BUILDING EXTRACTION FROM LIDAR USING EDGE DETECTION

by

Justin Miller
A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Master of Science
Geoinformatics and Geospatial Intelligence

Committee:

_____	Dr. Arie Croitoru, Thesis Director
_____	Dr. Anthony Stefanidis, Committee Member
_____	Dr. Peggy Agouris, Committee Member
_____	Dr. Anthony Stefanidis, Department Chairperson
_____	Dr. Donna M. Fox, Associate Dean, Office of Student Affairs & Special Programs, College of Science
_____	Dr. Peggy Agouris Dean, College of Science

Date: _____ Summer Semester 2015
George Mason University
Fairfax, VA

Building Extraction from LiDAR Using Edge Detection

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

by

Justin Miller
Bachelor of Science
Shenandoah University, 2008

Director: Arie Croitoru, Professor
Department of Geoinformatics and Geospatial Intelligence

Summer Semester 2015
George Mason University
Fairfax, VA

DEDICATION

This is dedicated to my wonderful and loving wife Jenn, son Camden, and my parents Jan and Bill. Without their patience and support throughout my life and school work, I would not have been so successful in my life and career.

ACKNOWLEDGEMENTS

I would like to thank my committee chairman, Dr. Arie Croitoru for his patience and sharing his wealth of knowledge in this complex field of study. I would also like thank my committee members Dr. Anthony Stefanidis and Dr. Peggy Agouris for their wisdom, advice, and feedback.

Also I would like to express my appreciation to the Army Geospatial Center for allowing me to use this dataset.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

Army Geospatial Center	AGC
Bare Earth Model	BEM
Digital Elevation Model	DEM
Digital Surface Model	DSM
Digital Terrain Model	DTM
Foliage Penetration	FOPEN
Geographic Information System	GIS
Global Positioning System	GPS
Internal Measurement Unit	IMU
Light Detection and Ranging	LiDAR
Laplacian of Gaussian	LoG
Synthetic Aperture Radar	SAR
Three-dimensional	3D
Two-dimensional	2D

ABSTRACT

BUILDING EXTRACTION FROM LIDAR USING EDGE DETECTION

Justin Miller, M.S.

George Mason University, 2015

Thesis Director: Dr. Arie Croitoru

Light Detection and Ranging (LiDAR) has become a versatile data source for many applications including building detection. Previously, manual photogrammetric methods were needed to accurately digitize building footprints, often resulting in ineffective and costly data collection process. Automated building extraction from imagery has been studied extensively, in particular using edge detection and image segmentation methods. However, the utilization of such methods, and in particular edge detection, for building extraction from LiDAR has not been fully explored. Consequently, this research explores the use of edge detection-based feature extraction as a possible framework for building detection and delineation in LiDAR data. In particular, building on existing edge detection and image segmentation operators, the proposed framework utilizes a rotating kernel for detecting the edges of buildings as well as the watershed segmentation operator for segmenting and identifying each building. Once identified, each building is then delineated using a combination of Hough transform and topological polygon

construction, resulting in the building's footprint. The building extraction process was tested on three different datasets containing buildings of various shapes, and the extraction results were compared to manually extracted footprints in order to evaluate the accuracy and precision of the proposed framework. The analysis results show that the proposed framework was 90% accurate with 95% of the extracted results area overlapped the manually extracted footprints.

CHAPTER 1: INTRODUCTION

Photogrammetry has been around for decades providing accurate measurements, terrain extraction, and feature extraction. The imagery sources available to photogrammetrists at their discipline's infancy were aerial imagery, satellite imagery, and synthetic aperture radar (SAR). Until recently, the only way to produce digital elevation models remotely was to go through a tedious process of triangulating the images and manually extracting features. As technology for photogrammetry improved, the triangulation process became faster and easier. Newer technologies, including LiDAR, were developed and started an evolution for terrain and feature extraction. LiDAR was developed over 40 years ago for mapping particles in the atmosphere (National Oceanic and Atmospheric Administration (NOAA) Costal Services Center, 2012). It wasn't until Global Position Systems (GPS) became available to moving sensors in the 1980s and the improvement of internal measurement units (IMU) in the 1990s when LiDAR started to be used for measuring the Earth's surface (National Oceanic and Atmospheric Administration (NOAA) Costal Services Center, 2012). Today LiDAR has become more accurate and developed providing capabilities far beyond photogrammetry.

1.1 The Need for Building Footprints

In recent years, there has been a significant increase in the need for accurate three-dimensional (3D) data of urban areas and the continuous updates of the Earth's

changing landscape. These needs have led to research efforts that aim to develop automatic and semiautomatic tools for the creation of such data. These automated and semiautomatic methods also need to produce accurate geospatial information while keeping labor and costs low. LiDAR is becoming the lead data source for producing accurate geospatial information.

For this reason, LiDAR could have an important role in the automation of building detection and the creation of 3D topographical databases (Lee et al., 2008). For the reasons above, building extraction using the automatic techniques on LiDAR data has a great potential as a research topic to meet the demands for applications that use building footprints and modeling. Applications that currently use building footprints and modeling consist of simulation of disaster scenarios, cartography, urban planning and management, wireless network planning, line-of-sight analysis, virtual tours, and many others (San & Turker, 2010). Urban landscape modeling is necessary for planning drainage systems, street improvement projects, disaster management, and other tasks (Yu et al., 2010).

The United States tears down about 1.75 billion square feet of buildings, renovates approximately 5 billion square feet, and builds approximately 5 billion square feet of new structures each year. It is estimated by 2035 that 75% of the structures will be either new or renovated. The newly built environments need planning, electricity, water, improved infrastructure, and other human-necessities and services. The amount of change predicted to occur over the next few decades shows that there is a need for building modeling methods to be faster, less labor intensive, and cheaper. (Architecture 2030 n.d.)

1.2 Statement of the Problem

The production process of the traditional photogrammetric method is efficient and cost effective when creating smooth, generalized terrain models but when the terrain models are more detailed, the efficiency decreases and the cost increases. This is due to the need for higher resolution imagery and higher accuracy of triangulated stereo pairs (Zhou et al., 2004). Triangulation is a complex and tedious process that aligns two images of the same area but collected at different angles to create a three-dimensional (3D) view of the area. A remote sensing method similar to LiDAR, SAR, has been researched as a data source to provide automated building extraction (Tupin & Roux, 2003). SAR is similar to LiDAR in that it is an active sensor but SAR data is often very difficult to interpret because it contains lay-overs, shadows, and noise, and in many cases buildings that are scarcely recognizable (Tupin & Roux, 2003). LiDAR contains minimal noise compared to SAR and is much more interpretable by humans.

Developed in the 1970s, LiDAR is not new or immature, but has not fully matured either (Miliaresis & Kokkas, 2007). There are many applications and algorithms still being developed and explored. Some current applications of LiDAR are urban planning, telecommunication, environmental monitoring, 3D modeling, and military operations. One feature that is used by these applications and many more is the location of buildings. Determining where the next housing development or cell phone tower should be placed, how the expansion of cities are affecting the environment, and navigating a military convoy are all situations where it is a necessity to know the locations of existing buildings. Previous photogrammetric techniques for building extraction used two-dimensional (2D) and stereo imagery to manually digitize buildings,

and like the traditional photogrammetric techniques used for terrain modeling, the techniques become inefficient and costs increase as the more detailed the products need to be.

Since the Earth's population continues to grow and the number of building footprints expand, databases need constant updating which can be time consuming and expensive. With technology advancing and becoming faster and more economical, and with the improved accuracy and resolution, LiDAR can be used to extract building footprints more efficiently and cost-effectively than other methods. Since raster processing is more mature than processing point clouds, DSMs can be used and processed like panchromatic images by using the same techniques and algorithms. This includes edge operators, raster math, and mathematical morphology. Research on using these algorithms and techniques on DSMs is limited which lead to the research in this thesis; the study of an approach to accurately and efficiently automate the creation of building footprints using methods already established and proven for imagery on raster data derived from LiDAR point clouds.

1.3 Organization of Thesis

The literature review chapter of this thesis discusses current and past research in edge detection of remotely sensed data. The approach studied and implemented in this thesis is discussed in Chapters 3 and 4. Data from Stafford County's government website of the study area and two sets of digitized data were used for comparison to the automated footprints and is discussed in Chapter 5. Final remarks of the comparison results and future research possibilities are discussed in the conclusion chapter.

CHAPTER 2: LITERATURE REVIEW

There has been a large amount of research on edge detectors, methods to process raster imagery, and feature extraction from imagery. Raster processing is well-studied, defined, and proven to produce various useful products. A better understanding of raster processing, in particular edge detection techniques, was established reviewing articles on the subject. LiDAR processing is not as established as raster processing and articles about established and proven LiDAR processes are limited but there is still active research about these processes. Processing LiDAR for building detection has become a more popular topic in the past decade. The topics of edge and building detection from raster imagery and LiDAR are reviewed and discussed in this chapter.

2.1 LiDAR Basics

LiDAR is becoming a more widely used remote sensing technique for various applications. LiDAR is an active sensor which provides its own energy source to send to the Earth's surface and then records the signal when reflected back to the sensor. Passive sensors (i.e. colored image products) use the Sun's energy that is reflected off the Earth's surface and then the sensor records the signal. LiDAR is an active sensor and unique compared to the conventional panchromatic, colored, or multispectral images because it uses its own energy, the laser, and records that reflection off the Earth's surface. The main product created from LiDAR is elevation models where traditional

photogrammetric tools and stereo imagery were used to create these elevation models (Zhou et al., 2004)

A LiDAR system for aerial collection is comprised of a GPS, IMU, and the LiDAR sensor. The sensor sends a laser pulse thousands of times a second while recording each returned pulse. The laser emits energy at a particular wavelength, most commonly in the near infrared electromagnetic spectrum range, and it depends on the material composition of an object on how well the material reflects the energy (National Oceanic and Atmospheric Administration (NOAA) Costal Services Center, 2012). The LiDAR system records the location from which the returned pulse came, the elevation at that location, and the intensity of the returned pulse as points called a point cloud. To record the elevation and location from which the pulse came, the LiDAR system uses the aircraft's position from the GPS information, the scanning angle of the sensor, rotations of the sensor's platform from the IMU, and the time the pulse took to return. The GPS information is not as accurate as needed to properly calculate the position of the returned pulse. Positional errors recorded by a GPS base station near the collection site is used to improve the GPS information in the LiDAR data.

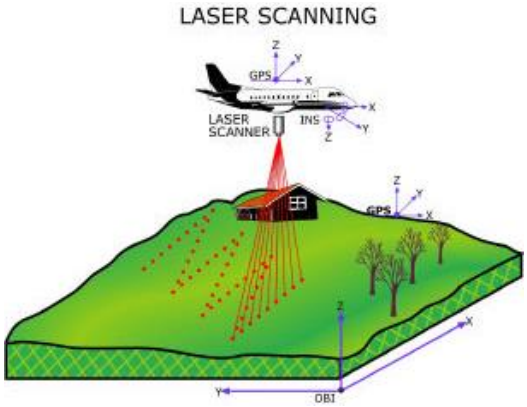


Figure 1: Laser Scanning (Renslow et al., 2012)

A single pulse can have multiple returns depending on the object or material on the ground on which it is collecting, and can help describe the characteristics of the ground cover. Consider the tree in Figure 2, for instance. The top most branch(es) will return the first signal and a second signal will be returned by the next branch, and so on until laser cannot penetrate any further or it reaches the ground which will be the last return. The capability of being able to penetrate objects and record multiple returns, like the tree, is referred to as foliage penetration (FOPEN).

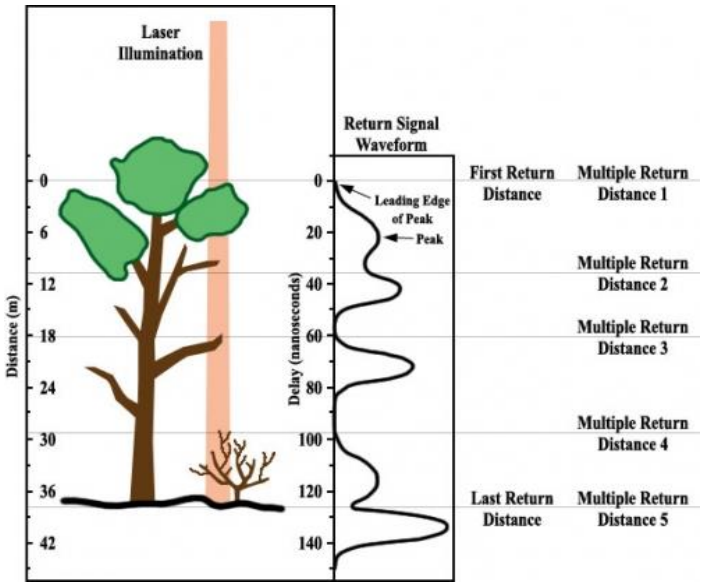


Figure 2: LiDAR Multiple Returns (Schuckman & Renslow, 2014)

Pre-processing steps that are needed to reduce noise in the point cloud include flight line matching, which is aligning strips of LiDAR from each pass of the sensor, and removing spikes or holes. Spikes are caused by flying objects (e.g. birds, clouds) and random atmosphere refraction. Holes are caused from materials that absorb all the energy of the laser pulse, such as water and LiDAR shadows. LiDAR shadows are caused by tall objects or steep terrain where the laser cannot penetrate causing a large area with no return data similar for a building casting a shadow from the Sun.

A digital surface model (DSM) and bare-earth model (BEM) can be created by converting the point cloud into a gridded or raster format where the pixel value is the elevation (z value) or the intensity value at that location. The resolution of the DSM is dependent on the density of the point cloud and the diameter of the laser beam. Usually each return is used to create its own DSM (i.e. first returns creates a first return DSM).

The value of a pixel is determined by an algorithm (e.g. Kriging, inverse weight, etc.) that uses the points inside, or closest to, the pixel's region. A BEM can also be created using algorithms to eliminate objects above the Earth's surface such as buildings, trees, cars, etc. to show only the ground's characteristics.

DSMs are usually created because of the immense amounts of storage and computation power needed to store, analyze, and visualize the point clouds. Raster processing is more developed and is faster than processing point clouds making DSMs more versatile for the basic LiDAR user. These images can be processed to detect the edges of objects that change in elevation such as the elevation difference between the edge of a building and the ground.

2.2 Edge Detection

Edge detection is the process of detecting and locating discontinuities and changes in pixel values within an image (Maini & Aggarwal, 2009). "The job of extracting edges from digital images is very important for any computer vision system whose goal is to perform object recognition, segmentation, or a variety of other image processing operations" (McLean & Jernigan, 1988). There has been extensive research on the different types of edge detection techniques for panchromatic and spectral images like the Sobel, Prewitt, Cranny, or Laplacian techniques (Heath et al., 1998) (Maini & Aggarwal, 2009).

Since DSMs can be processed just like an image, edge detectors can be a useful tool when finding boundaries of features. There are many algorithms and operators for edge detection, which have been around for decades, but they can be grouped into two

categories: Gradient based and Laplacian based (Maini & Aggarwal, 2009). Gradient based uses the first derivative of the image to find the maximum and minimum and Laplacian based uses the second derivative to find the zero crossing. Most of these edge detecting methods find the local derivative in a predefined kernel size which are much smaller than the image. The kernel is moved over the entire image, pixel by pixel, and performs the calculations, according to the edge detection method, on the pixels covered by the kernel (Maini & Aggarwal, 2009). The size of the kernel determines how sensitive it is to potential edges with smaller kernels being more sensitive. A sensitive kernel is not always the best choice and the size can depend on how noisy the image is.

Well known Gradient based edge detectors include Sobel, Prewitt, and Roberts and each have their own unique way of detecting edges (Juneja & Sandhu, 2009). The Sobel method consists of two 3 X 3 kernels detecting edges in the horizontal and vertical direction and can also be modified to detect edges in the positive and negative diagonal directions (Figure 3), in which each is implemented separately and then combined to obtain the optimal edges. This method is considered the historical standard and is still seen in published papers today (Heath et al., 1998). Prewitt is very similar to Sobel kernel in that it has the same kernel size, design, and zero center weight except it uses a negative and positive one in place of the twos (Figure 3). Roberts is a two by two kernel designed to find edges in the diagonal directions with a weight of one in adjacent corners and zero in the other two corners.

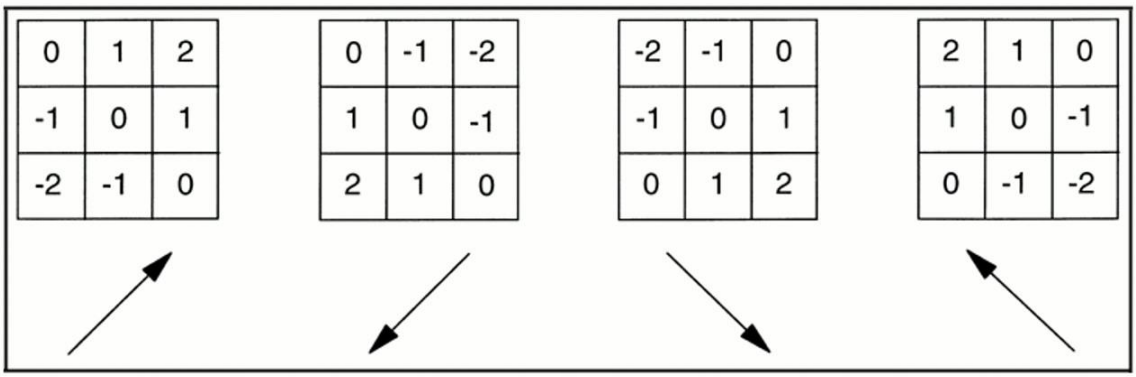


Figure 3: Sobel Edge Detection Kernel

Laplacian based methods consist of Laplacian and Laplacian of Gaussian (LoG). As mentioned previously, Laplacian methods use the second derivative and the common kernel of the Laplacian edged detector is shown in Figure 4. In order to use the second order derivative on a finite set of samples, an approximation of the second order derivative is used which is represented as the Laplacian edge detector kernel (Agouris et al., 1989). The downfall of this approximation is that it is very sensitive to noise. This led to the development of the LoG kernel. This edge detector uses a Gaussian smoothing to reduce the high frequency noise and then performs the Laplacian detection (Juneja & Sandhu, 2009). This can also be done in one step using one kernel which is the approximate of the LoG function.

