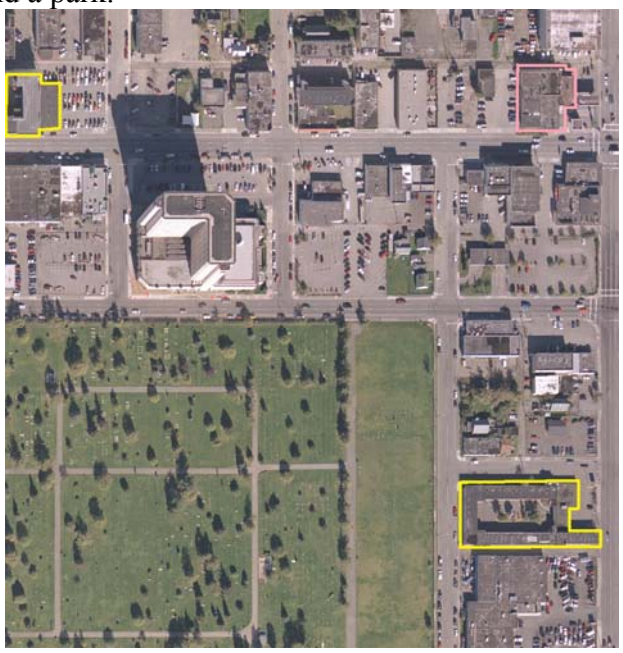


Figure 95 - Aerial Image Portion of Anchorage Data Set

Figure 96 is the building/non-building/ground classification output of the PHT algorithm where red triangles correspond to what the algorithm has classified as building, light green triangles correspond to what the PHT algorithm has classified as non-building and dark blue triangles correspond to what the PHT algorithm has classified as ground. Figure 98 is simply an angled version of Figure 96. Figure 97 is the generation plot where the colder color of the triangle, the older in generation the connected set that triangle belongs to and the hotter the color of the triangle corresponds to the younger in generation the connected set the triangle belongs to. Note that the oldest generation in this particular case does in fact correspond to a ground connected set and that ground connected set encompasses the other connected sets. In this case, the pseudo homotopy tree generated does in fact generalize to being an actual homotopy tree because the root node or oldest generation encompasses all younger generations. Furthermore, observe how the building encircled in pink (in the North East corner) in Figure 95 has its roof in the generation plot in Figure 97 mistakenly belonging to the same connected set as the nearby ground (ie the roof is dark blue instead of light green like the tops of other buildings). This is because of the clutter existent on the side of the building has made a ramp allowing the ground connected set surrounding the building to bleed or region grow onto the roof. Yet observe how in Figure 96 the building is still correctly identified. Heuristic 6, in Chapter 3, section 3.4 is what enables the algorithm to still correctly detect the building. Notice how the buildings in encircled in yellow in Figure 95 both have court yards. Now notice that those buildings' court yards in Figure 96 are correctly labeled as ground yet notice in Figure 97 those buildings' court yards belong to a younger connected set with few to no children. Heuristic 5 in Chapter 3,

section 3.4 enables the PHT algorithm to correctly identify courtyards as ground connected sets. Finally, notice that all the vegetation to the south west in Figure 95 and notice that only a single false positive is present in Figure 96 erroneously labelling a piece of vegetation as building. Notice how in Figure 97 and Figure 98 that many of those trees either don't have any NWC set children or if they do have NWC set children, the number of triangles composing those NWC sets is in fact too few to enable that vegetation as being classified as building by the PHT algorithm.

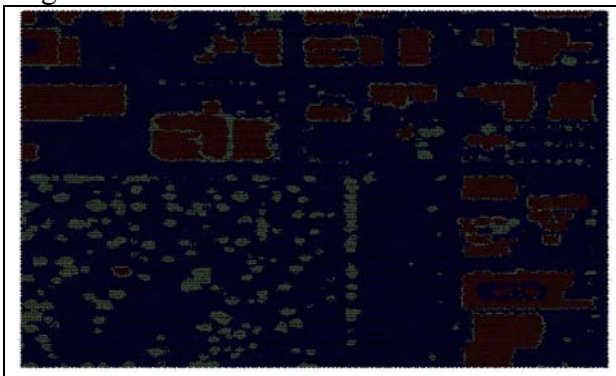


Figure 96 - Building/Nonbuilding/Ground PHT Classification Triangulation Plot

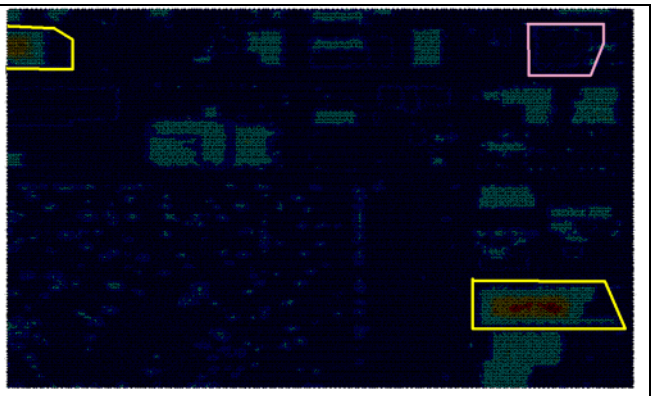


Figure 97 - PHT Generation Triangulation Plot

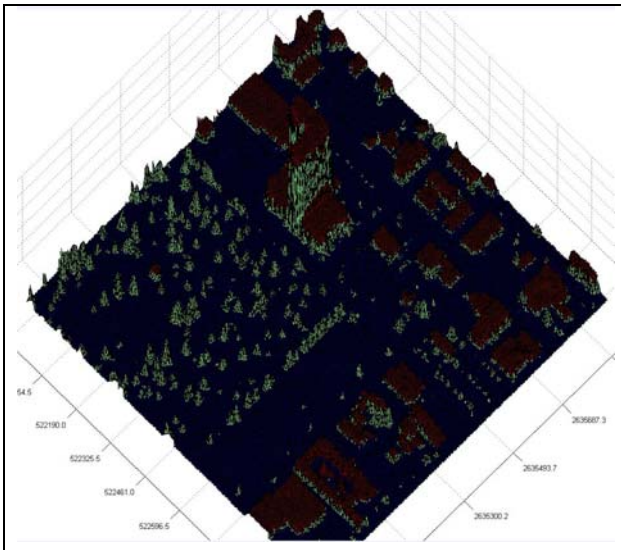


Figure 98 - Angled Building/Non-building/Ground PHT Classification Triangulation Plot