(a)

-1	-1	-1
-1	8	-1
-1	-1	-1

(b)

0	-1	0
-1	4	-1
0	-1	0

Figure 4: Laplace Operators (a) 4 directions (b) 8 directions

2.3 Pre-processing Prior to Detecting Buildings

Before detecting buildings from a DSM, some filtering is required in order to help the buildings stand out from other objects. These filtering techniques can be placed into two groups: ground filtering and remaining non-building removal (Meng et al., 2006). Ground filtering consists of identifying the ground and then separating the ground from objects above the ground. After the ground filtering, objects like trees, buildings, cars, etc., remain and need to be filtered so only the buildings remain. There are multiple algorithms and techniques for each filtering technique, as described below.

2.3.1 Ground Filtering

Most building detection techniques using LiDAR start with separating the ground from the features above ground. One common method is to take the DSM and subtract the Digital Terrain Model (DTM, also known as a BEM) as seen in Demir, et al (2009). The problem using this method arises when trying to generate a BEM. There are several complex algorithms identified by Sithole and Vosselman (2004) to produce a BEM and each approach has advantages and limitations depending on the terrain, landscape, outlier points, and computational speed. If the algorithm does not create the BEM correctly, manual editing is needed to reduce the risk of identifying buildings as ground.

Each algorithm tested by Sithole and Vosselman (2004) fall into four filter concepts: slope-based, block-minimum, surface-based, and clustering/segmentation. Each of these concepts compares the immediate surrounding points and makes an assumption about the structure of the bare-earth points. The sloped-based method uses the slope between two points and if the slope exceeds a threshold, the highest slope is considered a non-ground point. The block-minimum method uses a horizontal plane based off of the

lowest points and then uses a buffer above the plane to classify the points within the buffer as ground. The surface-based method does the same except it uses a parametric surface. Clustering/segmentation uses the concept that a cluster of points must not be ground points if that cluster is above its neighborhood. As mentioned earlier, each ground filtering algorithm has their advantages and limitations and should be based on the terrain and landscape.

Morgan and Tempeli (2000) who used a morphological filter on the DSM raster showed that the BEM generation process can be skipped. The morphological filter works on the DSM by using a moving window that changes in size. The idea is to compare the pixel of interest to others in the window to weight that pixel of how likely it is to be a ground or non-ground pixel according to how high the pixel is relative to its neighbors and how big the window size. Elaksher and Bethel (2002) uses a slightly different approach by using a minimum filter on the DSM, in which points above five meters is considered a building candidate.

2.3.2 Separating Building and Non-Building Objects

After separating points/pixels above the ground, trees and other non-building points have to be removed. Statistics, geometric constraints, and other remote sensing sources can be used to help distinguish the differences between objects in the DSM.

One common statistical technique is to take the last return and subtract it from the first return (Meng et al., 2006). The idea for the LiDAR data's first return minus the last return is that the laser pulse can penetrate tree canopies so there will be a height difference but man-made objects are not penetrable which will give a zero height

difference. However, there can be height differences in rough terrain and edges of buildings that need to be considered. One solution is to use a high gradient response of the last return and use a threshold to remove the building edges (Alharthy & Bethel, 2002). After the building edges are removed, all that should remain is vegetation which is used to mask out the DSM.

In "Automatic Construction of Building Footprints," Zhang, et al. (2006) present a four step process. First, patches of similar height below a minimum area are removed. Second, patches that are part of buildings (i.e. chimneys, tanks, etc.) are recovered by their relationship to large patches surrounding the removed patches. Third, isolated boundary points (points that were previously identified as ground, building boundary, or inside building boundary) that do not have an inside point neighbor are removed. Finally, the remaining patches are merged together according to their connectivity to other patches to form complete building patches and then another minimum area removal is used.

Other approaches go further into the geometric shape of vegetation to identify and remove them. Since buildings are used for storage or human accommodations, they should be high enough and large enough to identify and remove (Haithcoat et al., 2001). Therefore, any point below a certain height can be removed, which will eliminate objects like bushes and cars. Haithcoat, et al. (2001) suggested taking the gradient of the DSM and used a magnitude threshold to remove the trees. The idea behind that approach is that tree edges will have a higher magnitude from the Gaussian kernel and the shape of the boundaries will be irregular while the building boundaries will be rectangular.

Some authors suggested using local variance or statistics to identify vegetation. The elevation in tree canopies will vary locally and therefore the variance will be high while building roofs are usually flat or have a gradual slope (Zabuawala et al., 2009). Another technique is to take the difference between the elevation of a pixel and the average elevation of its 3 X 3 neighborhood (Ekhtari et al., 2009). This technique describes the roughness of the pixels and a roughness value above a determined threshold is considered non-building.

Zhang, et al. (2006) suggested that statistics and geometric constraints may not be the approach of choice, especially for areas with dense trees. With the combination of LiDAR data and imagery, especially a three or more band image, can help with the classification of vegetation. Awrangejeb, et al. (2010) used multispectral imagery along with a height threshold to identify vegetation. Normalized Difference Vegetation Index and texture information from the multispectral image was used to identify pixels in the DSM as vegetation. A classification method one could use in areas of dense trees it to measure how green a pixel is in a three band RGB image (Yu et al., 2011).

2.4 Building Detection with LiDAR Grids

After the pre-processing is complete, only buildings should remain, allowing us to address the problem of accurately describing, measuring, and defining the boundaries of the buildings. Before LiDAR, feature and building detection were done using edge detectors. However, research on using these same techniques with LiDAR data is limited.

Some research exists on derivatives of these techniques like the altered Canny algorithm found in "A Method of Edge Detection Based on Improved Canny Algorithm

for the LiDAR Depth-image" by Xu, et al. (2006), where the deficiencies of the original Canny algorithm were studied and the new method designed to eliminate those deficiencies. The altered Canny algorithm uses a median filter with an adaptable filter size based on noise distribution for smoothing instead of Gaussian smoothing, 3 X 3 neighborhood for gradient calculations, and an adaptive threshold based statistical measures producing better results. Some research has been done using the classic edge detectors like Sobel and the original Canny edge detector on an image derived from the LiDAR points intensity values to extract runway edges (Mareboyana & Chi, 2006). Weng, et al. (2010) used the original Canny edge detector and further refined the results using textures identified in aerial imagery. Wang, et al. (2010) were able to produce building footprints using a Canny edge detector with an adaptive threshold for detecting building edges and then used a Quadric Error Metric based simplification. The Quadric Error Metric algorithm was used to rapidly build footprints creating 90° angles on adjacent edges and straight lines.

Alternatively, seed growing methods have been commonly used for determining the extents of a building when given a set of parameters and the relationship of surrounding pixels determines which pixels are buildings. Miliaresis and Kokkas (2007) used slope between 40° and 90° as the parameter for potential building seeds. Then, a region growing segmentation algorithm was used to create building cells by connecting other pixels to the seed that have the same characteristics.

2.5 Building Detection with LiDAR Points

Working with LiDAR point clouds is more difficult than grids due to the large amount of data and complex geometry. Algorithms and processing methods need to be adapted to the environment of irregular spaced data in the point cloud. Some benefits of working with points are increased accuracy and preventing the loss of data. When converting from points to grids, interpolation is necessary to convert the irregular point spacing. Interpolating the points can have adverse effects such as smoothing edges or incorrectly interpolating values.

Verma, et al. (2006) uses the identified roof points to fit a plane to local point patches, group the patches based on similarities, and then defines approximate boundaries of the groups. Sampath and Shan (2010) define local neighbors of each point by using the Voronoi neighborhood method. These neighborhoods are then categorized as on a planar surface or non-planar along with the characteristics of the plane. Points with similar characteristics are grouped together and assumed to be on the same plane and therefore the building roof can be segmented.

2.6 Building Detection Using a Combination of LiDAR and Other Remote Sensors

Other researchers used a combination of imagery, DSMs, and DTMs for feature extraction. An aerial image can be used to detect and extract the edges of the buildings and the LiDAR can remove vegetation and provide the buildings' geometric shape for modeling (Zhou et al., 2004). Ekhtari, et al. (2009) used WorldView imagery to detect the edges of the buildings and then uses the LiDAR to mask out trees and other vegetation to help building edge reconstruction from gaps left by trees overhanging roofs.

Using WorldView and aerial imagery provides spectral information in addition to LiDAR's physical measurements (i.e. heights).

The spectral information can help define building edges by detecting spectral differences between the building and the surrounding features. Imagery limitations are the inability to distinguish the differences between objects that spectrally look similar, such as a black roof and an asphalt road, and shadows from the Sun can cause false edges. It is difficult to find and expensive to obtain both LiDAR data and imagery. Using the LiDAR data and imagery requires an additional process of geospatially registering and aligning them.

2.7 Uses for Building Footprints and 3D Buildings

Building footprints are essential geographic information system (GIS) data and are used for creating and updating building inventories for planning and analysis purposes. Planning applications include urban planning, telecommunication network planning and vehicle navigation (Miliaresis & Kokkas, 2007). Examples of using building footprints for analysis include estimating energy demands, quality of life, urban population, property taxes (Jensen, 2000), and damage assessments (Vu et al., 2005). In 1999, damage assessments of collapsed buildings were detected caused by the Izmit earthquake in Turkey using the before and after DEMs (Turker & Cetinkaya, 2005). The data allows analysis of complex urban characteristics and multidimensional scenarios for urban management (Zhao & Wang, 2014).

2.8 The Studied Approach

As discussed, there are multiple ways to use LiDAR to help automated feature extraction on buildings but there is a limited amount of research using LiDAR data for the entire process of building extraction. Motivated by this, the focus of the approach presented in this thesis is to use techniques proven for similar datasets (i.e. raster, imagery). Labor is meant to be kept at a minimum when constructing the proposed approach by using a process that is robust, automated, and without creating additional LiDAR products (e.g. BEM or point cloud classification). The more products that need to be created to build the footprints, typically the more editing and human interaction is needed defeating the intent and purpose of automation. The approach discussed assumes the user has a first and last return DSM or a processed point cloud data from which the DSMs can easily be extracted.

The first step of the proposed approach is the pre-process stage of creating a mask for removing natural features, especially large areas of trees and bushes using the first and last return DSM. Most previously researched algorithms and processes use a BEM. As discussed, BEMs are not always reliable and the accuracy of the BEM algorithm is dependent on the type of terrain and features. Further editing is most likely necessary after algorithms are used to create the BEMs, increasing the need for human interaction and labor.

Next, a 1 X 3 rotating kernel is used for detecting edges. This rotating kernel has not been proven for similar data types but the technique and idea of using kernels to do relative-local comparison has been. After the edge detection step, the vegetation mask is applied and each building is identified and separated by the morphological watershed

operator. This allows focus on one building at a time eliminating influences from other buildings. Each building's edge detection results are then converted to lines using Hough transform as possible lines for the building's footprint. Finally an algorithm, discussed in Chapter 4, is used on the lines to combine, align perpendicular and parallel, trim, and add lines where needed.

CHAPTER 3: PROPOSED EDGE DETECTION PROCESS

Chapters 3 and 4 discuss in detail the proposed approach of this thesis for detecting the edge of buildings and creating the footprints of the buildings. The high level approach consists of edge detection, creating the vegetation mask for vegetation removal, building separation, and footprint construction as described in Figure 5.

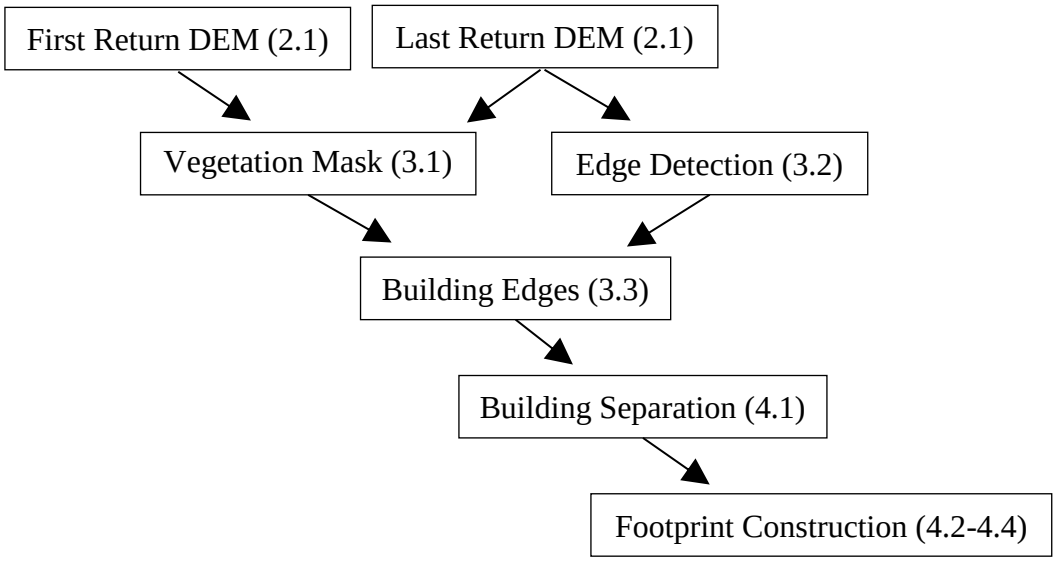


Figure 5: High Level Approach (numbers represent sections discussing the subject)

An important factor for successful edge detection is eliminating features that are not of interest. Non-building edges, especially from large areas of trees and bushes, make it difficult to discern building versus non-building after edge detection. The non-building

detection and edge detection processes are implemented in parallel and the results are combined to eliminate non-building features after edge detection. There are many different edge detection techniques that can be used but proposed in this thesis is the use of a 1 X 3 rotating kernel for detecting the edges. Vegetation removal and the 1 X 3 kernel edge detection are discussed in the chapter.

3.1 Vegetation Mask

The first step in the approach of this thesis is to separate and classify pixels as vegetation and non-vegetation. The method used in the approach is similar to Meng, et al. (2006) but without separating ground and non-ground points before classifying the vegetation. The process does not have to completely remove all the vegetation in this step because any smaller areas of vegetation possibly missed will be removed later in the "building separation" step in Section 4.1. The complete vegetation mask process is shown in Figure 6.

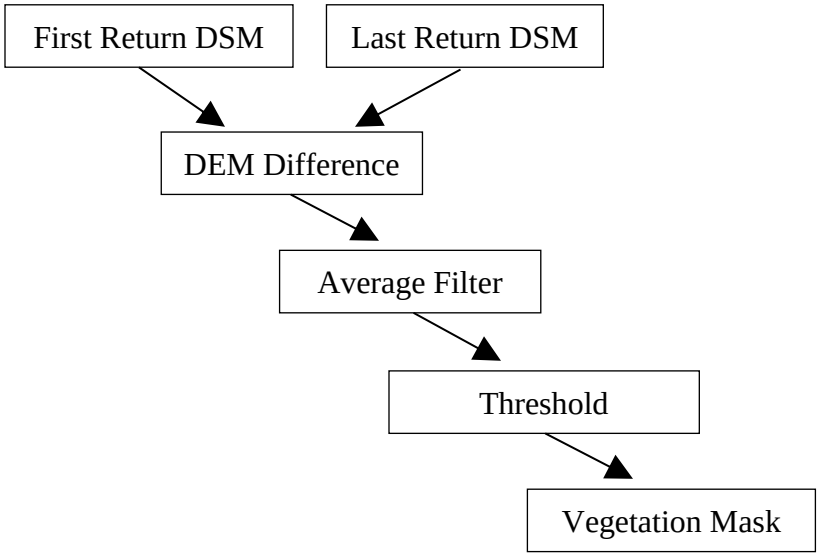


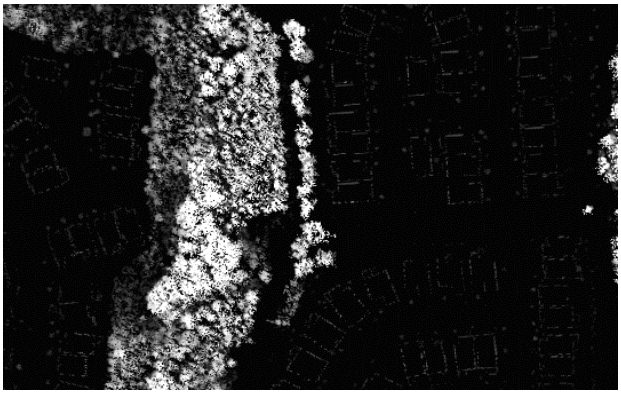
Figure 6: Vegetation Mask Process

The first return DSM is subtracted by the last return DSM which eliminates man-made features and other hard surfaces from the scene. Vegetation remains in the image and other features that are penetrable by the LiDAR. This method takes advantage of the foliage penetration (FOPEN) concept where LiDAR has the ability to penetrate vegetation as discussed in Section 2.1. The difference between the DSMs results shows that solid surfaces, the ground and buildings, will have a zero value and vegetation has a positive value. The positive value is the distance the LiDAR was able to penetrate the object.

An issue with the difference of the DSMs is that edges are left around buildings because a laser pulse can show the edge of the roof in the first return while showing the side of the building or the ground below the edge of the roof resulting in a positive value in the difference of the DSMs. A 7 X 7 average filter is used to eliminate most of the edges around the buildings (Figure 7). A 3 X 3 and 5 X 5 average filter is not big enough

to suppress the positive values around building edges in the difference of the DSMs and therefore a 7 X 7 filter is used. A mask image is produced by a threshold of greater than or equal to one in the averaged image and will be applied to edge detection results.

Most existing methods use the BEM to subtract from the DSM to separate non-ground pixels or points to form the ground which can produce inaccurate results (Demir et al., 2009) (Ekhtari et al., 2008). This process is typically done before classifying vegetation but is not part of the studied approach because the need of creating another product. In order to create the BEM, vegetation and man-made features are taken out of the elevation model and those areas are filled using an algorithm, such as Progressive Triangular Irregular Network Densification and Regularization method (Sithole & Vosselman, 2004), to estimate the elevation beneath those features. The results of these algorithms can vary and produce a false elevation, and therefore, skew the edge detection results. A different approach in this thesis is that the vegetation is not removed until after the edge detection is done. This is discussed in the next section.