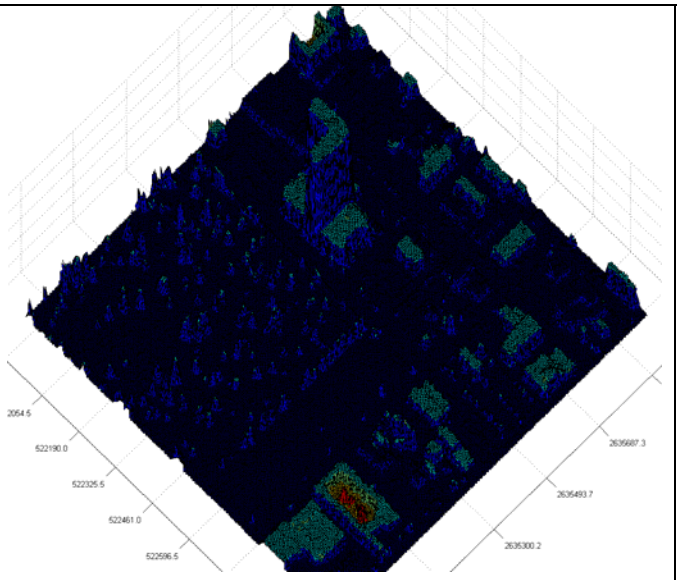


Figure 99 - Angled Generation Triangulation Plot

Figure 100 depicts an aerial image portion of the Fairfield data set. The areas encircled in yellow are courtyards which are surrounded by building and the building encompassed in pink you'll notice has a loading ramp leading from the ground to its roof. Furthermore, you'll notice

that this portion of the data set is actually split in half by the forest filled ravine that partitions sections of the data set.

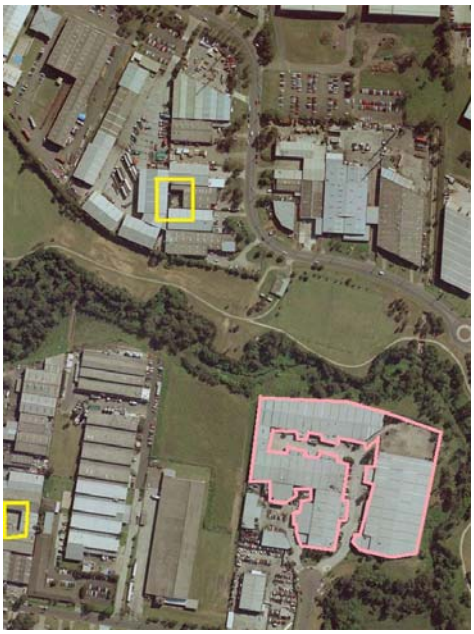


Figure 100 - Aerial Image Portion of Fairfield Data Set

The PHT algorithm building/non-building/ground classification output for the terrain depicted in Figure 100 is shown in Figure 101 from a nadir perspective and then again in Figure 103 from an angled perspective. The generation plot for the terrain depicted in Figure 100 is shown in Figure 102 from a nadir perspective and then again in Figure 104 from an angled perspective. Notice that in Figure 100, the old generation belongs to a ditch in the center of the data set. This ditch does not encompass all of the other connected sets, therefore the pseudo homotopy tree generated from the terrain depicted in Figure 100 would not generalize to a normal homotopy tree. Furthermore, notice that even though this portion of the data set is partitioned into multiple ground planes, the algorithm is still capable of identifying all of the ground planes. Notice how a ground connected set exists to the south of the forest that partitions the data set in the center, to the north of that forest, and finally another ground connected set to the north west in Figure 102. One can see in Figure 102 that in the upper North West corner, the ground is colored with a green non-wall connected set, which is of a different generation from the older, light blue ground connected sets elsewhere in the data set. Finally, the oldest generation, the dark blue connected set, exists as a ditch in the center of the generation triangulation plot in Figure 102 to the east (right above the pink encircled building). As with the Anchorage data set, notice how heuristic 6 enables the algorithm to detect the building encompassed in pink in Figure 100 and that heurstic 5 enables the algorithm to correctly label the court yards encompassed in yellow in Figure 100 as ground connected sets in the Fairfield data set. Finally, notice with of all the vegetation existent in the center of the portion of the data set, only a small portion of it comes up as erroneously labeled as building by the PHT algorithm in Figure 101.

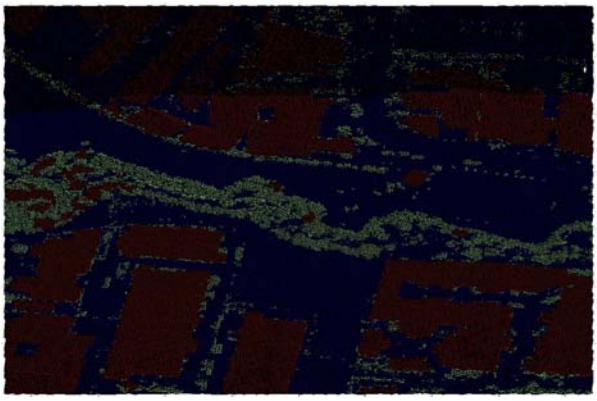


Figure 101 - Building/Non-building/Ground PHT Classification Triangulation Plot

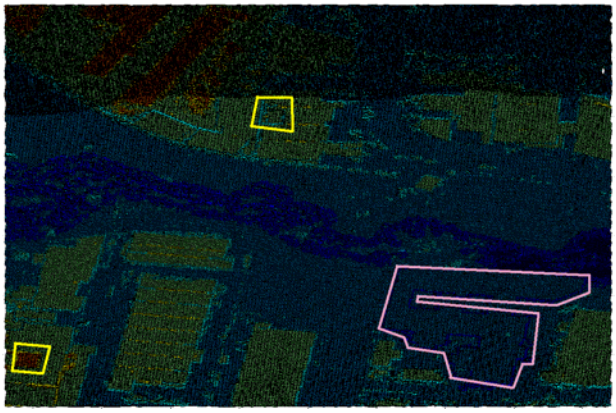


Figure 102 - PHT Generation Triangulation Plot

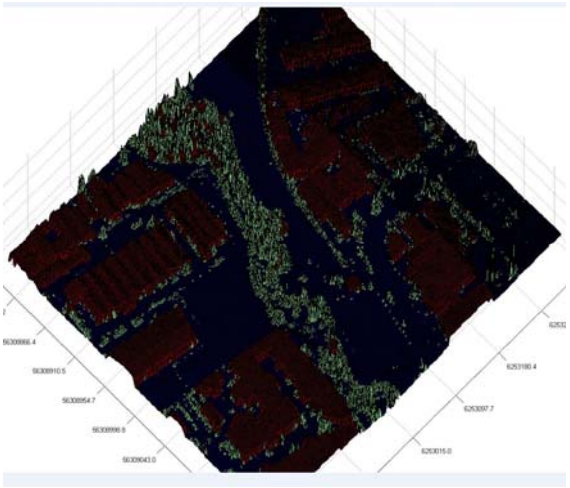


Figure 103 - Angled Building/Non-building/Ground PHT Classification Triangulation Plot

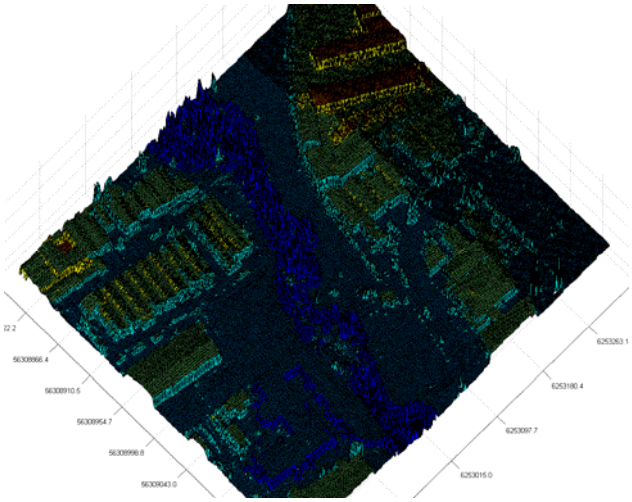


Figure 104 - Angled Generation Triangulation Plot

APPENDIX B:
FORWARD AND REVERSE PROOFS OF FOURIER SHIFT PROPERTY

The following is a proof for the Fourier Shift Property showing that a shift in an image's phase corresponds to a linear translation in the Cartesian image space :
Assume **f** is an image where f(x,y) returns the intensity of the image at row x and column y where x∈{1,...,M} and y∈{1,...,N}.

The objective is to prove the following:

$$F^{-1}\left\{F(u,v)\cdot e^{-j\cdot 2\cdot\pi\cdot\left(\frac{u\cdot x_0}{M}+\frac{v\cdot y_0}{N}\right)}\right\}=f(x-x_0,y-y_0) \tag{47}$$

Consider the 2D-Discrete Inverse Fourier Transform:

$$F^{-1}\{F(u,v)\}=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\cdot e^{j\cdot 2\cdot\pi\cdot\left(\frac{u\cdot x}{M}+\frac{v\cdot y}{N}\right)}=f(x,y) \tag{48}$$

Then, taking the inverse 2D-Discrete Fourier Transform of the exponential:

$$F^{-1}\left\{F(u,v)\cdot e^{-j\cdot 2\cdot\pi\cdot\left(\frac{u\cdot x_0}{M}+\frac{v\cdot y_0}{N}\right)}\right\}=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\cdot e^{-j\cdot 2\cdot\pi\cdot\left(\frac{u\cdot x_0}{M}+\frac{v\cdot y_0}{N}\right)}\cdot e^{j\cdot 2\cdot\pi\cdot\left(\frac{u\cdot x}{M}+\frac{v\cdot y}{N}\right)} \tag{49}$$

Then combining terms in the exponential:

$$F^{-1}\left\{F(u,v)\cdot e^{-j\cdot 2\cdot\pi\cdot\left(\frac{u\cdot x_0}{M}+\frac{v\cdot y_0}{N}\right)}\right\}=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\cdot e^{j\cdot 2\cdot\pi\cdot\left(u\cdot\frac{x-x_0}{M}+v\cdot\frac{y-y_0}{N}\right)}=f(x-x_0,y-y_0) \tag{50}$$

And finally simplifying yields:

$$F^{-1}\left\{F(u,v)\cdot e^{-j\cdot 2\cdot\pi\cdot\left(\frac{u\cdot x_0}{M}+\frac{v\cdot y_0}{N}\right)}\right\}=f(x-x_0,y-y_0) \tag{51}$$

As shown above then, the inverse Fourier transform of a linear phase shift corresponds to a linear translation in the Cartesian image space.

The following is a proof for the Fourier Shift Property showing that a linear translation in the Cartesian image space corresponds to a linear shift in the image's phase in the discrete Fourier space:
The objective is to prove the following:

$$F\{f(x-x_0,y-y_0)\}=F(u,v)\cdot e^{-j\cdot 2\cdot\pi\cdot\left(\frac{u\cdot x_0}{M}+\frac{v\cdot y_0}{N}\right)} \tag{52}$$

Consider the 2D-Discrete Forward Fourier Transform:

$$F\{f(x-x_0,y-y_0)\}=\frac{1}{M\cdot N}\cdot\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\cdot e^{-j\cdot 2\cdot\pi\cdot\left(\frac{u\cdot x}{M}+\frac{v\cdot y}{N}\right)} \tag{53}$$

Substitution: $m \to x - x_0$ and $n \to y - y_0$

Calculating x and y in terms of m and n: $x = m + x_0$ and $y = n + y_0$

Calculating limits in terms of m and n:

Lower Limits: $x = 0 \to m = -x_0$ and $y = 0 \to n = -y_0$

Upper Limits: $x = M - 1 \to m = M - 1 - x_0$ and $y = N - 1 \to n = N - 1 - y_0$

Then equation (53) becomes:

$$F\left\{f\left(x - x_0, y - y_0\right)\right\} = \frac{1}{M \cdot N} \cdot \sum_{m=-x_0}^{M-1-x_0} \sum_{y=-y_0}^{N-1-y_0} f\left(m,n\right) \cdot e^{-j \cdot 2 \cdot \pi \cdot \left(u \cdot \frac{m+x_0}{M} + v \cdot \frac{n+y_0}{N}\right)} \tag{54}$$

The summation in (63) is rewritten as follows:

$$F\left\{f\left(x - x_0, y - y_0\right)\right\} = \frac{1}{M \cdot N} \cdot \left[\begin{array}{l} \sum_{m=-x_0}^{0} \sum_{n=-y_0}^{0} f\left(m,n\right) \cdot e^{-j \cdot 2 \cdot \pi \cdot \left(u \cdot \frac{m+x_0}{M} + v \cdot \frac{n+y_0}{N}\right)} \\ + \sum_{m=0}^{M-1-x_0} \sum_{n=0}^{N-1-y_0} f\left(m,n\right) \cdot e^{-j \cdot 2 \cdot \pi \cdot \left(u \cdot \frac{m+x_0}{M} + v \cdot \frac{n+y_0}{N}\right)} \end{array}\right] \tag{55}$$

It is then assumed image f(x,y) is periodic with periods M and N:

$$f\left(x + M, y + N\right) = f\left(x, y\right) \tag{56}$$

With periodicity M and N for indices x and y, the following is observed:

$$\sum_{m=-x_0}^{0} \sum_{n=-y_0}^{0} f\left(m,n\right) \cdot e^{-j \cdot 2 \cdot \pi \cdot \left(u \cdot \frac{m+x_0}{M} + v \cdot \frac{n+y_0}{N}\right)} = \sum_{m=M-1-x_0}^{M-1} \sum_{n=N-1-y_0}^{N-1} f\left(m,n\right) \cdot e^{-j \cdot 2 \cdot \pi \cdot \left(u \cdot \frac{m+x_0}{M} + v \cdot \frac{n+y_0}{N}\right)} \tag{57}$$

Rewriting the summation:

$$F\left\{f\left(x - x_0, y - y_0\right)\right\} = \frac{1}{M \cdot N} \cdot \left[\begin{array}{l} \sum_{m=0}^{M-1-x_0} \sum_{y=0}^{N-1-y_0} f\left(m,n\right) \cdot e^{-j \cdot 2 \cdot \pi \cdot \left(u \cdot \frac{m+x_0}{M} + v \cdot \frac{n+y_0}{N}\right)} \\ + \sum_{m=M-1-x_0}^{M-1} \sum_{y=M-1-y_0}^{N-1} f\left(m,n\right) \cdot e^{-j \cdot 2 \cdot \pi \cdot \left(u \cdot \frac{m+x_0}{M} + v \cdot \frac{n+y_0}{N}\right)} \end{array}\right] \tag{58}$$