(a) 

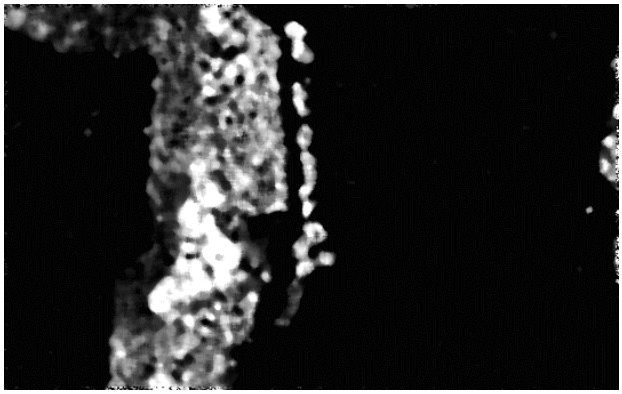
(b) 

Figure 7: Difference of DEMs (a) Image of first return subtracted by second return (b) 7 x 7 average filter of (a)

3.2 Edge Detection

Defining the complete and closed edge of the buildings is important for future steps in the approach in order to separate each building. A unique but simple edge detection process is used to define the edges of the buildings, utilizing concepts of edge detections discussed in Section 2.2, that have been used extensively with panchromatic imagery in comparing a pixel with its neighbors and deriving relative information. This process uses a rotating 1 X 3 logical kernel to find the local minimums and maximums as shown in Figure 8.

The 1 X 3 kernel is rotated in eight directions every 45° comparing the pixel of interest, pixel M, with the other two pixels, pixels 1 in Figure 8. The kernel starts in one of the eight directions testing if the pixel of interest is greater than the other pixels. If the pixel of interest is greater than the other pixels in the 1 X 3 kernel and is greater than two meters than the lowest value in the kernel, then one is added to a pixel, at the same location as the pixel of interest, in a new raster of the same size as the DEM. The kernel is then rotated to the other seven directions performing the same comparison and adding

one to the new raster each time the greater than criteria is met. Two meters was chosen to detect only edges that could be building edges and reduce detecting edges that are not buildings. For this thesis, it is assumed that bushes and cars are not taller than two meters, and that buildings are taller than two meters from the ground. The same is done with the comparison except testing if the pixel of interest is smaller than the other pixels in the kernel and is two meters below the highest value in the kernel. At the end of the process two new rasters, containing the local maximums and minimums, are created.

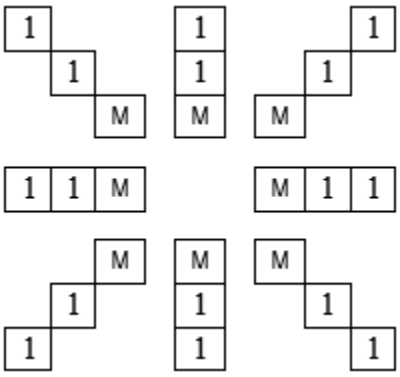


Figure 8: Rotating Minimum/Maximum Kernel

The key idea behind this process is that the local maximum result will outline the roof of the buildings as it steps down to the ground while the local minimum results will outline just off the edge as it steps up from the ground to the roof. The local minimum has the benefit of being able to define the separation between buildings close together. These rotating kernels are used on the last return DSM to minimize edges detected on vegetation and also help with the separation of buildings close together. The minimum and maximum results from the last return DSM is combined by subtracting the maximum

results from the minimum to eliminate noise and help distinguish the edges as shown in Figure 9. Due to noise in the DSM the maximum results could show a pixel was a maximum in one of the kernel's rotation where the rest of the rotations could have shown the pixel as a minimum. The raster created from combining the minimum and maximum results is the final step in detecting the edges of buildings. After combining the DSM results the vegetation mask from Section 3.1 is applied to eliminate vegetation edges. The vegetation removal does not remove all the vegetation but removes the larger areas so that the smaller areas can be filtered out later.

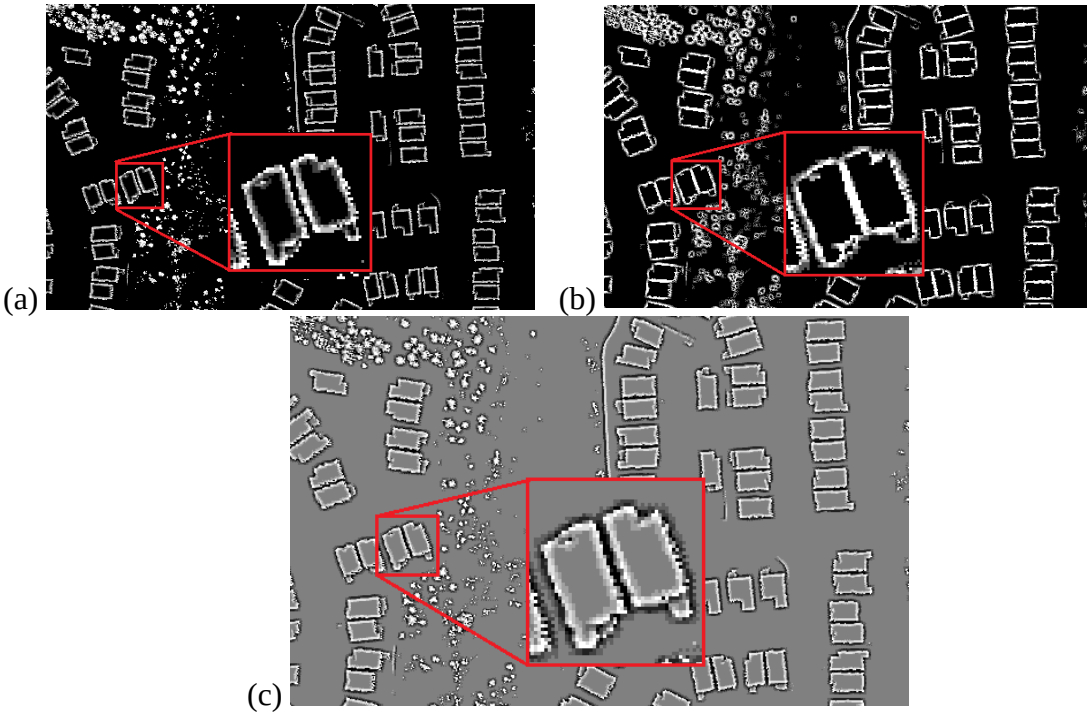


Figure 9: Minimum/Maximum 1 X 3 Rotating Kernel Results (a) Maximum results where pixel values are from 0 to 8 (b) Minimum Results where pixel values are from 0 to 8 (c) Maximum subtracted by Minimum results (edge detection results) where pixel values are from -8 to 8

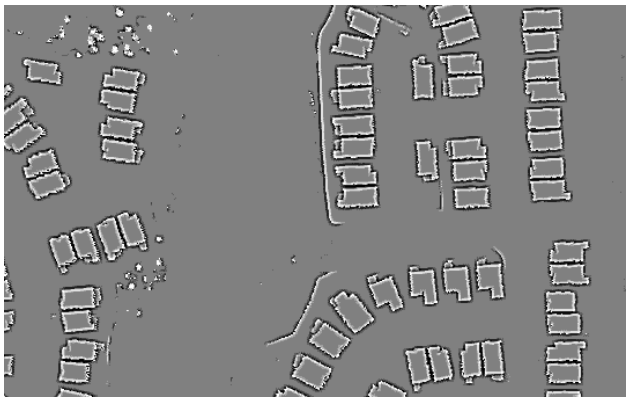


Figure 10: Vegetation Removal from the Edge Detection Results Using the Vegetation Mask (pixel values are from -8 to 8 and represent the results from the rotating kernel)

CHAPTER 4: PROPOSED FOOTPRINT CONSTRUCTION

Generally, the success of constructing footprints depends on two factors: how accurate the edge detector defines the edges and how well each building is identified and isolated from other buildings. Again, most of the processes used to obtain the possible lines of the footprint are methods and techniques that have been established and used for panchromatic imagery. The identification of individual buildings, defining footprint lines, and aligning the lines is discussed in detail later in this chapter and the steps are shown in Figure 11.

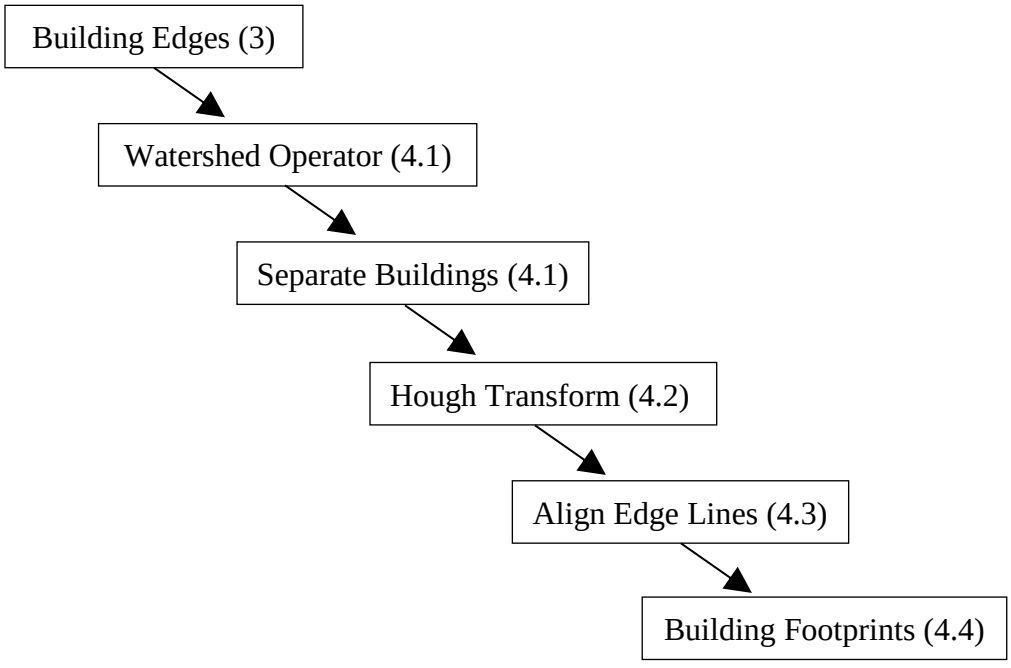


Figure 11: The Footprint Construction Process (numbers represent sections discussing the subject)

First the buildings are separated and isolated in their own raster in order to eliminate influences from other buildings when extracting the building's edges. Then the building edges are extracted from the edge detection results. The extracted edges are aligned and extended or trimmed according to adjacent edges. Finally the edges are connected forming a polygon and the vertices of the polygon were placed in a clockwise ordered and expanded to enlarge the polygon to be consistent with the size of the buildings.

4.1 Building Separation

Separating the buildings is a very important process in the proposed approach and has to be accurate, not only to identifying all the buildings but also for Hough transform (discussed in Section 4.2) to work correctly. It was found that without building segmentation, lines are created connecting multiple buildings and the formation of lines is influenced by any building in the scene. The watershed operator (Gonzalez & Woods, 2008) is used to segment the image to identify different regions and in this case the regions are the buildings. This technique has also been used for decades for image processing and segmentation.

The watershed operator finds a region by searching for low valued pixels surrounded by higher value pixels. If the edge detection did not completely enclose the building, the watershed operator will not define that building as a separate region. With the rotating kernel used for edge detection, a thick closed boundary is able to be

established as shown in Figure 12. Any value in the combined minimum and maximum raster is converted to a binary image where any value greater than or less than zero is converted to one.

Figure 12: Binary Minimum/Maximum Raster

The total area of a region is defined by the extents of thick boundary giving plenty of area to make sure the region contains the entire building. If boundaries of two regions are connected, the area of the regions stopped where boundaries connect as seen in Figure 13. In the results of the watershed operator each pixel within a region is given the same unique value to the other regions. These unique values were used one by one as a mask to extract the original detected edges from the rotation kernel and placed into its own raster as shown in Figure 13.

Before the building is saved as its own raster and considered a building, the mask has to contain 49 pixels, an area of a 7 X 7 pixel square, and contain 49 boundary pixels in the original edge detection results which will be the perimeter of a 10 X 10 pixel

square building. A pixel in the DSM is approximately one meter in length and width corresponding to a distance of seven meters for seven pixels and ten meters for ten pixels. The minimum area and perimeter threshold assumes that a building will have an area at least 49 square meters and a perimeter approximately 49 meters long. Most of the building edges from the edge detection results were at two or more pixels thick which means the building's perimeter could be less than 49 meters. Both the minimum area and perimeter thresholds are needed to ensure proper identification of buildings and non-building regions. Using just one of the thresholds could lead to non-buildings being identified as buildings.

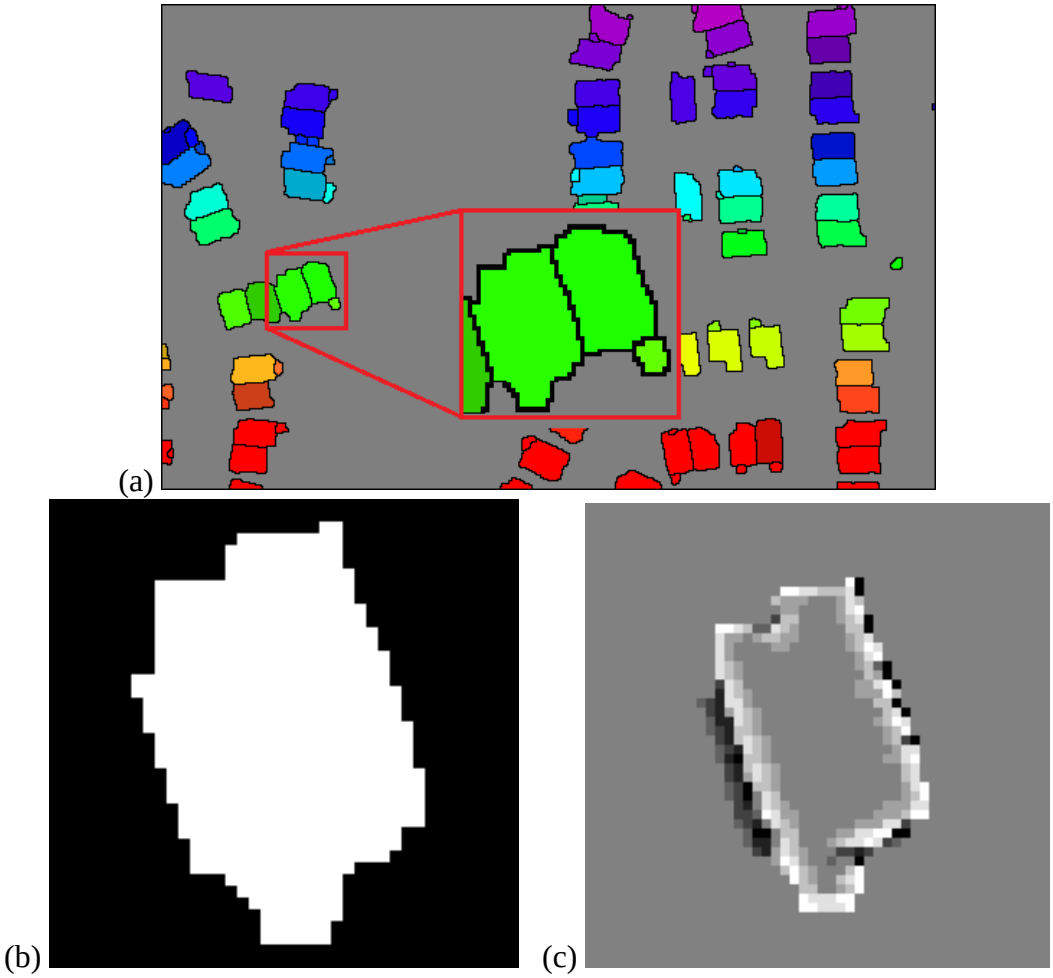


Figure 13: Watershed and Extraction Results (a) Watershed results colored by region number (b) Binary mask of left building in (a) inset (c) extracted edge detection results

4.2 Detecting Lines by Hough Transform

The raster edges of the buildings need to be converted to vector format that can define building outlines as polygons. A similar approach is found in Croitoru and Doytsher (Croitoru & Doytsher, 2004), where the buildings were discovered, separated, and then Hough transform was performed on each building separately. The only difference between Croitoru and Doytsher's process and the proposed approach is how the lines are extracted from Hough space.

The method used in the proposed approach first converted the edge detection results into Hough space using Hough transform. The transform uses the Standard Hough Transform (SHT) equation (Gonzalez & Woods, 2008),

Equation 1: The Hough Transform

$$\rho = x\,cos\theta + y\,sin\theta$$

, where x and y is the location of a pixel in the edge detection results, ρ is the distance of the shortest vector from the image's origin to a line through pixel (x,y), and θ is the angle of the vector from the x-axis. θ's range is from -90° to 90° and ρ's range is from $-(m^2 + n^2)^{1/2} \leq \rho \leq (m^2 + n^2)^{1/2}$, which is the maximum distance from the origin to the farthest image extent (where m is image's columns and n is image's rows), to make the raster parameter space (θ,ρ). The SHT equation is iterated every one degree for a single edge pixel and ρ is rounded to the nearest integer to derive a matrix representing Hough space as shown in Figure 14. For each iteration, a one value is added to the corresponding (θ,ρ) cell which creates maximums in Hough space after all edge pixels are completed representing candidates for edge lines.