Combining the summations:

$$F\left\{f\left(x - x_0, y - y_0\right)\right\} = \frac{1}{M \cdot N} \cdot \left[\sum_{m=0}^{M-1} \sum_{y=0}^{N-1} f\left(m,n\right) \cdot e^{-j \cdot 2 \cdot \pi \cdot \left(u \cdot \frac{m+x_0}{M} + v \cdot \frac{n+y_0}{N}\right)}\right] \tag{59}$$

Rewriting the Exponential:

$$F\left\{f\left(x - x_0, y - y_0\right)\right\} = \frac{1}{M \cdot N} \cdot \left[\sum_{m=0}^{M-1} \sum_{y=0}^{N-1} f\left(m,n\right) \cdot e^{-j \cdot 2 \cdot \pi \cdot u \cdot \frac{m+x_0}{M}} \cdot e^{-j \cdot 2 \cdot \pi \cdot v \cdot \frac{n+y_0}{N}}\right] \tag{60}$$

Moving Terms outside the summation which are independent of m and n:

$$F\left\{f\left(x-x_0,y-y_0\right)\right\}=\frac{1}{M\cdot N}\cdot\left[\sum_{m=0}^{M-1}\sum_{y=0}^{N-1}f\left(m,n\right)\cdot e^{-j\cdot 2\cdot\pi\cdot\left(\frac{u\cdot m}{M}+\frac{v\cdot n}{N}\right)}\right]\cdot e^{-j\cdot 2\cdot\pi\cdot\left(\frac{u\cdot y_0}{N}+\frac{x\cdot y_0}{N}\right)} \tag{61}$$

Substituting the summation as it is the definition of the Fourier Transform yields:

$$F\left\{f\left(x-x_0,y-y_0\right)\right\}=F\left(u,v\right)\cdot e^{-j\cdot 2\cdot\pi\cdot\left(\frac{u\cdot y_0}{N}+\frac{x\cdot y_0}{N}\right)} \tag{62}$$

APPENDIX C:
SEQUENTIAL GREEDY INSERTION STEPS

Many of the application specific needs will ultimately determine the nature of the triangulation algorithm chosen. The definition of these needs will therefore reduce the number of algorithms that will be choice for the problem at hand. In order to retain the most amount of information and accuracy as possible, it is imperative that the TIN is derived from the raw LIDAR point cloud data. The selected triangulation algorithm therefore must have high accuracy in approximating the raw LIDAR point cloud data with the implemented TIN. The triangles in the TIN are to be clustered by a clustering algorithm. The dimensions of the triangles that are of importance to the clustering algorithm are as follows: the triangles' vertices, their centers, and the normal angle to their defined surface. These dimensions therefore must be incorporated in the data structure encoding the resulting TIN.

In order to design or select an algorithm, the necessary rules for the desired triangulation must be specified:

1. No intersecting triangle edges are to exist within the TIN.

2. Furthermore, no overlapping triangles are to exist within the TIN.

3. No gaps are permissible within the TIN.

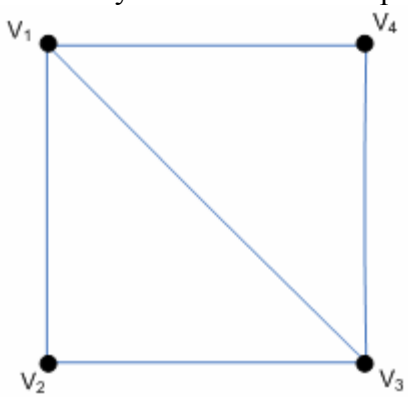
4. When considering a point for the formation of a triangle, the neighboring points closest to the point in consideration must have the highest favored potential for triangle formation.

5. As a result from rules 1 and 2, from a top down view, all triangles must be visible. Therefore, the formation of triangles in 3-dimensional space, surfacing over triangles underneath, is prohibited.

The algorithm selected to realize the triangulation of the irregular point spacings in the provided LIDAR data is Garland and Heckbert's sequential greedy insertion algorithm. In [6], Garland and Heckbert present both the sequential and parallel greedy insertion algorithms. The version of the greedy insertion algorithm, which only inserts a single point in each pass is called sequential greedy insertion, while the version of the algorithm in which inserts multiple points in each pass is called parallel greedy insertion. While the parallel version does cut down execution

time, the savings realized come at the cost of the algorithm's performance in terms of accuracy; which is why the sequential version is selected.

The sequential greedy insertion algorithm simultaneously optimizes two adaptive optimization cost functions: (1) local Delaunay triangulation; (2) global point insertion. The algorithm starts by considering the quadrilateral formed by the outermost four points in terms of x and y or longitude and latitude spacing. Then an arbitrary triangulation is formed (two triangles are randomly formed from the 4 points).



$$V_4 = <\max(x_i), \max(y_i)>$$
$$V_1 = <\min(x_i), \max(y_i)>$$
$$V_2 = <\min(x_i), \min(y_i)>$$
$$V_3 = <\max(x_i), \min(y_i)>$$

Figure 105: Initial Triangulation

That formation is then checked to see if flipping the diagonal will optimize the arbitrarily formed configuration to conform to Delaunay triangulation.

For all triangles, the distances between the triangles (planes) Figure 106 and the points that they encompass (in x and y or longitude and latitude spacing) are calculated.

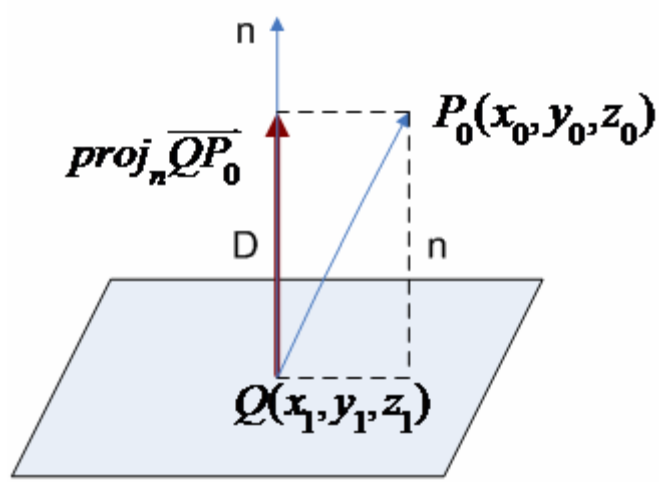


Figure 106: Distance between a given point and a plane

After all of the distances between the unused points and the existing triangulated surface are calculated, for each triangle, the unused point furthest from that triangle is cached into that triangles data structure.