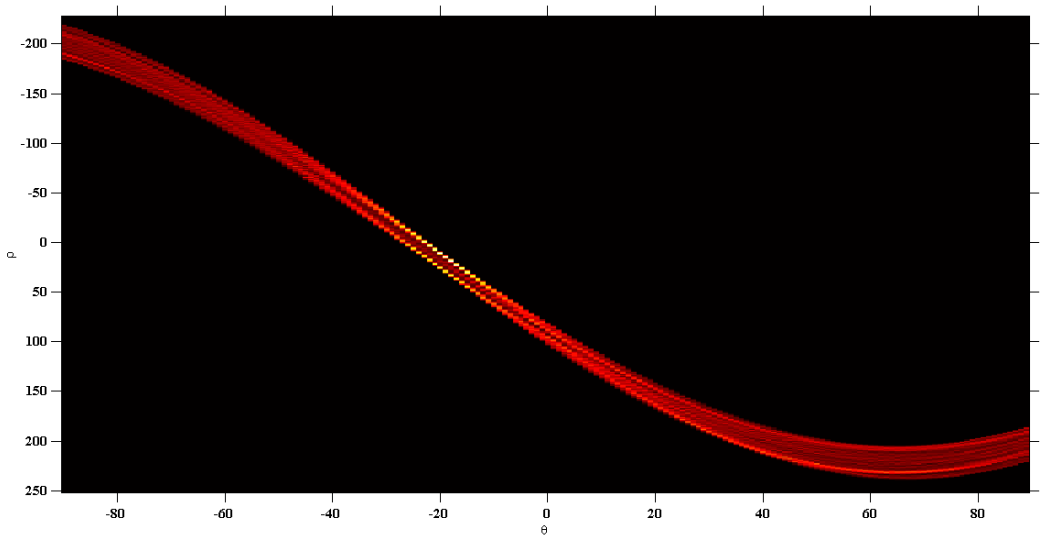


Figure 14: Hough Space, ρ is the y-axis and θ is the x-axis (the whiter the pixel the more edge detection pixels have the same ρ and θ value)

Equation 2: Slope-Intercept Form of the Hough Transform

$$y = -x\,\frac{cos\theta}{sin\theta} + \frac{\rho}{sin\theta}$$

The approach finds the maximums in the Hough space and formulates a line as shown in Equation 2, from that maximum. The line equation is then used to find all edge pixels that intersect the line creating line segments as shown in Figure 15a. A line segment consists of consecutive edge pixels along the line without crossing over a non-edge pixel. Line segments are kept if the segment length is greater than or equal to four pixels and line segments from the same line are joined if a gap of less than or equal to two pixels between segments. The four pixel threshold represents the line segment has to be at least four meters long to be considered a building edge. The gap between line segments cannot be greater than two meters and the line segments considered the edge of the building. The θ value and the maximum Hough space value is kept as part of the

attribute of the line segment where θ is the slope of the line, 0° is a vertical line and 45° is a positive slope line, and the maximum Hough space value is the number of edge pixels that made the complete line in the scene. Next, the edge pixels used to create the line segment are converted non-edge pixel, except the pixels at the beginning and end of the line segment, and the edge detection image is updated. The beginning and end pixels are kept as an edge pixel because of the possibly the end pixels is the corner of the building. If the end pixel is the corner of the building, it is needed to find the adjacent edge of the building.

The updated edge image is converted into Hough space and the creation of line segments is carried out again with the new maximum value from the Hough space. The edge image is again updated removing the edge pixels used to create the line. The cycle of converting the updated edge detection image into Hough space, creating a line from the maximum value in Hough space, and updating the edge image continues until no more line segments meeting the four pixel length requirement is found. This iteration process produces multiple line candidates around the edge of the building as shown in Figure 15b.

At this point in the process one side of the building can contain multiple lines as shown in Figure 15b and the lines are joined according to their angle, θ from the Hough transform that created the line, and its distance in relation to the other lines. The angle between two lines has to be less than or equal to 15° and within two pixels of each other to consider combining them. The 15° and two pixel maximum threshold was a safe estimate to prevent lines representing a different edge combining. This is the first step of

aligning and combining lines and a threshold to combine all the lines on the same edge is not needed. Any missed lines on the same edge are combined later in the line aligning process. The new slope and end points of the combined line segments were determined by a weighted average of each lines' length and Hough space value. The Hough space value is the value at (θ,ρ) used to create the line segment. The weighted average is calculated as:

Equation 3: Distance-Hough Maximum Weighted Average

$$w_i = {(d_i + h_i)} \Big/ {\sum_1^j (d_j + h_j)}$$

where d is the length of the line segment, h is the Hough space value, i is the line or measurement being weighted and j is the number of lines being combined.

To combine the lines, first the angles of the lines are averaged using a weighted average. The new angle, θ, is then used with the x and y value of the weighted centroid, weights to calculate the centroid is determined by using Equation 3, of the lines to create a function representing the combined lines. The end points of the new line is calculated using the new function. Figure 15c (orange line is the new line) using. The weighted centroid uses the weighted average equation and applies it to the end points of the lines (weighted center mass shown as the green dot in Figure 15c). The extents of the new line use the maximum and minimum x value of the end points of the lines. If the angle of the line is greater than 45°, then the maximum and minimum y value is used. The results of combining the initial lines from Hough space is shown in Figure 15d.

Different condition statements in programming this process are needed to handle horizontal lines (θ=90°) and vertical lines (θ=0°) for combining the lines. Condition statements are also needed for direction and steepness of the slope to calculate new endpoints after adjusting or combining lines. If θ was between -45° and 45°, the x-value had to be solved for because of the rapidly changing y-value, due to steep slope. Therefore when lines were combined with steep slopes, the minimum and maximum y-values of the two lines were used to calculate the new end points instead of the minimum and maximum x-values.

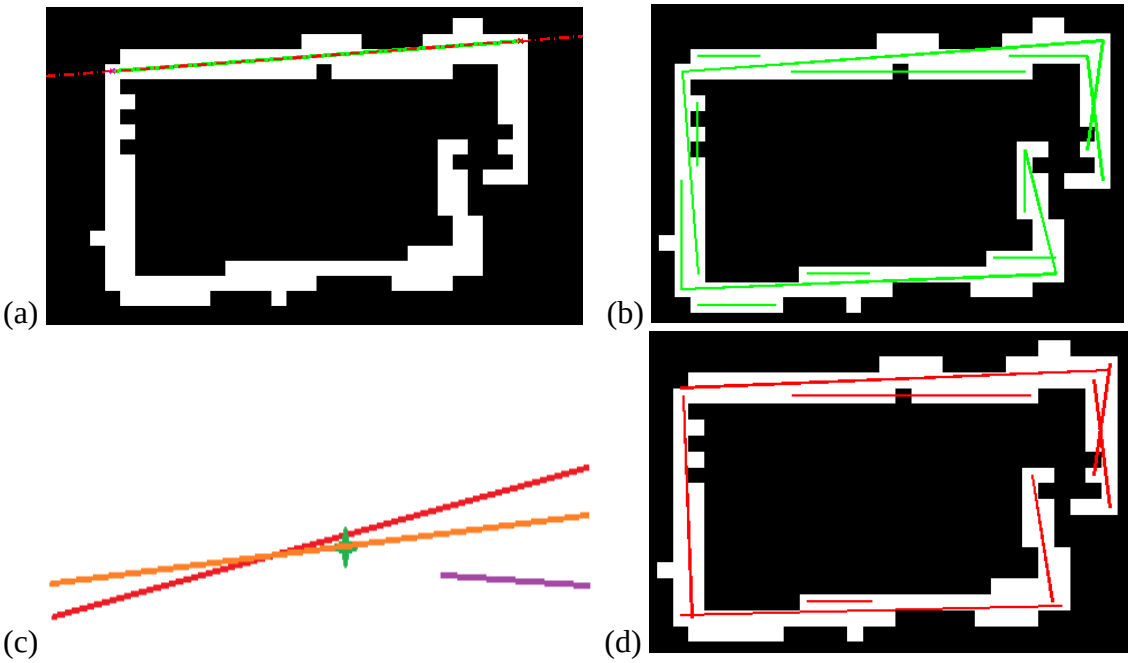


Figure 15: Examples of the Hough Lines Extraction Process (a) one line created from Hough space (b) all lines from Hough space (c) line combination concept (red and purple lines are being combined, green cross is the weighted centroid, and the orange line the new combined line) (d) combining the lines results

4.3 Aligning Building Edges

After the lines are created and combined, further editing and alignment is needed in order to make adjacent edges perpendicular and opposite edges parallel, assuming the buildings were constructed this way and only have 90° angles. The ends of adjacent lines also need to be trimmed or extended so the corresponding end points are topologically connected. Alignment of the lines starts by taking the two longest edges and averaging them so they are either parallel or perpendicular according to their current orientation. The lines θ value are compared to each other by taking the absolute value of $\theta_1 - \theta_2$. If the absolute value was less than or equal to 35° of 0° or 180° then they are considered parallel or if it less than or equal to 35° of -90° or 90° then they are considered perpendicular.

The large 35° threshold is used to include a large range since all sides are assumed to be parallel or perpendicular. If the 35° threshold is not met, then longest and the third longest line are compared for alignment. If that combination doesn't meet the 35° threshold then the second longest and third longest are compared and this iteration continues until the combination of the two longest lines that meet the threshold are found.

For parallel lines, the new lines are rotated about their mid-point to their averaged θ (figure 15a). For the perpendicular lines the averaged difference of the lines' θ value, $\frac{1}{2}(\theta_1 - \theta_2)$, which represents how far each line has to rotate to be perpendicular to one another. If the θ difference was less than 90° then the then the lines were rotated away from each other by the averaged difference and vice versa if the θ difference is greater than 90° (shown in figure 15b). The rest of the edge lines are then rotated to be

perpendicular or parallel to the θ established from averaging the two longest lines if they are within 35°. Lines that are not within 35° are deleted.

Another iteration of combining lines as discussed in Section 4.2 is completed before endpoints of adjacent lines are connected. End points are connected by trimming or extending adjacent lines to their intersection. First two lines have to be 90° from each other to make sure they are possible adjacent edges and secondly the endpoints had to be within two pixels in order to connect them. The two pixel threshold is used to avoid connecting endpoints that could end up extending a line through an extrusion from the building. The intersection of the two lines is calculated by setting each line's linear slope-intercept equation equal to each other, as expressed by the following equation:

Equation 4: Hough Lines Intersection Formula

$$-x\left({cos\theta_1}/{sin\theta_1}\right)+\left({\rho}/{sin\theta_1}\right)=-x\left({cos\theta_2}/{sin\theta_2}\right)+\left({\rho}/{sin\theta_2}\right)$$

A solution is derived solving for x and substituting x back into one of the line's slope-intercept equation. This (x,y) pair is then used as the connection point for the two lines and the lines were trimmed or extended to that point. Checks were needed for connecting the end points to make sure the correct and closest endpoints were being connected. Otherwise the algorithm could connect lines from the opposite side of the building.

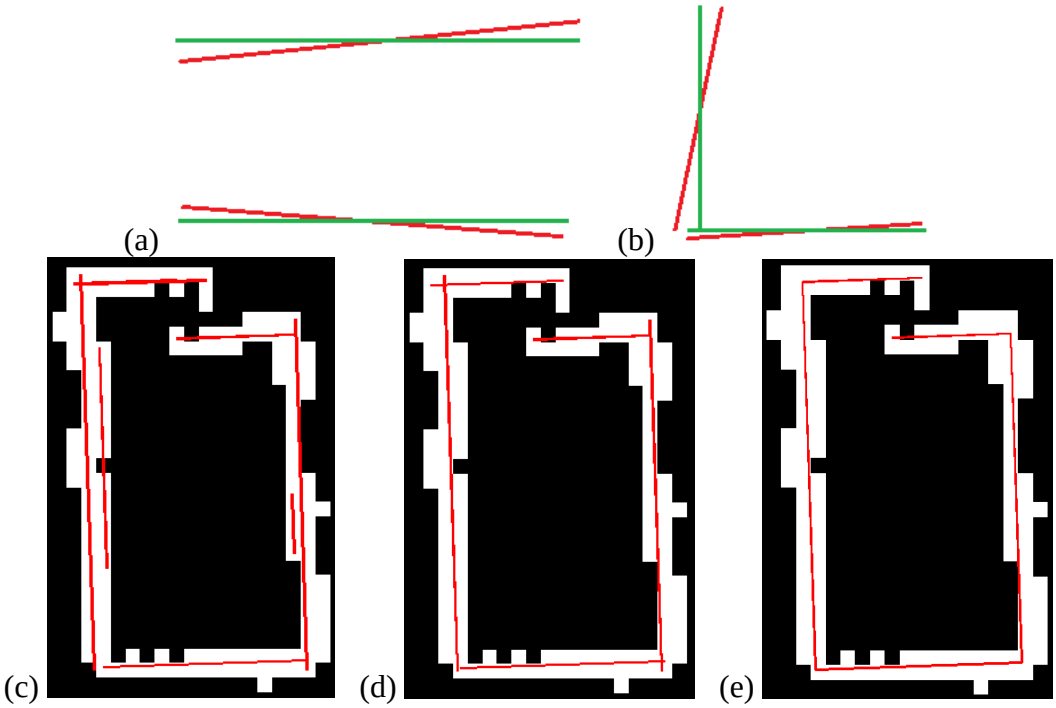


Figure 16: The Edge Alignment Process
(a) aligning parallel lines (b) aligning perpendicular lines (c) all edges aligned (d) second iteration of combining lines (e) connecting end points

4.4 Completing the Polygon

This next step is used to add, trim, or extend lines to connect endpoints that are still not connected because of previous constraints on creating and editing lines. The algorithm connects unconnected endpoints that are closest to each other and then connects the next closest unconnected endpoints. If the endpoints being connected are part of lines that are perpendicular then the lines were trimmed or extended to their intersection point as discussed in Section 4.3.

If the lines of the endpoints are parallel then the lines are combined or a new line is created connecting the two lines depending on the distance between the two lines. If the lines are within four pixels then they are combined as described in Section 4.2,

otherwise a line is added perpendicular to the two lines to connect the endpoints. The four pixel threshold has been the minimum length for a buildings edge and if the parallel lines are separated more than four lines, then that edge of the building was missed during the Hough transformation process. The new line is created to pass through the weighted average (Equation 3) of the unconnected points and the parallel lines are trimmed or extended to intersect the new perpendicular line and the same was done with the new line as shown in Figure 17.

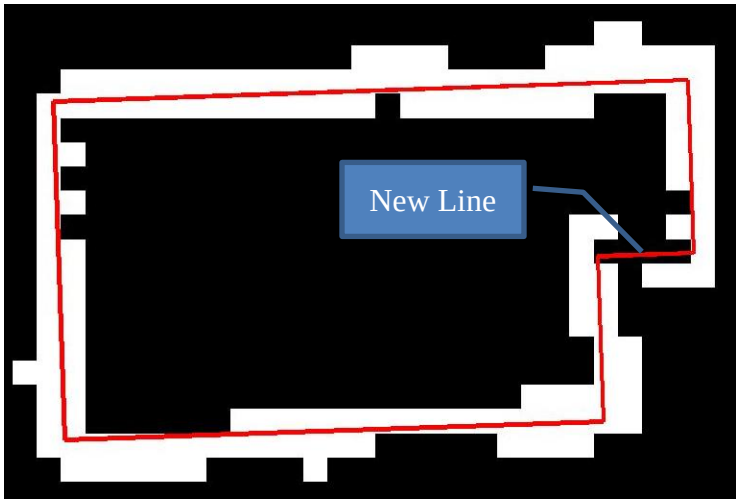


Figure 17: Completed Polygon

In the final stage of the process the building polygons are created. The edge lines are not in any particular order. Therefore the lines are ordered by starting with one line and finding the connected line, then finding the next connected line until the next connected line is the line used to start this stage in the process. The order of vertices, clockwise or counter-clockwise, are calculated by determining how many vertices have a

positive or negative cross product between adjacent vertices as shown in Figure 18a (Bourke, 1998). If there are more vertices with a positive cross product then the vertices' order was counter-clockwise and the order was reversed for creating a vector data layer.

The size of the building needs to be expanded to correct for a systematic bias in the proposed approach. The vertices expanded from the center of the polygon by a distance of two pixels. The distance of two was determined by testing different distances and is discussed in Section 5.3. The direction of the expansion is defined by the direction of a vector created half between adjacent lines and the cross product. If the cross product was positive then the vertex was expanded in the opposite direction of the vector. If the cross product was negative, the vertex was expanded the same direction as the vector.

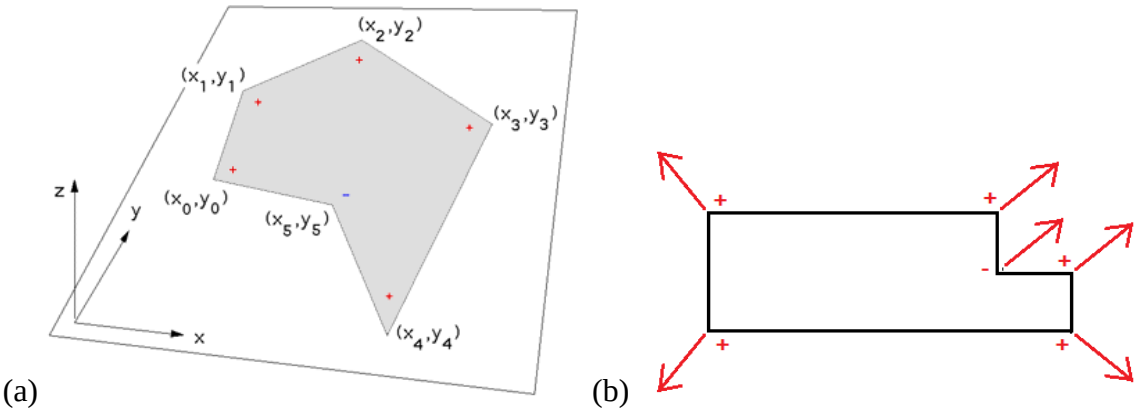


Figure 18: Vertices Cross Product (a) Right hand rule cross product (Bourke, 1998) (b) direction of expansion based on cross product

CHAPTER 5: EXPERIMENT RESULTS

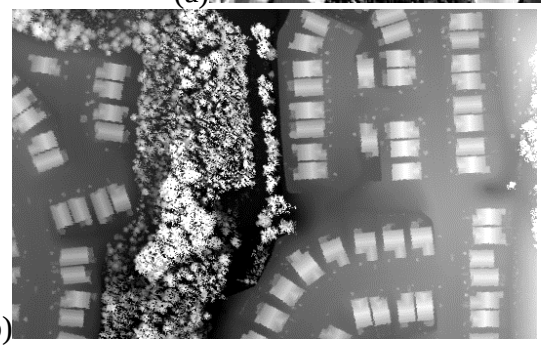
The first section of this chapter describes the data used for the experiments and the data used for comparing the extracted footprints to three different datasets; shapefile of structures in the test area's county, digitized buildings from the DSM used for the tests, and digitized building from high resolution orthorectified imagery. This chapter discusses the results and analysis of the results of the approach. The limitations and advantages of the proposed approach are also discussed.

5.1 Data

The aerial LiDAR data used in the research was produced by the Army Geospatial Center (AGC) and permission was granted by the AGC for the use of this data in this thesis. The LiDAR was collected over the Fredericksburg, VA area (Figure 19) and the point cloud was transformed into a first return and last return regular grid and intensity image with a sub-meter post spacing (resolution). The area of the collection is five kilometer by three kilometer (5km X 3km) and a three tenths kilometer by two tenths kilometer (0.3km X 0.2km) subset of the collected area was used to perform the experiments. The DEMs are projected using the WGS 84 UTM Zone 18N coordinate system. The subset consists of few different shaped buildings ranging from simple rectangles to more complex shapes at various angles throughout the DEM. Trees, bushes, and rolling terrain added complexity to the scene.

(a)

(b)



(c)

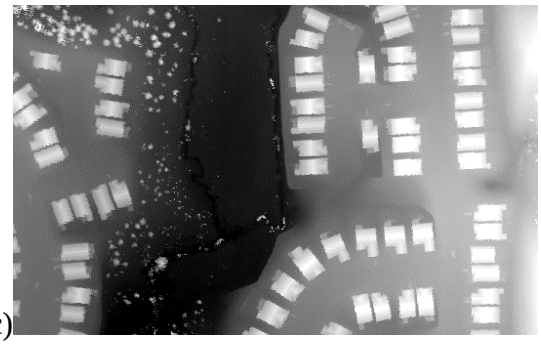


Figure 19: LiDAR Data Test Area (a) first return shaded relief combined with intensity image (b) first return (c) second return

The three datasets used for comparison consist of data of all structures in Stafford County from the county's website (http://www.staffordcountyva.gov/index.aspx?NID=1319), digitized buildings from colored high resolution orthorectified imagery downloaded from the county's website, and digitized buildings from the first return DSM. The structures and the colored imagery were projected in NAD 83 Virginia State Plane and the structures data and digitized buildings from the colored imagery were re-projected to WGS 84 UTM Zone 18. The image footprint of Stafford County is approximately 30.7 kilometers by 50 kilometers

(30.7km X 50km) with one foot (0.34 meters) resolution. The image was downloaded in quadrants in MrSID format.

There was poor correlation between the structures downloaded from the county's website and the digitized buildings from the imagery and DSM. The method of how Stafford County collected the structure data is unknown but comparing this data to the LiDAR data revealed inaccuracies in official data suggesting the need for an improved data collection process. Table 2 in Section 5.2 shows the correlation between the datasets and Figure 21 presents a visual comparison. Also, visually there appears to be a much stronger correlation between the digitized datasets, therefore the discussion of the results will be focused on comparing the extracted footprints to the digitized datasets.

(a)  (b) 

Figure 20: Colored High Resolution Imagery (a) image of test area (b) a close-up view of buildings in test area

(a) 

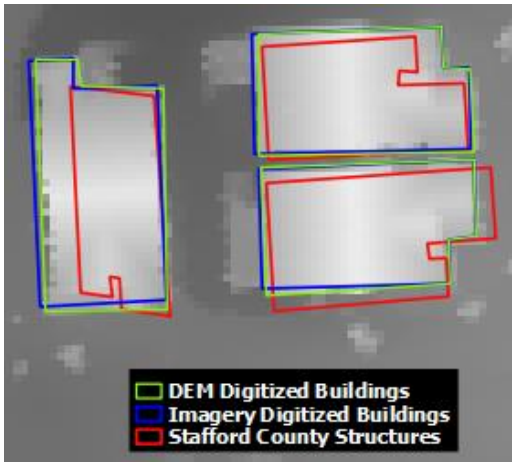
(b) 

Figure 21: Stafford County's Structures Visual Comparison (a)Comparison on colored image (b) comparison on LiDAR DEM

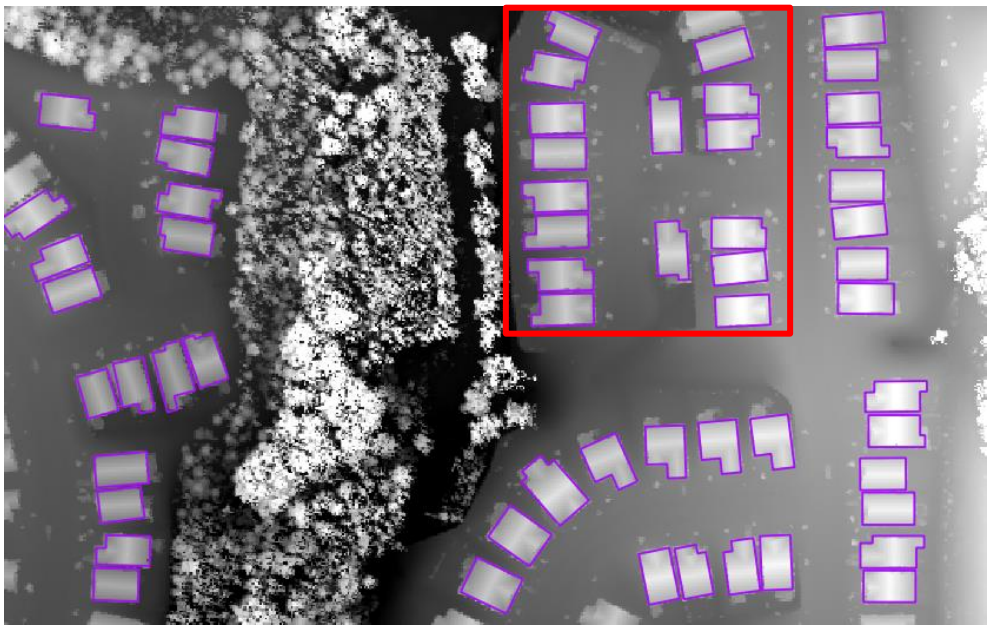
Figure 22: Stafford County in Virginia (https://www.openstreetmap.org/export#map=8/37.892/-77.761&layers=Q)

5.2 The Extracted Footprints

The experimented approach detected all 57 buildings in the scene and extracted the footprints shown in Figure 23 and Figure 24. The algorithm performed according to the proposed approach and removed vegetation, detected edges, and constructed

footprints. The extracted footprints have 90° angles consistent with the buildings in the scene and the buildings were properly separated.

(a)



(b)

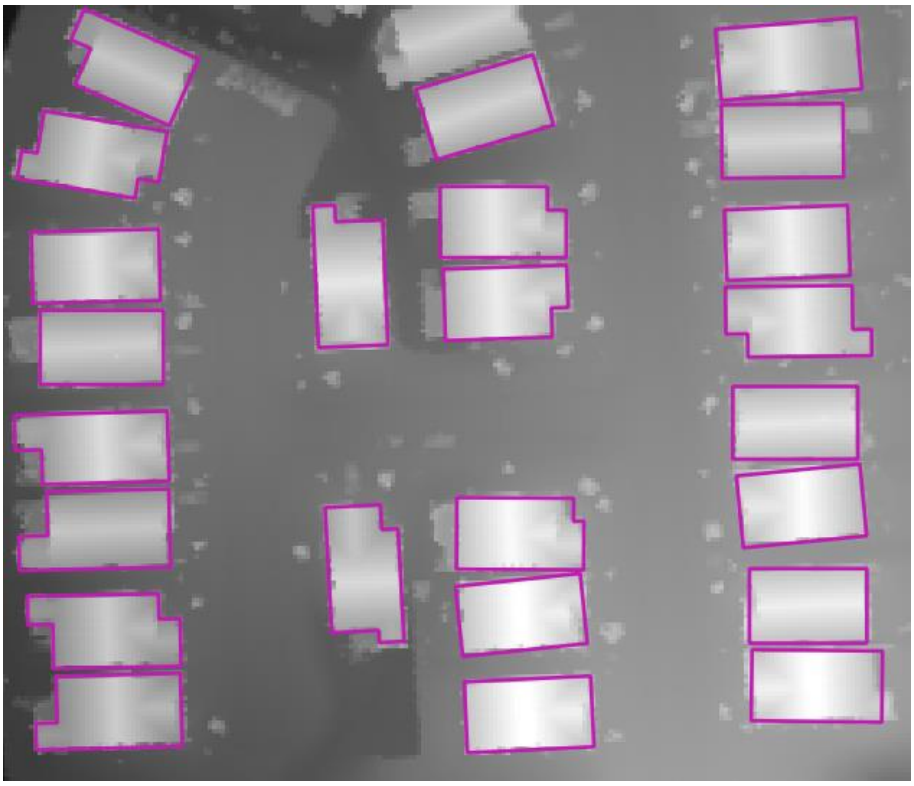


Figure 23: Extracted Footprints Overlaid on the LiDAR's First Return DSM (extracted footprints in purple) (a) extracted footprints for test area (b) close-up of area outlined by red box in (a)

(a)

(b)

Figure 24: Extracted Footprints Overlaid on the Colored High Resolution Image (extracted footprints in purple) (a) extracted footprints for test area (b) close-up of area outlined by red box in (a)

5.3 Method of Evaluation

The 57 buildings from each dataset and the extracted footprints were compared to the corresponding buildings one by one in the Stafford County's structures and the digitized datasets. Each extracted footprint and the corresponding building in the other datasets were assigned the same identification number to ensure the correct buildings were compared. The datasets where analyzed and the buildings' perimeter, area, and centroid were calculated in QGIS 2.8.1. The following statistics were used for the one to one comparisons:

Equation 5: Area/Perimeter Error

$$\delta_a/\delta_p = \frac{M_d - M_e}{M_d} * 100$$

Equation 6: Centroid Error

$$\epsilon = \sqrt{(x_d - x_{e)}{}^2 + (y_d - y_{e)}{}^2}$$

Equation 7: Percent Overlapped Area

$$\gamma = \frac{A_d \cap A_e}{A_d} * 100$$

Equation 8: Commission (Zhang et al., 2006)

$$\alpha = \frac{A_{eNON}}{A_d} * 100$$

Equation 9: Omission (Zhang et al., 2006)

$$\beta = \frac{A_{dNON}}{A_d} * 100$$

Equation 10: Percent Accuracy

$$\eta = \frac{(A_d \cap A_e) - A_{eNON}}{A_d} * 100$$

Table 1: Equations' Symbols

Symbols	Description	Symbols	Description	Subscript	Description
δ_a	Area Error	M	Measured area/perimeter	e	Extracted Footprint
δ_b	Perimeter Error			d	Digitized Building/Structure
ϵ	Centroid Error	x	x-coordinate		
γ	Overlapped Area	y	y-coordinate		
α	Commission	A	Area		
β	Omission	A_{NON}	Non-Overlapping Area		
η	Accuracy				

The percent area/perimeter error shows the relationship between the size of the digitized/structure footprint and the automated polygon while the centroid error shows if there is an offset between the two footprints. The percentage of the overlapped area (represents precision) between the two footprints shows how well the automated polygon fits to the digitized/structure footprint while the average of all the percentage of overlapped areas shows precision. Commission (percentage of false positive area) show shows how much of the automated footprints does not fit to the digitized/structure footprints as the false positive due to rotation, offset, and contraction while the omission (percentage of false negative) shows the incorrectly identified areas as the false negative due to any rotation, offset, and enlargement. The overall accuracy is produced by the percent accuracy equation which shows how accurate the edge detection performed by subtracting the incorrectly identified areas from the correctly identified areas. Minimum,

maximum, average, and standard deviation (Std Dev in tables) use all the one to one comparisons for that measurement (i.e. area error, perimeter error, centroid error, etc.).

5.4 The Comparison Between Tests and Digitized Buildings

As mentioned previously, the structures from Stafford County did not correlate well with the digitized datasets and therefore, only the comparison between the extracted footprints and the digitized datasets is discussed in detail in this section. The automated footprints were still compared with each other to show the possible inaccuracies from official authorities which highlighted the need for updated building footprints from older techniques. The statistical comparison of the extracted footprints to the digitized datasets in Table 2 shows the magnitude and direction of the area error of 15% and 18% and an accuracy of 58% and 56%. The table also shows the digitized datasets are correlated to each other since the magnitudes and directions are similar when comparing the structures to the Imagery Digitized buildings and the structures to the DEM Digitized buildings. Because the structures are not correlated the comparisons discussed in this section is focused on the extracted footprints and digitized datasets.

Overall the results are promising for the researched approach with respect to accuracy and precision. There are two test sets discussed in this section: Test Set 1 and Test Set 2. Test Set 1 was created by the algorithm discussed in Sections 4.2 - 4.4 without the expansion discussed at the end of 4.4 and the former constraints of minimum line length of five and maximum gap of three. Test Set 2 used the expanded footprints and updated constraints.

Table 2: Stafford County's Structures Statistical Comparison

	Area Error (%)		Area Overlapped (%)	
	Imagery Digitized	DEM Digitized	Imagery Digitized	DEM Digitized
Min	-7%	-1%	61%	59%
Max	37%	37%	92%	91%
Avg	15%	18%	79%	78%
Std Dev	8%	8%	7%	7%
	Commission Error (%)		Omission Error (%)	
	Imagery Digitized	DEM Digitized	Imagery Digitized	DEM Digitized
Min	8.4%	9.1%	0.0%	0.0%
Max	39.2%	40.8%	21.1%	18.2%
Avg	21.2%	22.1%	6.5%	4.5%
Std Dev	7.3%	6.8%	5.7%	4.3%
	Centroid Error (m)		Accuracy (%)	
	Imagery Digitized	DEM Digitized	Imagery Digitized	DEM Digitized
Min	0.15	0.16	21.67%	18.32%
Max	3.10	2.81	83.15%	81.83%
Avg	1.25	1.20	57.63%	55.72%
Std Dev	0.74	0.67	14.61%	13.69%

Through testing the approach, some biases and inaccuracies were observed in Test Set 1. The inaccuracies were solved by correcting the initial constraints set in Section 4.2. When creating the lines from Hough space the length of line segments were constrained to be five pixels or larger and gaps between line segments had to be joined within three pixels. With these constraints some extrusions were excluded from the final footprint as seen in Figure 25. Reducing the constraints for the length of line segments to four pixels

and gaps to two pixels allowed more extrusions to be detected and included in the footprint.

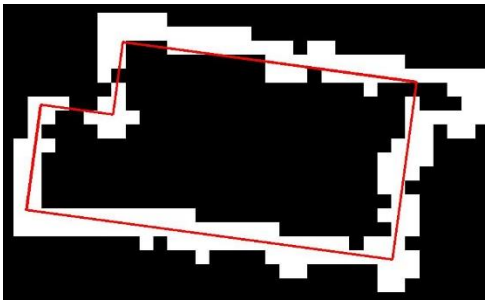
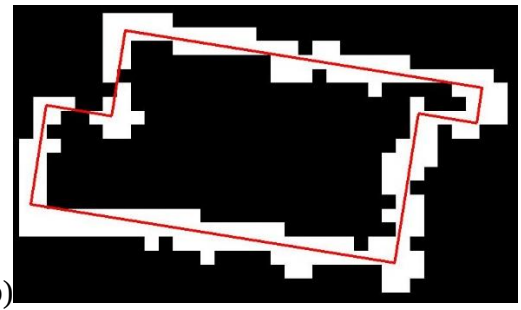
(a)  (b) 

Figure 25: Constraint Correction from Test Set 1 (a) and Test Set 2 (b)

Figure 27 shows that the area and perimeter were consistently smaller than the digitized and structure footprints. This bias is possibly a result of using the last return DSM and using Hough transform on thick image edges. The last return is the return of a laser pulse that traveled the furthest and therefore the elevation of the actual edge at or near ground level. The actual edge of the building was likely observed in the first return DSM but since vegetation is prevalent in the first return, last return was chosen for edge detection. Hough transform extracted multiple lines for the same edge because of the thick image edges. The final result of combining the lines averaged the lines closer to the middle of the thick image edge which is further away from the edge of the building as shown in Figure 26. From three tests of increasing the expansion it was determined a distance of approximately two pixels of expansion in all directions provided more accurate results in terms of area and perimeter sizes and the final results are shown in

Figure 28. A possible slight bias that the automated footprint has a smaller perimeter still remains in Test Set 2.

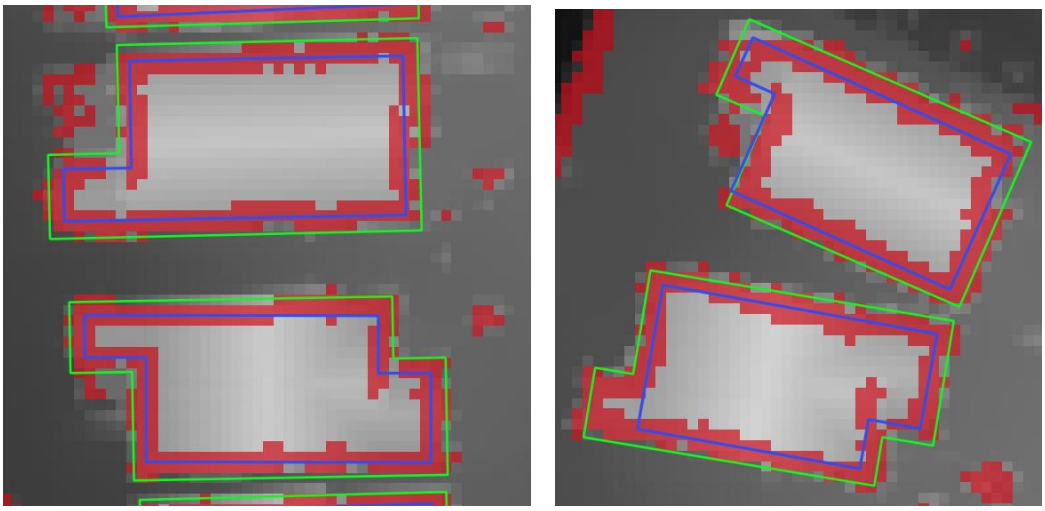


Figure 26: Test Set 1 Compared to Test Set 2, Edge Detection, and First Return DSM Test Set 1 is the blue polygons, Test Set 2 is the green polygons, red pixels are the edge detection results, and grey scale pixels is the first return DSM.