All of the LIDAR points that are considered, the point having the greatest distance from the TIN (labeled the candidate point) is the point inserted next (hence the name greedy insertion). Three cases can occur when inserting a given point: (1) the candidate point is inserted inside a triangle; (2) the candidate point is inserted at the edge of the outermost initial quadrilateral; and (3) the candidate point is inserted on a triangle edge.

The first point insertion case results in the formation of three triangles. The point is inserted and three lines are drawn from the point to the vertices of the encompassing triangle. This scenario is depicted in Figure 107.

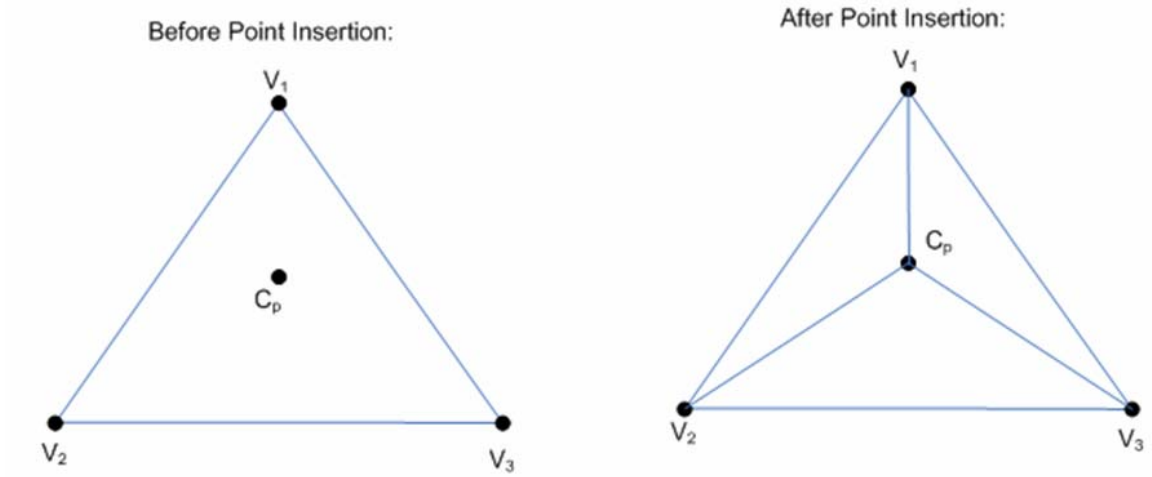


Figure 107: Point Insertion (Case 1)

For the second point insertion case, the candidate point is inserted at the edge of the TIN resulting in the formation of 2 new triangles, as depicted in Figure 108.

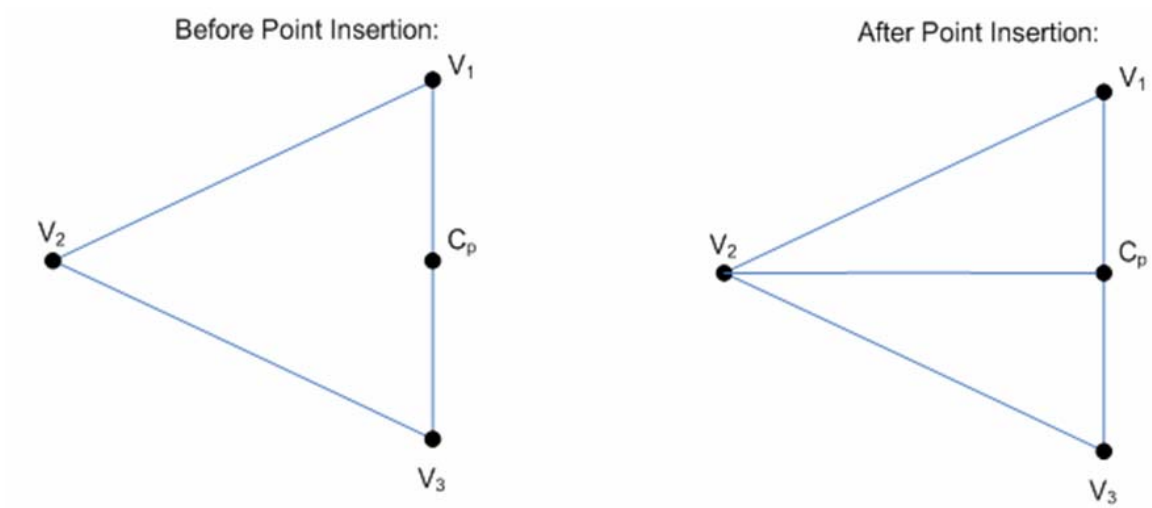


Figure 108: Point Insertion (Case 2)

In the third point insertion case, the candidate point is inserted along the edge of a triangle. The algorithm is designed to delete the edge and then connect lines from the candidate point to the

vertices of the two triangles which share the common edge in which the candidate point was inserted along. The third point insertion scenario is depicted in Figure 109.

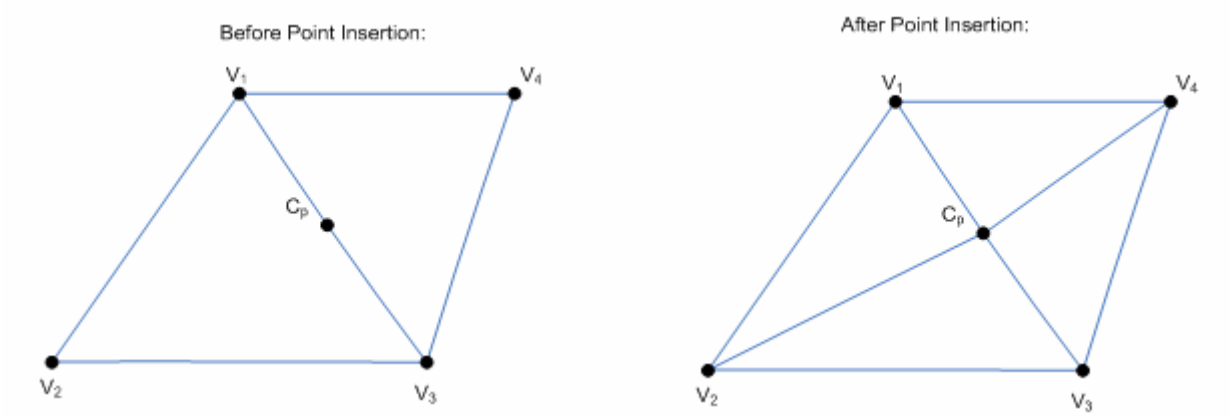


Figure 109: Point Insertion (Case 3)

After the insertion of the points, the edges of the triangles are checked for flipping. The edges are flipped to form a new diagonal if the flipping maximizes the lesser of the interior angles of the triangles (Delaunay triangulation). If for two given triangles, their edges are flipped, then all of the adjacent triangles to those triangles are then checked to see if edge flipping should be done with triangles adjacent to them. This process continues until it is determined that no adjacent triangle will further optimize the TIN via diagonal flipping in accordance to Delaunay triangulation. This local optimization procedure is implemented to combat the formation of slivers. A sliver is qualitatively defined as a triangle whose largest angle is 'relatively close' to 180 degrees. Therefore, triangle 'B' depicted in Figure 110 is desired over triangle 'A'.

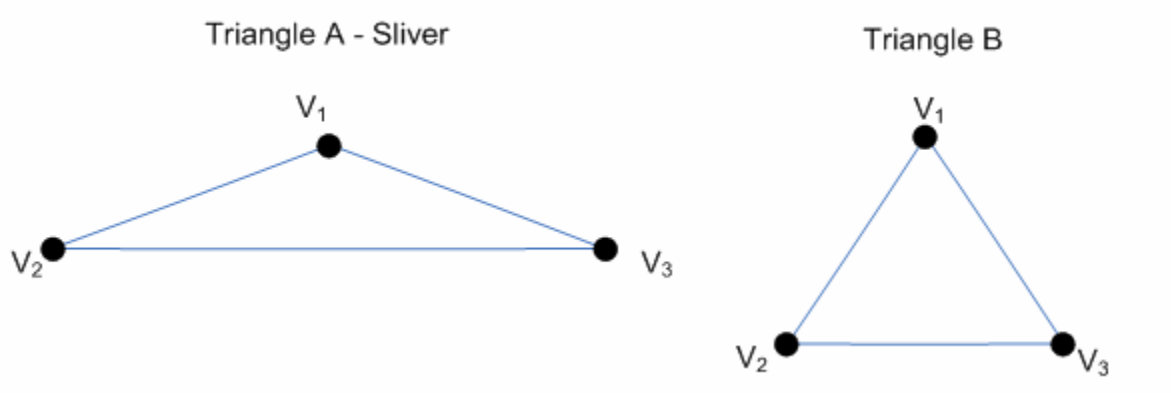


Figure 110: Sliver Example

All of the above procedures are depicted in the block diagram contained in Figure 111.

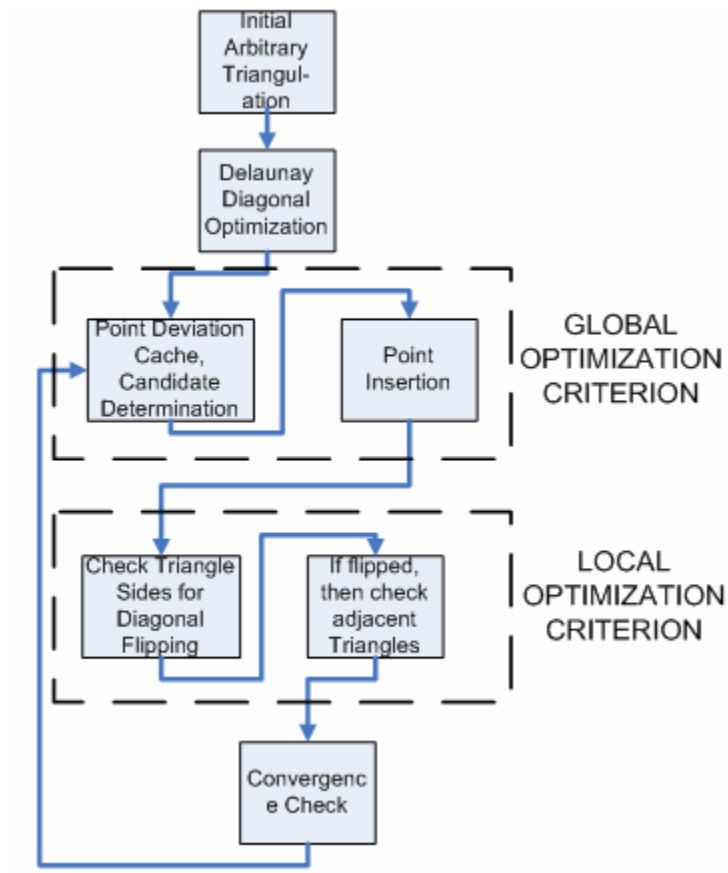


Figure 111: Greedy Insertion Block Diagram

A step by step description of the Greedy Insertion Triangulation Algorithm is as follows:

Step 1 - Initial Triangulation

Step 1a – Select the 4 outermost corner points of the LIDAR data (some points may be artificially created)

Step 1b – Perform Delaunay triangulation of selected 4 points (2 triangles formed) swapping the edges to obtain the optimal mesh

Step 1c – Mark the 4 points as used

Step 1d – For each of the two triangles formed, calculate the distance between the unused points and the plane formed by the triangle encompassing those unused points in x and y dimensions

Step 1d (i) – Cache the candidate point (point farthest away from triangle in z-direction) for each triangle formed

Step 2 - Largest Deviation Point Insertion

Step 2a - Select the candidate point (the point with largest deviation from triangulated mesh). Note: if this is the first iteration of the algorithm, all errors must be calculated as none are cached

Step 2b – Insert the Point into the Triangulated Mesh (mark it as used)

Step 3 – Locate and Flip if Necessary

Step 3a – Locate the triangle within the triangulated mesh containing the recent inserted point

Step 3b – Split the located triangle into the necessary triangles containing the inserted point (based on the condition of insertion – [Figure 107],[Figure 108], or [Figure 109])
Step 3c - Remove the original triangle (triangles are not allowed to overlap one another)
Step 3d – Recursively check each of the outer edges of the triangle containing the inserted point to see if flipping the edges will further optimize the existing triangulated mesh.
If a triangle edge is flipped, check the edges of both of those triangles and see if their adjacent triangle diagonals should be flipped (repeat until flipping will no longer further optimize the TIN according to local cost function).
Step 4– In the regions affected by insertion and flipping, recalculate the following parameters
Step 4a - The plane equations associated with the modified triangulations
Step 4b – Locate the triangles containing the unused points
Step 4c – Calculate the error between the unused point and the triangulated surface
Step 4d – For each triangle record the candidate value for the unused points (point with largest error deviation)
Step 5 - Return to step 2 and repeat if point budget or error approximation has not been met
If convergence in Step 5 was realized, finish inserting points and remove all triangles associated with artificial points (effectively removing those points from the triangulation)

The elevation coordinates of the LIDAR data are actually only accurate to a certain order of magnitude (in the order of centimeters). Making matters worse, the LIDAR data suffers from systematic errors and noise. Therefore, noise is existent in the data and presents difficulties for coplanar clustering based on the normal vectors of the triangles existent in the TIN generated from the raw LIDAR. An ideal set of coplanar triangles Figure 112, actually exist as points jittering about that plane, as shown in Figure 113. The noise causes the LIDAR points to deviate from the ideal plane, thereby causing the normal vectors of the triangles to deviate from their ideal directions.