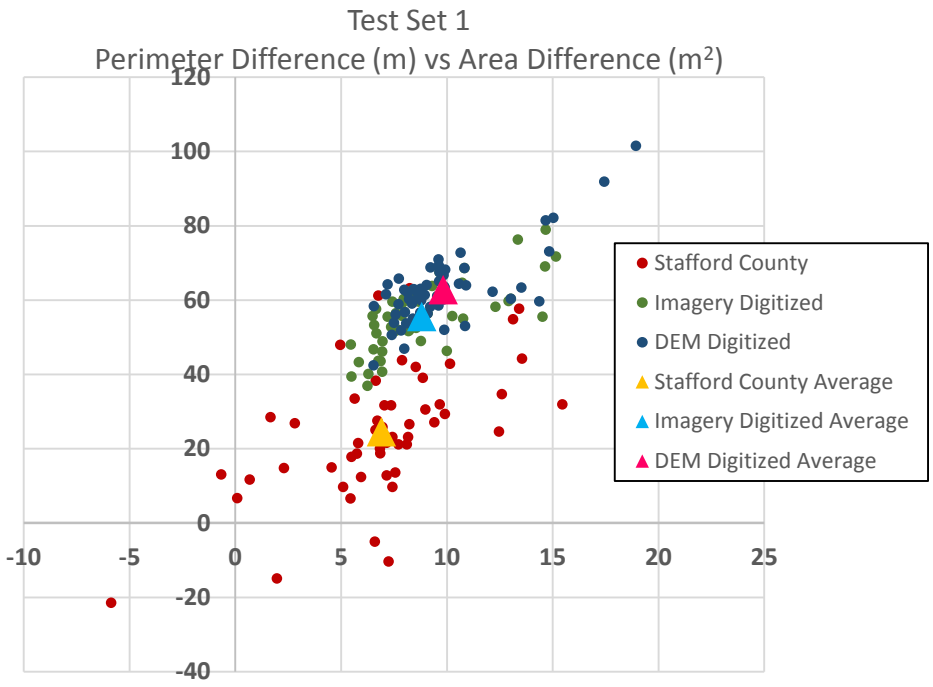


Figure 27: Test Set 1 Bias (perimeter on the x-axis and area on the y-axis)

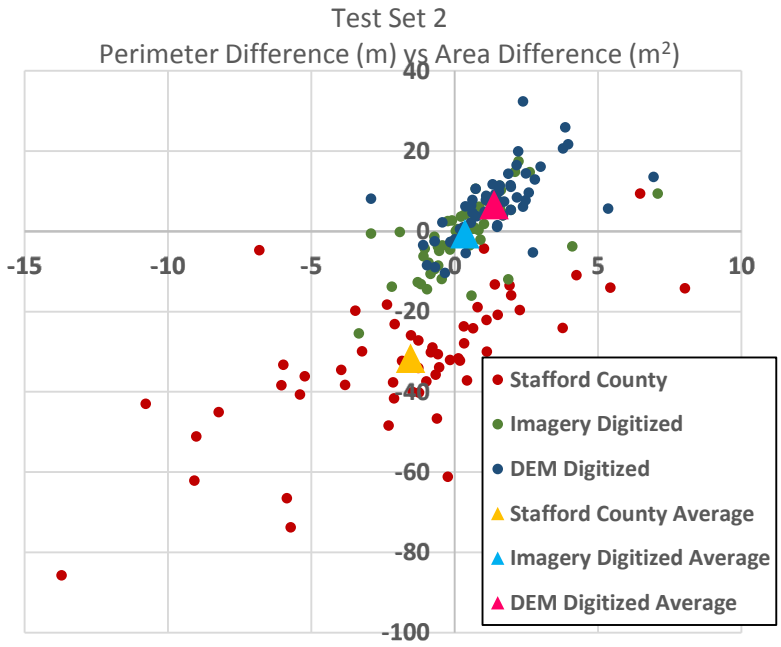
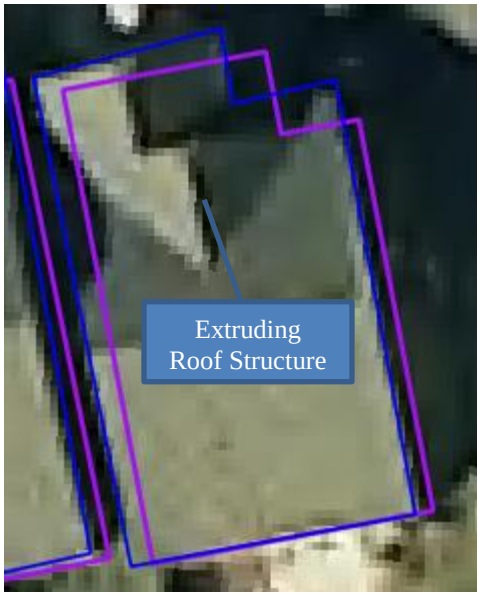


Figure 28: Test Set 2 with Bias Correction (perimeter on the x-axis and area on the y-axis)

The distance between centroids (centroid error) between Test Set 1 and Test Set 2 changed slightly and the average of the centroid errors were visually unnoticeable using Figure 30 and Figure 31. There was a slight increase in precision in the comparison of the digitized datasets and Test Set 2 possibly due to the fact that Test Set 2 was more sensitive to extrusions from the building with the updated constraints. Visually there was a small offset between the DEM and imagery possibly causing a small bias where the automated footprints are consistently to the right of the imagery digitized buildings which is shown in Figure 31.

There was one instance observed where a large dormer, or tall extruding roof structure, extended upwards from the main section of the room causing edge detection results and ultimately reducing the performance of the approach (Figure 29).

(a) 

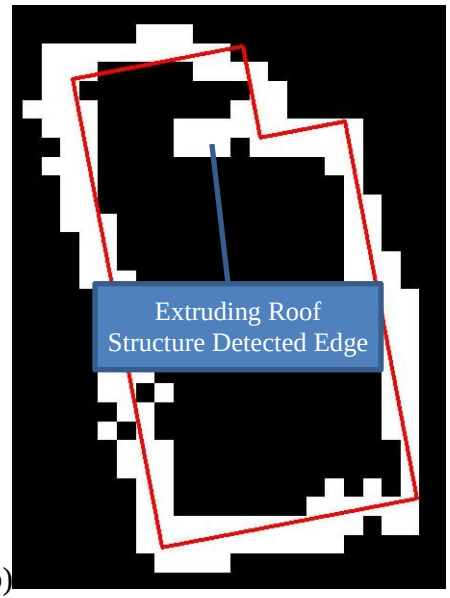
(b) 

Figure 29: Tall Extruding Roof Structure Affecting Performance of Approach (a) Image showing extruding roof structure, the extracted footprint (purple polygon), and the image digitized building (blue polygon) (b) image showing extracted footprint (red polygon) and edge detection (white pixels)

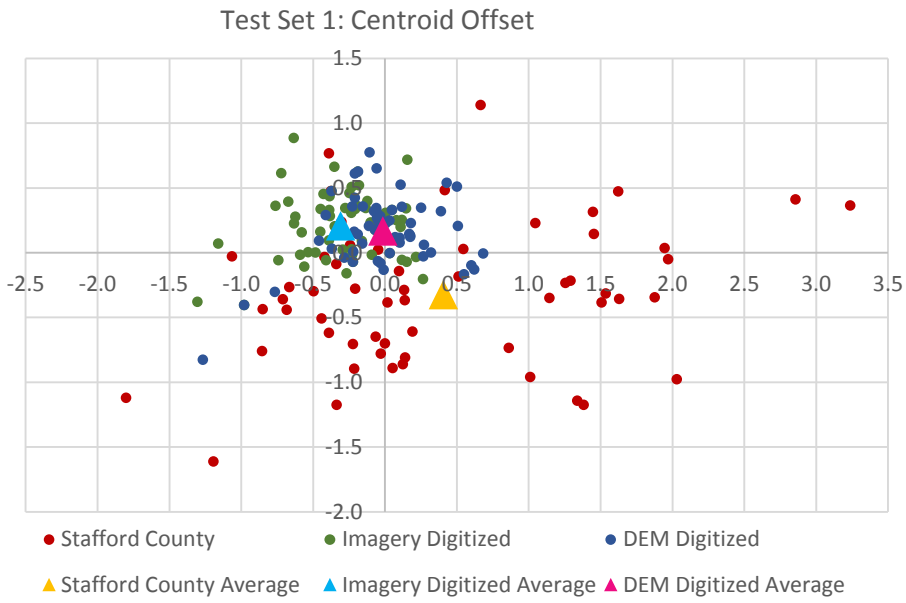


Figure 30: Set 1 Centroid Error Accuracy and Precision

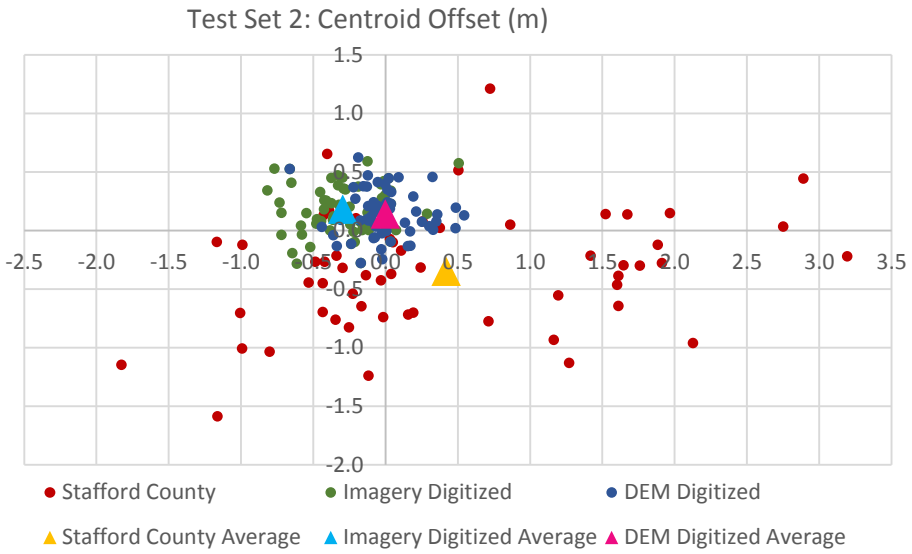


Figure 31: Set 2 Centroid Error Accuracy and Precision

Another inaccuracy in the extracted footprint compared to the digitized buildings was rotation. Part of the percent accuracy equation (Equation 10) is subtracting the area that did not intersect the digitized from the intersecting area. This accounted for rotation where if the automated footprint was the same size and shape as the as the digitized building then it would reduce the accuracy. This also accounted for buildings that were bigger than the digitized buildings but on average they were smaller suggesting offset or rotation. The bias in centroid offset was very small which leaves rotation being the main cause for inaccuracies. This was more prevalent in the comparison of the DEM digitized buildings since the average accuracy percentage is lower for the DEM digitized buildings than the Imagery digitized.

Possible sources for inaccuracy include using too thick of edge detected boundaries, gathering all lines from Hough space rather only using one or two lines per

side, using the wrong weighting or method for combining line, or possibly a combination of all of them. The rotation also had an adverse effect on the building's alignment relative to the others around it. Most of the buildings that were a few meters apart where parallel with each other but visually some footprints were not. Also, visually there was no correlation between the orientation of the building in the image and the amount of rotation.

Figure 32 and Figure 33 shows the distribution of the area difference, area of digitized building subtracted by the automated footprint, along with the normal distribution curve for the dataset. The comparison with the DEM digitized buildings has a slight positive bias compared to the average showing that the automated footprints, on average, are smaller but there is no bias with imagery digitized buildings.

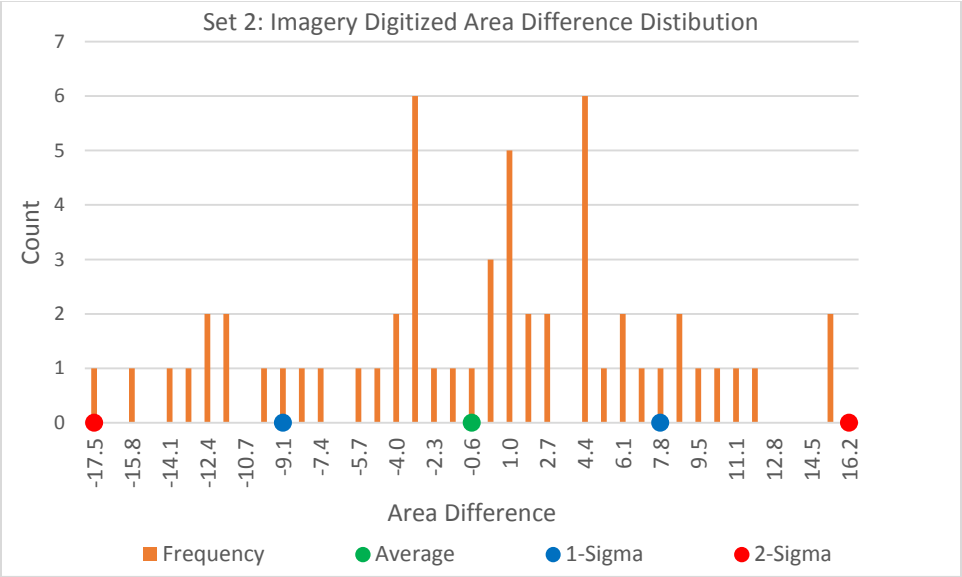


Figure 32: Imagery Digitized Area Difference Distribution

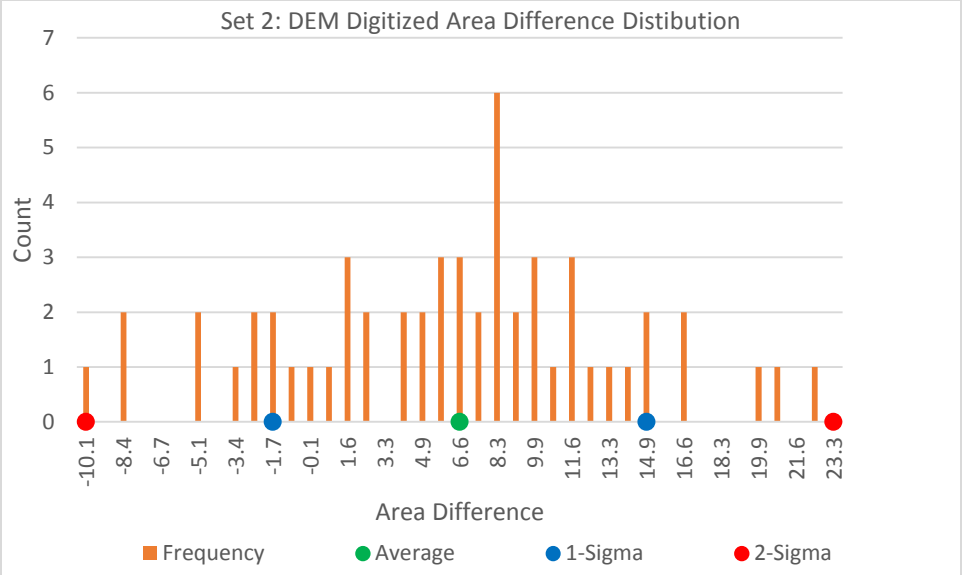


Figure 33: DEM Digitized Area Difference Distribution

Table 3 shows the statistical analysis of the comparison study. This analysis shows great promise for the approach discussed in this thesis. The commission and omission errors of 5% and 6%, respectively for imagery digitized buildings, proves very little of the automated footprints did not overlap the imagery digitized buildings and vice versa. The area error of 0.3% and 3.1%, comparison to the imagery digitized and DEM digitized buildings respectively, shows the extracted footprints on average have the same size as the digitized buildings and the area overlapped was 95% and 94% showing 5% and 6% of the extracted footprints did not overlap the digitized buildings. The extracted buildings had a maximum centroid error of 0.9 and 0.8 meters showing the center of all the extracted footprints were within 1 meter of the digitized buildings' center. The accuracy also shows that on average you can expect an approximately 90% match to buildings in orthorectified, high resolution imagery.

Table 3: Statistical Comparison of Automated Footprints

	Area Error (%)			Area Overlapped (%)		
	Stafford County	Imagery Digitized	DEM Digitized	Stafford County	Imagery Digitized	DEM Digitized
Min	-53.8%	-12.5%	-4.5%	77.7%	86.5%	82.9%
Max	4.6%	7.6%	14.8%	100.0%	99.4%	99.3%
Avg	-18.7%	-0.3%	3.1%	94.0%	94.9%	93.9%
Std Dev	11.2%	4.1%	3.8%	5.5%	2.7%	3.1%
	Commission Error (%)			Omission Error (%)		
	Stafford County	Imagery Digitized	DEM Digitized	Stafford County	Imagery Digitized	DEM Digitized
Min	0.0%	0.6%	0.7%	8.6%	0.5%	0.1%
Max	22.3%	13.5%	17.1%	55.2%	13.1%	8.3%
Avg	6.0%	5.1%	6.1%	24.7%	5.5%	3.0%
Std Dev	5.5%	2.7%	3.1%	10.4%	2.7%	2.1%
	Centroid Error (m)			Accuracy (%)		
	Stafford County	Imagery Digitized	DEM Digitized	Stafford County	Imagery Digitized	DEM Digitized
Min	0.08	0.07	0.01	55.4%	73.1%	65.7%
Max	3.20	0.93	0.84	100.00%	98.9%	98.7%
Avg	1.12	0.44	0.29	88.0%	89.7%	87.8%
Std Dev	0.73	0.21	0.16	11.1%	5.4%	6.2%

While this thesis's approach performed well in the experiments described within, several factors could affect its performance. The approach was tested on a relatively small sample of only 57 buildings that were simple or slightly complex, contained no trees overhanging roofs, and contained consistent rolling terrain over the scene. With more complex buildings not all adjacent edges are 90° from each other as assumed in the proposed approach. If angles other than 90° were to be considered in this approach, it

may have produced edges that were not as accurate as the results presented here. One of the constraints for aligning edges to the correct angle was that it needs to be within 35° from the calculated orientation in Section 4.3. 35° is a wide constraint and would have to be lowered to account for angles other than 90° for adjacent sides. This could lead to sides or parts of a side that are improperly aligned. This approach should be able to handle more complex buildings if adjacent edges are 90° but have eight or more sides; which is more than the sample sets used in this experiment.

The approach in this thesis was able to remove enough vegetation to eliminate any influences but if there were trees overhanging roofs, a slightly different approach may be needed. The vegetation mask created in Section 3.1 is used to remove most of the vegetation in the edge detection results which in turn could remove edges of a building if the mask contained an overhanging tree. If the building edge is not complete it will not be segmented in the watershed results, and ultimately not labeled as a building. The edge detection is performed on the last return DSM and could possibly not contain that tree and also likely not contain the outermost edges of tree area which utilizes LiDAR's FOPEN capability. A possible solution to this issue is to use an erosion morphological filter on the vegetation mask to decrease the tree areas and in turn remove the overhanging tree in the mask. Figure 34 depicts a scenario, in which the green area represents a possible vegetation mask and the blue area represents the vegetation mask with the erosion morphological filter applied. In this case, the tree will be removed from the mask and not affect the detected building edge from the last return DSM.