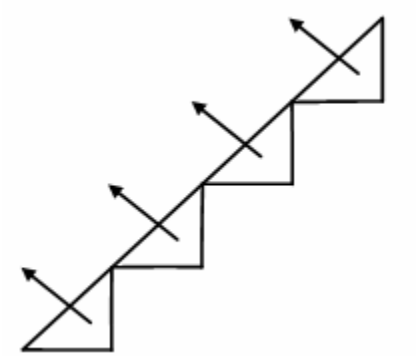


Figure 112: Ideal LIDAR Points

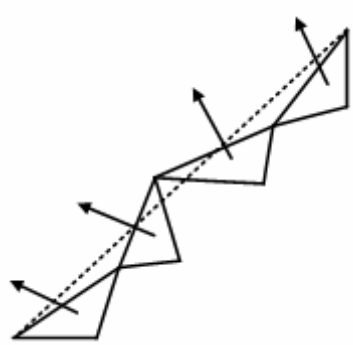


Figure 113: Actual LIDAR Points

One way to filter these errors would be to exploit the very nature of the triangulation algorithm selected. Sequential greedy insertion inserts the points farthest away from the initial plane established. Therefore points along roof ridges, roof corners, and building edges are the points inserted first. The points inserted last are the points closest to an established plane, the points with the smallest errors. It is possible to simply program the sequential greedy insertion triangulation algorithm to only triangulate points above a certain error threshold. However, the insertion of fewer points leads to a less accurate TIN and furthermore, leads to fewer triangles sharing the same plane. Rather than not inserting the triangles, leading to fewer members of a

given coplanar cluster, it would be advantageous to correct the inaccuracies of the points along the z or height dimension. Since the points, which jitter about the already established roof plane, are contained in a well defined plane, it is possible to remove the jitter or systematic error or noise by placing points below a certain threshold distance on the plane in which they are contained. While the longitude and latitude dimensions were preserved, the elevation dimension of a candidate point was modified if the candidate point met the following conditions: the perpendicular distance, defined in equation (63), of the candidate point was less than .2 meters ($D_c \leq .2m$) from the containing triangulated plane; and the pitch of the roof , defined in equation (64) was less than 60 degrees ($\theta \leq 60^o$).

$$D_c = \frac{|a\cdot(x_0 - x_1) + b\cdot(y_0 - y_1) + c\cdot(z_0 - z_1)|}{\sqrt{a^2 + b^2 + c^2}} \tag{63}$$

$$\theta = 90 - \sin^{-1}\left(\frac{D_c}{Z_c}\right) \tag{64}$$

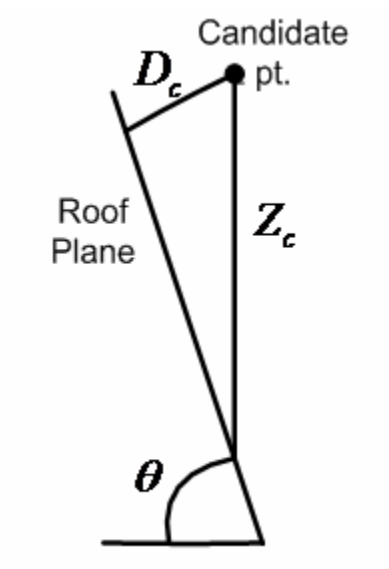


Figure 114 - Pitch of Roof Plane (Theta)

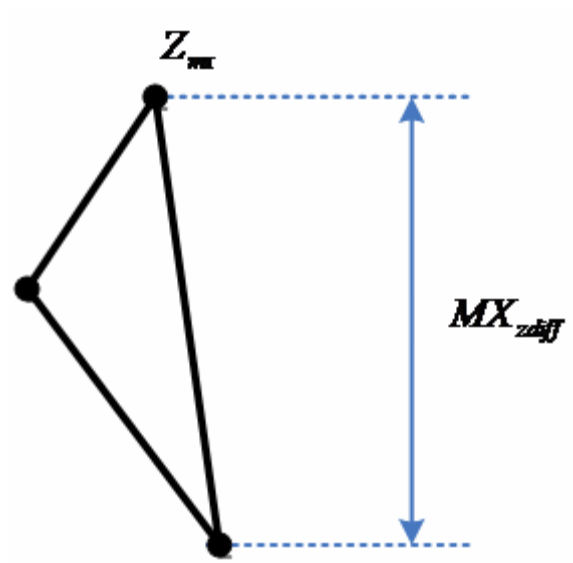


Figure 115 - Triangle Elevation Difference

Without the constraint imposed on the roof plane pitch ($\theta \leq 60^o$), building edge points, which were not yet inserted/triangulated and less than .2 meters perpendicular distance from the building were being merged into the building's edge, thus distorting the building outline. The second constraint therefore confines points which only exist on a plane with a pitch ($\theta \leq 60^o$) to become merged with that existent roof plane. Most of the building structures existent in the data set considered had roof planes with pitches less than 60 degrees.

This filtering technique was found to remove the noise depicted in Figure 113.

APPENDIX D:
RESEARCH WEBSITE

This dissertation in its entirety in addition to other materials related to Nicholas Shorter's research are hosted online at the following location: http://www.nshorter.com

APPENDIX E:
WORDS OF INSPIRATION FROM EISENHOWER

The following speech, in which Eisenhower delivered to troops heading out to the D-Day invasion, has inspired me to great lengths during the most difficult times of my academic career:

You are about to embark upon the Great Crusade, toward which we have striven these many months. The eyes of the world are upon you. The hope and prayers of liberty-loving people everywhere march with you. In company with our brave Allies and brothers-in-arms on other fronts, you will bring about the destruction of the German war machine, the elimination of Nazi tyranny over the oppressed peoples of Europe, and security for ourselves in a free world. Your task will not be an easy one. Your enemy is well trained, well equipped and battle-hardened. He will fight savagely…I have full confidence in your courage, devotion to duty and skill in battle. ***We will accept nothing less than full Victory!*** *Good luck! And let us beseech the blessing of Almighty God upon this great and noble undertaking.*

Dwight D. Eisenhower, June 6, 1994

LIST OF REFERENCES

[1] Chen, L. C., T.-A. Teo, et al. (2005). Building reconstruction from LIDAR data and aerial imagery. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. 4: 2846-2849.

[2] Chen, Q.-S., M. Defrise, et al. (1994). "Symmetric Phase-Only Matched Filtering of Fourier-Mellin Transforms for Image Registration and Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence 16(12): 1156-1168.

[3] Cho, W., Y. S. Jwa, et al. (2004). Pseudo-grid Based Building Extraction Using Airborne LiDAR Data. 20th ISPRS Congress: Geo-Imagery Bridging Continents. Istanbul, Turkey: 1-4.

[4] Clode, S. P., P. J. Kootsookos, et al. (2004). Accurate Building Outlines from ALS Data. 12th Australasian Remote Sensing and Photogrammetry Conference Esplanade Hotel, Fremantle, Perth, Western Australia

[5] Filin, S. (2005). Elimination of Systematic Errors From Airborne Laser Scanning Data. Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International. 1: 4.

[6] Garland, M. and P. Heckbert (1995). Fast polygonal approximation of terrains and height fields. Technical Report, Department of Computer Science, Carnegie Mellon University.

[7] Gevers, T. and A. W. M. Smeulders (2000). "PicToSeek: combining color and shape invariant features for image retrieval." IEEE Transactions on Image Processing 9(1): 102-119.