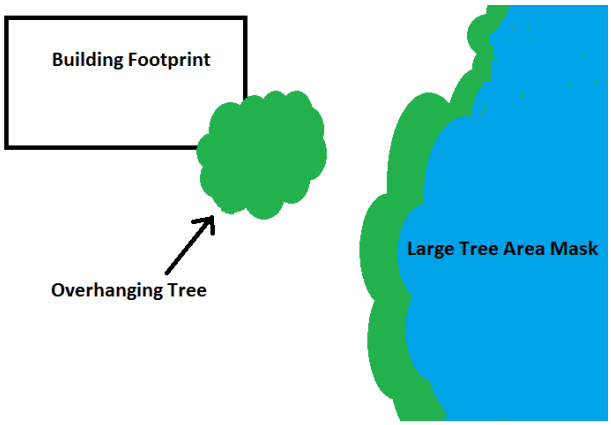


Figure 34: Overhanging Tree Removal (green area is possible vegetation mask and blue area is after an erosion morphological filter is applied to the vegetation mask)

The terrain could also affect the results of this proposed approach. In the experiments presented, all measurements were kept relative to small and specific local areas, reducing possible influences of the terrain. Other potential error sources include buildings on steep terrain or in cities with tall non-building objects such as overpasses, suspended exits, or entrance ramps.

CHAPTER 6: CONCLUSION

The results from these extracted footprints compared to the digitized datasets show the approach can be consistent and accurate and most importantly there is no human interaction involved after the creation of the first and last return DSMs.

LiDAR is not a new technology and was intended for mapping particles. Through advancements in technology it has become a useful tool for mapping the surface of the Earth. Through the years, LiDAR has surpassed traditional photogrammetry techniques in efficiency and cost for complicated terrain modeling. The phenomenology and capabilities, such as FOPEN, are being studied for applications to extract additional information about objects in the data, such as buildings.

Buildings are an important object to depict from remotely sensed data for several applications including urban planning, environmental monitoring, 3D modeling, and military operations. These applications rely heavily on the accuracy and efficiency of building datasets in order to perform proper analysis and produce quality products.

This thesis presented an approach to automate building footprint extraction from LiDAR data providing accurate and consistent results using a combination of the new capabilities of LiDAR and the proven techniques, like raster processing, used to exploit older and more understood data. Through literature review it was discovered that there is minimal research using DEMs generated from LiDAR using proven techniques for

imagery. Instead of the pixels in a raster representing spectral information it represents spatial information. There are no limitations for the imagery proven techniques of edge detection according to what the pixel value represents and therefore can be used on DEMs.

The approach of this research uses the same concept of edge detection for imagery in that local relative comparisons are made to discover local discontinuities. The rotating kernel used in this approach defined complete and closed building edges and used the watershed operator to identify and segment the buildings. Each segmented building was then transformed into Hough space and building edges were extracted as line segments. Finally, the line segments were processed through an algorithm to combine and align forming the buildings' footprint as polygons.

Three datasets were used to compare the extracted footprints created in the approach for accuracy and precision: structures downloaded from Stafford County's website and two sets of digitized buildings from the LiDAR DEM used in this research and a high resolution orthorectified image. The structures did not correlate well to the digitized data or the automated footprints. The automated footprints correlated very well with the digitized data resulting in commission and omission errors of 5% and 6% when compared to the imagery digitized buildings and 6% and 3% when compared to the DEM digitized buildings, showing the accuracy and precision of the approach in this thesis. The results and statistical analysis also prove great promise of the approach to replace or supplement traditional timely, ineffective, and costly photogrammetric processes.

More research could be done to ensure its consistency in producing these or similar results using data from other environments. The study area only contained simple and no complex buildings with adjacent sides 90° from each other. Further research modifying the Hough lines alignment algorithm could determine if the concepts expressed in this thesis can be applied to complicated scenes. Diverse terrain types and environments, such as steep slopes and heavy forests, could also be researched with this approach.

One reason for commission error (false positives) is the rotation of the extracted footprints from the digitized datasets. Possible solutions for providing a better orientation of the extracted footprint is to use other local features. In a housing development most houses are parallel and in the same orientation as the houses around them. After using the proposed approach footprints within a close proximity can be used to align the footprints to each other providing better orientations and fewer commission errors. Roads can also be used since most buildings are perpendicular to the road.

Testing needs to be done on LiDAR datasets with occlusions such as LiDAR shadows and LiDAR system artifacts. LiDAR shadows can occur from tall buildings or when collecting an area off nadir where no returns are recorded on the opposite side of objects. LiDAR system artifacts could be caused by GPS recording issues, incorrect settings on the sensor for specific collections, and more. The test dataset used in experimenting the proposed approach did not contain artifacts or shadows. Testing on flawed datasets should be done to observe the effect it would have on the proposed approach.

Three-dimensional datasets are also becoming more in-demand for analysis and visualization. The automated footprints were created from 3D data which future research could exploit and translate the heights in the DEMs to the footprints creating 3D vector datasets. There are multiple ways to detect edges using traditional kernels, such as Roberts, Sobel, and Laplacian, and each have their advantages and limitations but could provide additional information about the building and structures on the roof. A suggestion for future research is to study each of the kernels to determine if one kernel or a combination of more than one kernel can provide better results than this study or be used to model and classify different types of buildings.

APPENDIX – COMPARISON TABLE FOR EACH EXTRACTED BUILDING

Table 4: Comparison Table Part 1a

	Area Difference			Perimeter Difference			X-Centroid Difference			Y-Centroid Difference		
Building #	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized
1	-4.40	4.31	6.81	1.03	1.70	1.55	-0.48	-0.24	-0.07	-0.27	0.05	-0.06
2	-73.84	0.57	9.83	-5.71	0.67	1.56	-1.16	-0.36	-0.12	-1.59	0.12	0.09
3	-48.47	-3.39	4.80	-2.30	-0.16	1.06	-0.12	-0.03	0.33	-1.24	0.39	0.46
4	-19.69	0.29	8.37	2.28	0.84	2.19	-1.83	-0.41	-0.08	-1.15	0.25	0.24
5	-37.25	-2.15	1.48	0.45	0.91	1.50	-0.42	-0.73	-0.33	-0.27	0.24	-0.14
6	-35.77	2.65	2.18	-0.66	-0.08	-0.41	-0.25	-0.16	0.17	-0.83	0.00	-0.13
7	-14.12	-12.06	-5.35	5.44	1.88	2.75	0.72	-0.12	0.00	1.21	0.59	0.40
8	-61.19	0.88	7.39	-0.23	0.70	1.73	0.51	-0.19	0.01	0.51	0.37	0.35
9	-66.59	17.44	19.90	-5.84	2.24	2.23	1.61	-0.47	-0.03	-0.65	0.09	-0.16
10	-37.67	0.47	6.40	-2.13	0.11	0.59	0.05	-0.02	0.49	-0.10	0.27	0.19
11	-13.31	-8.56	12.83	1.42	-0.57	2.81	1.97	0.29	0.16	0.15	0.14	-0.14
12	9.31	4.38	7.75	6.48	0.42	0.63	0.71	0.00	0.03	-0.78	0.30	0.18
13	-38.43	-3.81	11.05	-6.03	-0.57	1.97	-0.43	-0.22	0.14	-0.70	-0.01	0.06
14	-46.72	-4.95	4.58	-0.60	-0.53	0.63	-1.16	0.02	0.17	-0.10	0.18	-0.01
15	-20.86	-4.36	16.04	1.52	-1.03	3.01	1.68	-0.15	0.33	0.14	0.05	0.00
16	-34.09	11.34	11.34	-1.25	1.58	1.58	0.19	0.04	0.04	-0.70	-0.10	-0.10
17	-51.19	1.81	0.46	-9.00	1.04	0.20	1.61	-0.07	0.35	-0.39	0.16	0.08
18	-4.77	5.76	25.88	-6.81	1.59	3.87	2.75	-0.33	-0.05	0.03	0.39	0.21
19	-1.90	-16.08	7.69	0.08	0.60	2.48	0.86	-0.37	-0.08	0.05	0.45	-0.07
20	-40.08	3.89	3.89	-1.45	1.69	1.69	0.11	0.02	0.02	-0.17	0.44	0.44
21	-85.77	-13.25	-10.42	-13.70	-1.16	-0.33	0.16	-0.01	0.19	-0.72	0.42	0.29
22	-19.80	10.51	10.51	-3.45	0.74	0.74	1.20	-0.17	-0.17	-0.56	-0.28	-0.28
23	-37.47	-9.51	-5.52	-0.98	-0.68	0.40	-0.80	-0.48	-0.08	-1.04	0.05	0.06
24	-30.00	-3.56	6.06	-3.22	0.53	2.39	-0.40	-0.36	-0.02	0.65	0.09	-0.25
25	-34.59	-0.22	32.27	-3.94	-1.90	2.40	3.20	0.51	0.55	-0.22	0.57	0.13
26	-24.25	0.34	1.07	0.65	0.44	1.49	-1.00	-0.44	-0.08	-0.71	0.08	0.15
27	-27.26	4.86	9.60	-1.26	0.50	2.59	-0.22	-0.30	0.09	-0.54	0.45	0.45
28	-33.33	-13.82	-3.47	-5.97	-2.18	-1.08	1.60	-0.29	-0.12	-0.47	0.36	0.08
29	-41.72	-14.46	-8.96	-2.10	-0.96	-0.67	2.89	0.04	0.21	0.44	0.34	0.16
30	-24.14	-3.82	5.58	3.78	4.12	5.37	-0.34	-0.58	-0.17	-0.76	-0.04	0.08
31	-45.12	-2.56	-2.56	-8.22	-0.69	-0.69	0.03	-0.24	-0.24	-0.07	-0.12	-0.12
32	-32.33	-7.85	11.63	-1.83	-0.89	1.33	-0.29	-0.42	-0.21	-0.32	0.26	0.27
33	-18.99	7.41	3.71	0.81	1.16	0.77	1.17	-0.42	-0.06	-0.94	0.18	0.03
34	-43.05	-0.64	8.06	-10.77	-2.91	-2.90	1.53	-0.45	-0.31	0.14	0.33	0.09

Table 5: Comparison Table Part 1b

	Area Difference			Perimeter Difference			X-Centroid Difference			Y-Centroid Difference		
Building #	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized
35	-40.73	5.31	5.31	-5.38	1.97	1.97	0.26	-0.66	-0.66	0.08	0.52	0.52
36	-32.29	8.00	14.36	0.20	1.48	2.49	2.13	-0.33	-0.13	-0.96	0.47	0.37
37	-36.17	-6.25	0.98	-5.22	-1.08	0.32	-0.39	-0.25	-0.12	0.17	0.20	0.47
38	-38.36	14.79	20.61	-3.82	2.12	3.79	-0.99	-0.17	0.36	-0.12	0.07	0.14
39	-30.67	0.00	7.59	-0.58	0.05	1.50	-0.43	-0.77	-0.05	0.13	0.53	0.41
40	-23.75	4.10	6.13	0.33	0.69	0.39	0.38	-0.54	0.04	0.02	0.15	0.33
41	-15.98	2.53	7.75	1.98	-0.22	1.10	-0.17	-0.21	0.01	-0.65	-0.10	-0.01
42	-23.18	3.60	14.27	-2.08	0.23	1.88	-0.03	-0.61	-0.01	-0.43	-0.29	0.07
43	-27.94	-4.58	5.12	0.34	-0.15	1.23	1.76	-0.32	-0.06	-0.30	0.22	0.20
44	-40.28	7.93	7.93	-1.24	1.15	1.15	0.24	0.04	0.04	-0.32	0.22	0.22
45	-18.34	-10.67	-0.52	-2.35	-0.83	0.40	1.42	0.07	0.49	-0.22	0.00	0.02
46	-14.25	9.37	13.51	8.04	7.10	6.94	1.91	-0.12	0.30	-0.28	0.15	0.03
47	-31.75	-12.74	-1.06	0.14	-1.28	0.27	-0.53	-0.38	0.25	-0.45	0.00	0.07
48	-10.97	-3.53	-2.76	4.25	-0.42	-0.12	-0.02	-0.72	-0.44	-0.74	-0.04	0.03
49	-13.57	4.00	-2.69	1.93	0.34	-0.14	1.89	-0.72	-0.12	-0.13	0.15	0.21
50	-30.22	-1.42	16.42	-0.82	-0.70	2.18	1.27	-0.27	-0.22	-1.13	0.35	0.37
51	-62.16	-25.51	-8.54	-9.07	-3.33	-0.95	-0.99	-0.52	-0.10	-1.01	-0.14	0.12
52	-32.15	1.12	2.11	-0.15	0.42	0.58	-0.13	-0.37	0.01	-0.39	0.23	0.24
53	-22.12	14.73	21.61	1.12	2.63	3.96	-0.34	-0.82	-0.15	-0.22	0.34	0.38
54	-28.97	10.22	11.46	-0.76	1.62	1.96	0.04	-0.65	-0.19	-0.37	0.40	0.62
55	-30.01	6.11	8.80	1.14	0.87	1.12	-0.20	-0.64	-0.36	0.10	-0.20	-0.04
56	-25.96	-3.79	9.18	-1.51	-0.56	1.43	1.65	-0.11	0.00	-0.30	0.00	0.22
57	-33.93	-11.98	-2.08	-0.53	-0.44	0.02	-0.43	-0.58	-0.02	-0.45	0.04	-0.03
Min	-85.77	-25.51	-10.42	-13.70	-3.33	-2.90	-1.83	-0.82	-0.66	-1.59	-0.29	-0.28
Max	9.31	17.44	32.27	8.04	7.10	6.94	3.20	0.51	0.55	1.21	0.59	0.62
Avg	-31.62	-0.65	6.60	-1.54	0.36	1.38	0.42	-0.29	0.00	-0.35	0.18	0.14
Std Dev	17.09	8.42	8.33	4.03	1.61	1.58	1.12	0.27	0.23	0.50	0.21	0.20

Table 6: Comparison Table Part 2a

	Centroid Error			Area Error			Perimeter Error			Overlapped Area		
Building #	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized
1	0.55	0.25	0.10	-2%	2%	3%	2%	3%	3%	91%	95%	95%
2	1.97	0.38	0.16	-46%	0%	4%	-10%	1%	2%	91%	96%	94%
3	1.25	0.39	0.56	-30%	-2%	2%	-4%	0%	2%	97%	96%	93%
4	2.16	0.48	0.25	-10%	0%	4%	4%	1%	4%	84%	94%	94%
5	0.50	0.77	0.36	-22%	-1%	1%	1%	1%	2%	99%	95%	96%
6	0.87	0.16	0.22	-22%	1%	1%	-1%	0%	-1%	96%	97%	97%
7	1.41	0.60	0.40	-7%	-6%	-3%	8%	3%	4%	84%	95%	95%
8	0.72	0.42	0.35	-43%	0%	4%	0%	1%	3%	100%	94%	93%
9	1.74	0.48	0.17	-46%	8%	9%	-10%	3%	3%	96%	91%	91%
10	0.12	0.27	0.53	-22%	0%	3%	-4%	0%	1%	100%	97%	94%
11	1.98	0.32	0.21	-8%	-5%	6%	2%	-1%	5%	88%	98%	92%
12	1.06	0.30	0.19	5%	2%	4%	10%	1%	1%	85%	95%	95%
13	0.82	0.22	0.15	-23%	-2%	5%	-10%	-1%	3%	92%	98%	95%
14	1.17	0.18	0.17	-28%	-2%	2%	-1%	-1%	1%	99%	99%	96%
15	1.68	0.16	0.33	-12%	-2%	8%	2%	-2%	5%	93%	97%	90%
16	0.73	0.10	0.10	-19%	5%	5%	-2%	3%	3%	98%	94%	94%
17	1.66	0.17	0.36	-29%	1%	0%	-15%	2%	0%	99%	97%	97%
18	2.75	0.50	0.21	-2%	3%	12%	-12%	2%	6%	83%	94%	88%
19	0.87	0.58	0.10	-1%	-8%	3%	0%	1%	4%	92%	96%	93%
20	0.21	0.44	0.44	-24%	2%	2%	-2%	3%	3%	100%	93%	93%
21	0.74	0.42	0.35	-54%	-6%	-4%	-24%	-2%	0%	100%	98%	98%
22	1.32	0.33	0.33	-10%	5%	5%	-6%	1%	1%	92%	93%	93%
23	1.31	0.48	0.10	-22%	-5%	-3%	-2%	-1%	1%	94%	97%	96%
24	0.77	0.37	0.25	-17%	-2%	3%	-5%	1%	4%	98%	96%	96%
25	3.20	0.76	0.56	-23%	0%	15%	-7%	-3%	4%	78%	94%	83%
26	1.23	0.45	0.17	-13%	0%	1%	1%	1%	2%	94%	95%	93%
27	0.59	0.54	0.46	-15%	2%	4%	-2%	1%	4%	97%	93%	91%
28	1.67	0.46	0.14	-18%	-7%	-2%	-9%	-3%	-2%	96%	98%	99%
29	2.93	0.34	0.27	-25%	-7%	-4%	-4%	-2%	-1%	86%	98%	97%
30	0.84	0.58	0.19	-13%	-2%	3%	6%	6%	8%	93%	93%	91%
31	0.08	0.26	0.26	-25%	-1%	-1%	-14%	-1%	-1%	99%	98%	98%
32	0.44	0.49	0.34	-18%	-4%	5%	-3%	-2%	2%	99%	97%	94%
33	1.50	0.46	0.07	-11%	4%	2%	1%	2%	1%	88%	93%	95%
34	1.53	0.56	0.32	-24%	0%	4%	-18%	-4%	-4%	89%	90%	88%

Table 7: Comparison Table Part 2b

	Centroid Error			Area Error			Perimeter Error			Overlapped Area		
Building #	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized
35	0.27	0.84	0.84	-27%	3%	3%	-10%	3%	3%	94%	90%	90%
36	2.34	0.57	0.39	-23%	4%	8%	0%	3%	4%	85%	92%	90%
37	0.42	0.32	0.48	-21%	-3%	0%	-9%	-2%	0%	98%	98%	94%
38	1.00	0.19	0.39	-22%	6%	9%	-6%	3%	6%	98%	92%	89%
39	0.45	0.93	0.41	-16%	0%	3%	-1%	0%	2%	98%	93%	94%
40	0.38	0.56	0.33	-12%	2%	3%	1%	1%	1%	97%	95%	95%
41	0.67	0.23	0.01	-8%	1%	3%	3%	0%	2%	95%	95%	95%
42	0.43	0.68	0.07	-12%	2%	6%	-3%	0%	3%	98%	93%	94%
43	1.79	0.39	0.20	-15%	-2%	2%	1%	0%	2%	93%	95%	95%
44	0.40	0.23	0.23	-28%	4%	4%	-2%	2%	2%	100%	95%	95%
45	1.44	0.07	0.49	-6%	-4%	0%	-3%	-1%	1%	91%	98%	96%
46	1.94	0.20	0.31	-9%	5%	7%	13%	12%	11%	88%	89%	89%
47	0.69	0.38	0.26	-17%	-6%	-1%	0%	-2%	0%	98%	98%	97%
48	0.74	0.72	0.44	-5%	-1%	-1%	6%	-1%	0%	95%	95%	97%
49	1.89	0.73	0.24	-7%	2%	-1%	3%	1%	0%	82%	92%	98%
50	1.70	0.44	0.43	-19%	-1%	8%	-1%	-1%	4%	90%	95%	91%
51	1.41	0.54	0.15	-37%	-13%	-4%	-15%	-5%	-1%	99%	99%	98%
52	0.41	0.44	0.24	-19%	1%	1%	0%	1%	1%	100%	95%	95%
53	0.40	0.88	0.41	-13%	7%	10%	2%	4%	6%	99%	87%	89%
54	0.38	0.77	0.65	-17%	5%	5%	-1%	3%	3%	100%	90%	94%
55	0.23	0.67	0.36	-14%	2%	3%	2%	1%	2%	99%	93%	95%
56	1.68	0.11	0.22	-17%	-2%	5%	-3%	-1%	3%	92%	98%	94%
57	0.63	0.58	0.03	-19%	-6%	-1%	-1%	-1%	0%	98%	97%	99%
Min	0.08	0.07	0.01	-54%	-13%	-4%	-24%	-5%	-4%	78%	87%	83%
Max	3.20	0.93	0.84	5%	8%	15%	13%	12%	11%	100%	99%	99%
Avg	1.12	0.44	0.29	-19%	0%	3%	-3%	1%	2%	94%	95%	94%
Std Dev	0.73	0.21	0.16	11%	4%	4%	7%	3%	2%	6%	3%	3%

Table 8: Comparison Table Part 3a

	Commission			Omission			Percent Accuracy		
Building #	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized
1	9%	5%	5%	12%	3%	2%	81%	89%	89%
2	9%	4%	6%	55%	4%	2%	81%	91%	89%
3	3%	4%	7%	33%	5%	4%	95%	93%	87%
4	16%	6%	6%	26%	6%	2%	68%	88%	88%
5	1%	5%	4%	23%	6%	3%	97%	89%	93%
6	4%	3%	3%	26%	2%	2%	93%	93%	93%
7	16%	5%	5%	23%	12%	8%	69%	89%	89%
8	0%	6%	7%	43%	5%	4%	100%	88%	85%
9	4%	9%	9%	50%	1%	0%	93%	83%	83%
10	0%	3%	6%	22%	3%	3%	99%	94%	89%
11	12%	2%	8%	19%	7%	2%	77%	96%	84%
12	15%	5%	5%	11%	3%	1%	69%	90%	91%
13	8%	2%	5%	30%	4%	0%	85%	95%	89%
14	1%	1%	4%	29%	4%	1%	99%	97%	93%
15	7%	3%	10%	19%	5%	2%	85%	94%	81%
16	2%	6%	6%	21%	0%	0%	96%	89%	89%
17	1%	3%	3%	30%	3%	3%	98%	93%	94%
18	17%	6%	12%	19%	3%	0%	67%	87%	76%
19	8%	4%	7%	9%	12%	4%	85%	91%	86%
20	0%	7%	7%	24%	5%	5%	100%	87%	87%
21	0%	2%	2%	54%	8%	7%	99%	95%	96%
22	8%	7%	7%	19%	2%	2%	83%	87%	87%
23	6%	3%	4%	28%	7%	6%	88%	95%	93%
24	2%	4%	4%	19%	6%	1%	96%	92%	92%
25	22%	6%	17%	45%	6%	2%	55%	89%	66%
26	6%	5%	7%	20%	5%	6%	88%	90%	87%
27	3%	7%	9%	18%	5%	4%	94%	85%	83%
28	4%	2%	1%	22%	9%	2%	92%	96%	99%
29	14%	2%	3%	39%	9%	8%	72%	96%	93%
30	7%	7%	9%	20%	8%	6%	86%	87%	83%
31	1%	2%	2%	26%	3%	3%	99%	96%	96%
32	1%	3%	6%	20%	7%	1%	97%	94%	88%
33	12%	7%	5%	23%	3%	3%	76%	86%	91%
34	11%	10%	12%	35%	10%	8%	78%	80%	76%

Table 9: Comparison Table Part 3b

	Commission			Omission			Percent Accuracy		
Building #	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized	Structures	Imagery Digitized	DEM Digitized
35	6%	10%	10%	33%	8%	8%	88%	79%	79%
36	15%	8%	10%	39%	3%	2%	70%	84%	81%
37	2%	2%	6%	23%	5%	5%	96%	96%	89%
38	2%	8%	11%	24%	2%	2%	95%	84%	78%
39	2%	7%	6%	18%	7%	3%	96%	85%	88%
40	3%	5%	5%	15%	4%	2%	93%	89%	91%
41	5%	5%	5%	12%	4%	1%	91%	90%	91%
42	2%	7%	6%	14%	5%	0%	96%	86%	87%
43	7%	5%	5%	22%	7%	3%	86%	90%	89%
44	0%	5%	5%	28%	1%	1%	100%	90%	90%
45	9%	2%	4%	15%	6%	4%	83%	96%	92%
46	12%	11%	11%	21%	6%	4%	76%	79%	78%
47	2%	2%	3%	20%	8%	4%	96%	97%	93%
48	5%	5%	3%	10%	6%	4%	89%	90%	93%
49	18%	8%	2%	25%	6%	3%	65%	83%	96%
50	10%	5%	9%	29%	5%	1%	81%	91%	82%
51	1%	1%	2%	39%	13%	6%	97%	99%	97%
52	0%	5%	5%	19%	5%	4%	100%	89%	89%
53	1%	13%	11%	15%	6%	1%	98%	73%	77%
54	0%	10%	6%	17%	5%	1%	100%	80%	87%
55	1%	7%	5%	14%	5%	2%	99%	86%	89%
56	8%	2%	6%	25%	4%	1%	84%	97%	88%
57	2%	3%	1%	21%	9%	2%	97%	93%	98%
Min	0%	1%	1%	9%	0%	0%	55%	73%	66%
Max	22%	13%	17%	55%	13%	8%	100%	99%	99%
Avg	6%	5%	6%	25%	5%	3%	88%	90%	88%
Std Dev	6%	3%	3%	10%	3%	2%	11%	5%	6%

REFERENCES

Agouris, P., Schenk, A. F., & Stefanidis, A. (1989). Zero-crossings for edge detection. *ASPRS-ACSM Fall Convention*, 91–99.

Alharthy, A., & Bethel, J. (2002). Heuristic Filtering and 3D Feature Extraction From LIDAR Data. In *ISPRS Commission III, Symposium 2002* (Vol. XXXIV, pp. 29–34). Graz, Austria.

Awrangjeb, M., Ravanbakhsh, M., & Fraser, C. S. (2010). Automatic detection of residential buildings using LIDAR data and multispectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *65*(5), 457–467. http://doi.org/10.1016/j.isprsjprs.2010.06.001

Bourke, P. (1998, March). Determining whether or not a polygon (2D) has its vertices ordered clockwise or counterclockwise. Retrieved from http://debian.fmi.uni-sofia.bg/~sergei/cgsr/docs/clockwise.htm

Croitoru, A., & Doytsher, Y. (2004). RIGHT-ANGLE ROOFTOP POLYGON EXTRACTION IN REGULARISED URBAN AREAS: CUTTING THE CORNERS. *The Remote Sensing Photogrammetry Society*, (19), 311–341.

Demir, N., Poli, D., & Baltsavias, E. (2009). DETECTION OF BUILDINGS AT AIRPORT SITES USING IMAGES & LIDAR DATA AND A COMBINATION OF VARIOUS METHODS. In *Object Extraction for 3D City Models, Road Databases and Traffic Monitoring - Concepts, Algorithms and Evaluation* (Vol. XXXVIII, pp. 71–76). Paris, France.

Ekhtari, N., Valadan Zoej, M. J., Sahebi, M. R., & Mohammadzadeh, A. (2009). Automatic building extraction from LIDAR digital elevation models and WorldView imagery. *Journal of Applied Remote Sensing*, *3*(1), 033571–033571–12. http://doi.org/10.1117/1.3284718

Ekhtari, N., Zoej, M., & Mohammadzadeh, A. (2008). AUTOMATIC BUILDING DETECTION FROM LIDAR POINT CLOUD DATA. *International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences*, *XXXVII*, 473–477.

Elaksher, A., & Bethel, J. (2002). Reconstructing 3D Building Wireframes from Multiple Images. *International Archives of Photogrammetry and Remote Sensing*, *XXXIV*, 1682–1750.

Gonzalez, R., & Woods, R. (2008). *Digital Image Processing* (Third Edition). Upper Saddle River, NJ: Pearson Education, Inc.

Haithcoat, T. L., Song, W., & Hipple, J. D. (2001). Building footprint extraction and 3-D reconstruction from LIDAR data. In *Remote Sensing and Data Fusion over Urban Areas, IEEE/ISPRS Joint Workshop 2001* (pp. 74–78). Rome. http://doi.org/10.1109/DFUA.2001.985730

Heath, M., Sarkar, S., Sanocki, T., & Bowyer, K. (1998). Comparison of edge detectors - A methodology and initial study. *COMPUTER VISION AND IMAGE UNDERSTANDING*, *69*(1), 38–54.

Jensen, J. (2000). *Remote Sensing of the Environment: An Earth Resource Perspective* (2nd Edition). Upper Saddle River, NJ: Prentice Hall.

Juneja, M., & Sandhu, P. (2009). Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain. *International Journal of Computer Theory and Engineering*, *1*(5), 614–621.

Lee, D. H., Lee, K. M., & Lee, S. U. (2008). Fusion of Lidar and Imagery for Reliable Building Extraction. *Photogrammetric Engineering & Remote Sensing*, *74*(No. 2), 215–225.

Maini, R., & Aggarwal, H. (2009). Study and Comparison of Various Image Edge Detection Techniques. *International Journal of Image Processing*, *3*(1), 1–11.

Mareboyana, M., & Chi, P. (2006). Parallel algorithm for linear feature detection from airborne LiDAR data (Vol. 6209, p. 62090I–62090I–6). http://doi.org/10.1117/12.668782

McLean, G. F., & Jernigan, M. E. (1988). Hierarchical edge detection. *Comput. Vision Graph. Image Process.*, *44*(3), 350–366. http://doi.org/10.1016/0734-189X(88)90129-6

Meng, X., Currit, N., & Wang, L. (2006). MORPHOLOGY-BASED BUILDING DETE CTION FROM AIRB ORNE LIDAR DATA. In *ASPRS 2008 Annual Conference*. Portland, Oregon.

Miliaresis, G., & Kokkas, N. (2007). Segmentation and object-based classification for the extraction of the building class from LIDAR DEMs. *Computers & Geosciences*, *33*(8), 1076–1087. http://doi.org/doi: 10.1016/j.cageo.2006.11.012

Morgan, M., & Tempfli, K. (2000). AUTOMATIC BUILDING EXTRACTION FROM AIRBORNE LASER SCANNING DATA. In *XIXth ISPRS Congress Technical Commission III: Systems for Data Processing, Analysis and Representation* (Vol. XXXIII, Part B3, pp. 616–623). aMSTERDAM.

National Oceanic and Atmospheric Administration (NOAA) Costal Services Center. (2012). Lidar 101: An Introduction to Lidar Technology, Data, and Applications. NOAA Coastal Services Center. Retrieved from http://coast.noaa.gov/digitalcoast/_/pdf/lidar101.pdf

Renslow, M., Greenfield, P., & Guay, T. (2012, November). Evaluation of Multi-Return LIDAR for Forestry Applications. Remote Sensing Applications Center. Retrieved from http://www.ndep.gov/USDAFS_LIDAR.pdf

Sampath, A., & Shan, J. (2010). Segmentation and Reconstruction of Polyhedral Building Roofs From Aerial Lidar Point Clouds. *IEEE Transactions on Geoscience and Remote Sensing*, *48*(3), 1554–1567. http://doi.org/10.1109/TGRS.2009.2030180

San, D. K., & Turker, M. (2010). BUILDING EXTRACTION FROM HIGH RESOLUTION SATELLITE IMAGES USING HOUGH TRANSFORM. *International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences*, *XXXVIII*(Part 8), 1063–1068.

Schuckman, K., & Renslow, M. (2014). Characteristics of Lidar Data. The Pennsylvania State University. Retrieved from https://www.e-education.psu.edu/geog481/l1_p6.html

Sithole, G., & Vosselman, G. (2004). Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *59*(1–2), 85–101. http://doi.org/10.1016/j.isprsjprs.2004.05.004

Architecture 2030. (n.d.). The Solution. Retrieved from http://architecture2030.org/files/2010_handout.pdf

Tupin, F., & Roux, M. (2003). Detection of building outlines based on the fusion of SAR and optical features. *ISPRS Journal of Photogrammetry and Remote Sensing*, *58*(1-2), 71–82. http://doi.org/10.1016/S0924-2716(03)00018-2

Turker, M., & Cetinkaya, B. (2005). Automatic detection of earthquake-damaged buildings using DEMs created from pre- and post-earthquake stereo aerial photographs. *International Journal of Remote Sensing*, *26*(4), 823–832. http://doi.org/10.1080/01431160512331316810

Verma, V., Kumar, R., & Hsu, S. (2006). 3D Building Detection and Modeling from Aerial LIDAR Data. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vol. 2, pp. 2213–2220). http://doi.org/10.1109/CVPR.2006.12

Vu, T., Matsuoka, M., & Yamazaki, F. (2005). Towards an Object-Based Detection of Earthquake Damaged Buildings. *International Society for Photogrammetry and Remote Sensing*, 1–5.

Wang, Z., Li, H.-Y., & Wu, L.-X. (2010). QEM-based simplification of building footprints from Airborne LiDAR data. *Geoscience and Remote Sensing Symposium*, 1186–1189. http://doi.org/10.1109/IGARSS.2010.5654122

Weng, S., Zhao, G., & He, B. (2010). Rapid reconstruction of 3D building models from aerial images and LiDAR data. In *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)* (Vol. 2, pp. V2–195–V2–198). http://doi.org/10.1109/ICACTE.2010.5579254

Xu, J., Wan, Y., & Zhang, X. (2006). A method of edge detection based on improved canny algorithm for the lidar depth image. In *Geoinformatics 2006: Remotely Sensed Data and Information* (Vol. 6419, p. 64190O–64190O–9). http://doi.org/10.1117/12.712923

Yu, H., Zhang, Y., Cheng, G., & Ge, X. (2011). Rural residential building extraction from laser scanning data and aerophotograph based on quadtree segmentation. In *2011 International Conference on Remote Sensing, Environment and Transportation Engineering (RSETE)* (pp. 8476–8479). http://doi.org/10.1109/RSETE.2011.5964137

Yu, Y., Liu, X., & Buckles, B. (2010). A Cue Line based method for Building Modeling from LiDAR and Satellite imagery. *Computing Communications and Networking Technologies*.

Zabuawala, S., Nguyen, H., Wei, H., & Yadegar, J. (2009). Fusion of LIDAR and aerial imagery for accurate building footprint extraction (Vol. 7251, p. 72510Z–72510Z–11). http://doi.org/10.1117/12.806141

Zhang, K., Yan, J., & Chen, S.-C. (2006). Automatic Construction of Building Footprints From Airborne LIDAR Data. *IEEE Transactions on Geoscience and Remote Sensing*, *44*(9), 2523–2533. http://doi.org/10.1109/TGRS.2006.874137

Zhao, T., & Wang, J. (2014). Use of lidar-derived NDTI and intensity for rule-based object-oriented extraction of building footprints. *International Journal of Remote Sensing*, *35*(2), 578–597. http://doi.org/10.1080/01431161.2013.871394

Zhou, G., Song, C., Simmers, J., & Cheng, P. (2004). Urban 3D GIS From LiDAR and digital aerial images. *Computers & Geosciences*, *30*(4), 345–353. http://doi.org/10.1016/j.cageo.2003.08.012

BIOGRAPHY

Justin Miller was born in Alexandria, VA and raised in Culpeper, VA by his loving and supportive parents Jan and Bill along with his brother Jason. After graduating from Culpeper County High School in 2004 he was accepted to Shenandoah University in Winchester, VA. There he graduated with a bachelors in Mathematics and a minor in Physics and Chemistry in 2008. Currently Justin is a candidate for a Master of Science degree in Geoinformatics and Geospatial Intelligence at George Mason University

From 2008 to 2011, Justin was employed by The Boeing Company supporting the National Geospatial-Intelligence Agency conducting photogrammetric feature and terrain extraction. From 2011 to 2013, he was employed by General Dynamics and Companion Security Group (CSG) supporting the Army Geospatial Center performing quality control and assurance on high resolution ortho-mosaiced imagery and Light Detection and Ranging (LiDAR) products. Since 2013 he has been employed by the National Geospatial-Intelligence Agency as a Photogrammetric Image Scientist and plans on advancing his career in remote sensing and imagery sciences.

Justin was married on June 20, 2009 to his beautiful wife Jenn. Jenn and Justin welcomed their delightful son Camden on January 4, 2014. Justin and Jenn plan to spend many summers on the beach and on the water at his parents' house on the Potomac River and enjoy watching their son grow and flourish.