[8] Gonzalez, R. C., R. E. Woods, et al. (2003). Digital Image Processing Using MATLAB. New Jersey, Prentice Hall.

[9] Heipke, C., H. Mayer, et al. (1997). "Evaluation of Autoamtic Road Extraction." International Archives of Photogrammetry and Remote Sensing 32: 47-56.

[10] Hu, J., S. You, et al. (2004). Building Modeling From LiDAR and Aerial Imagery. Proceedings of ASPRS. Denver, USA.

[11] Katartzis, A. and H. Sahli (2008). "A Stochastic Framework for the Identification of Building Rooftops Using a Single Remote Sensing Image." IEEE Transactions on Geoscience and Remote Sensing 46(1): 259 - 271

[12] Lammons, G. (2006). After The Storm. Military Geospatial Technology. 4.

[13] Lefevre, S., J. Weber, et al. (2007). Automatic Building Extraction in VHR Images Using Advaned Morphological Operators. Urban Remote Sensing Joint Event. Univ. Louis Pasteur, Illkirch: 1-5.

[14] Liu, W. and V. Prinet (2005). Building detection from high-resolution satellite image using probability model. Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International.

[15] Mahalanobis, A. (2007). Multidimensional Algorithms for Target Detection in LiDAR Imgary. University of Central Florida, Electrical and Computer Engineering Seminar Series. Orlando, Florida.

[16] Matikainen, L., J. Hyyppa, et al. (2003). Automatic detection of buildings from laser scanner data for map updating. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Services. 34: 218-224.

[17] Mayer, H. (1999). "Automatic Object Extraction from Aerial Imagery - A Survey Focusing on Buildings." Computer Vision and Image Understanding 74(2): 138-149.

[18] Meyer, Fernand, "Topographic distance and watershed lines," Signal Processing , Vol. 38, July 1994, pp. 113-125.

[19] Morgan, M. and A. Habib (2002). "Interpolation of LiDAR Data and Automatic Building Extraction". CSM - ASPRS 2003 Annual Conference Proceedings.

[20] Muller, S. and D. Zaum (2005). Robust Building Detection in Aerial Images. IAPRS. Vienna, Austria. 36.

[21] NAHB (2009). "National Association of Home Builders - Rapid Changes Forseen for Home of the Future." Retrieved May 12, 2009, from http://www.nahb.org/news_details.aspx?newsID=2886.

[22] NASA (2009). "Measuring Vegetation (NDVI & EVI)." Retrieved May 11, 2009, from http://earthobservatory.nasa.gov/Features/MeasuringVegetation/measuring_vegetation_2.php.

[23] Ortner, M., X. Descombes, et al. (2008). " A Marked Point Process of Rectangles and Segments for Automatic Analysis of Digital Elevation Models." IEEE Transactions on Pattern Analysis and Machine Intelligence 30(1): 105-119.

[24] Otsu, N. (1979). "A Threshold Selection Method From Gray-Level Histograms." IEEE Transactions on Systems, Man and Cybernetics 9(1): 62-66.

[25] Persson, M., M. Sandvall, et al. (2005). Automatic Building Detection from Aerial Images for Mobile Robot Mapping. Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on.

[26] QTM (2009). "Applied Imagery Quick Terrain Modeler Download Website." Retrieved February 16, 2009, from http://appliedimagery.com/download.php

[27] Reddy, B. S. and B. N. Chatterji (1996). "An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration." IEEE Transactions on Image Processing 5(8): 1266-1271.

[28] Rottensteiner, F., J. Trinder, et al. (2003). Building Detection Using LiDAR Data and Multispectral Images. Digital Image Computing: Techniques and Applications. Sydney, Australia. 2: 673-682.

[29] Rottensteiner, F., J. Trinder, et al. (2005). "Using the Dempster-Shafer method for fusion of LIDAR data and multi-spectral images for building detection." Information Fusion 6: 283-300.

[30] Rottensteiner, F., J. Trinder, et al. (2007). "Building detection by fusion of airborne laser scanner data and multi-spectral images: Performance evaluation and sensitivity analysis." ISPRS Journal of Photogrammetry & Remote Sensing 62: 135-149.

[31] Shorter, N. and T. Kasparis (2006). "Fuzzy SART Clustering for 3D Reconstruction from Irregular LIDAR Data." WSEAS Transactions on Signal Processing 2(8): 1122-1129.

[32] Shorter, N. and T. Kasparis (2008). Triangulated, Connected Sets for Building Detection from Irregularly Spaced LiDAR. The 3rd International Symposium on Communications, Control, and Signal Processing. St. Julians, Malta: 560-565.

[33] Shorter, N. and T. Kasparis (2008). Autonomous Registration of LiDAR Data to Single Aerial Image. IEEE International Geoscience & Remote Sensing Symposium. Boston, Massachusetts. 5: 216-219.

[34] Sirmacek, B. and C. Unsalan (2008). Building Detection From Aerial Images Using Invariant Color Features and Shadow Information. 23rd International Symposium on Computer and Information Sciences. Istanbul: 105.

[35] Soille, P. (1999). Morphological Image Analysis Principles and Applications. Berlin, Germany, Springer Verlag.

[36] Sun, B. Y., D.-S. Huang, et al. (2005). "LiDAR Signal Denoising Using Least Square Support Vector Machine." IEEE Signal Processing Letters 12(2): 101-104.

[37] Toth, Charles K (2009). Topographic Laser Ranging and Scanning Principles and Processing. Boca Raton, Florida (USA), CRC Press.

[38] Unsalan, C. and K. L. Boyer (2004). A system to detect houses and residential street networks in multispectral satellite images. Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. 3: 49-52.

[39] Verma, V., R. Kumar, et al. (2006). 3D Building Detection and Modeling from Aerial LIDAR Data. Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on 2.

[40] Vosselman, G. (2000). Slope Based Filtering of Laser Altimetry Data. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Amsterdam, Netherlands. 33: 935-942.

[41] Vosselman, G., B. G. H. Gorte, et al. (2004). Change detection for updating medium scale maps using laser altimetry. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Istanbul, Turkey.

[42] Vu, T. T., M. Matsoka, et al. (2004). LiDAR-based change detection of buildings in dense urban areas. Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International. 5: 3413-3416.

[43] Xie, M., K. Fu, et al. (2006). Building Recognition and Reconstruction from Aerial Imagery and LIDAR Data. CIE '06. International Conference on Radar: 1-4.

[44] Yin, S. and W. Wang (2006). LiDAR Signal Denoising Based on Wavelet Domain Spatial Filtering. CIE '06 Interntional Conference on Radar. Shanghai: 1-3.

[45] Zhang, K., S.-C. Chen, et al. (2003). "A progressive morphological filter for removing nonground measurements from airborne LIDAR data." Geoscience and Remote Sensing, IEEE Transactions on 41(4): 872-882.

[46] Zhou, Q. - Y. and Neumann, U. (2008). Fast and extensible building modeling from airborne LiDAR data. GIS 2008: 7

[47] Zitova, B. and J. Flusser (2003). "Image Registration Methods: A Survey." Image and Vision Computing 21: 977-1000.