The Thesis Committee for Harish Sangireddy
Certifies that this is the approved version of the following thesis:

Point Cloud Classification for Water Surface Identification in Lidar Datasets

APPROVED BY
SUPERVISING COMMITTEE:

Co-Supervisors:

David R. Maidment

Paola Passalacqua

Point Cloud Classification for Water Surface Identification in Lidar Datasets

by

Harish Sangireddy, B.Tech.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2011

Acknowledgements

I would like to thank Dr. David Maidment, my advisor and Dr Paola Passalacqua, my co-supervisor for their support, guidance and new visions. I would also like to thank Center for Research in Water Resources (CRWR) for funding and supporting my master's research. And finally thanks to my Parents and my friends for their belief and motivation.

Abstract

Point Cloud Classification for Water Surface Identification in Lidar Datasets

Harish Sangireddy, M.S.E

The University of Texas at Austin, 2011

Supervisor: David R. Maidment & Paola Passalacqua

Light Detection and Ranging (Lidar) is a remote sensing technique that provides high resolution range measurements between the laser scanner and Earth's topography. These range measurements are mapped as 3D point cloud with high accuracy (< 0.1 meters). Depending on the geometry of the illuminated surfaces on earth one or more backscattered echoes are recorded for every pulse emitted by the laser scanner. Lidar has the advantage of being able to create elevation surfaces in 3D, while also having information about the intensity of the returned pulse at each point, thus it can be treated as a spatial and as a spectral data system.

The 3D elevation attributes of Lidar data are used in this study to identify possible water surface points quickly and efficiently. The approach incorporates the use of

Laplacian curvature computed via wavelets where the wavelets are the first and second order derivatives of a Gaussian kernel.

In computer science, a kd-tree is a space-partitioning data structure used for organizing points in a k dimensional space. The 3D point cloud is segmented by using a kd-tree and following this segmentation the neighborhood of each point is identified and Laplacian curvature is computed at each point record. A combination of positive curvature values and elevation measures is used to determine the threshold for identifying possible water surface points in the point cloud. The efficiency and accurate localization of the extracted water surface points are demonstrated by using the Lidar data for Williamson County in Texas. Six different test sites are identified and the results are compared against high resolution imagery. The resulting point features mapped accurately on streams and other water surfaces in the test sites. The combination of curvature and elevation filtering allowed the procedure to omit roads and bridges in the test sites and only identify points that belonged to streams, small ponds and floodplains. This procedure shows the capability of Lidar data for water surface mapping thus providing valuable datasets for a number of applications in geomorphology, hydrology and hydraulics.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 MOTIVATION

Lidar (Light Detection and Ranging) is changing the way the world is measured, captured and mapped. The technology is based on a laser emitting light pulses hitting the ground and reflecting back to the sensor which enables the distances to various objects to be measured accurately, based upon the speed of light. The pulses which are captured as returns by the sensor are able to map features like vegetation, buildings, power lines and the earth's surface. It is like mapping the earth's surface from the sky.

Flood monitoring and flood protection is one of the most important issues of watershed management. Floodplain mapping and risk maps for floods are important for disaster management and in formulating policies. A lot of effort and energy has been spent on creating floodplain maps by using data collected at various scales. Federal Emergency Management Agency (FEMA) requires elevation data of 2 feet accuracy in flat areas and 4 feet accuracy in rolling and hill areas for floodplain mapping (NRC 2007). These standards correspond to a root mean square error in elevation of 0.61 to 1.22 feet respectively. The National Elevation Dataset which is derived from contour information in USGS 1:24,000 scale topographic maps has a root mean square error of 7.68 feet. This means that the current flood plain maps are based upon a very old National Elevation Database (NRC 2007). The National Research Council's committee on Floodplain Mapping Technologies (NRC, 2007) concluded that new elevation data should be collected and should employ Lidar as the primary technology for digital elevation data acquisition and the FEMA floodplain maps should be updated with consistent elevation data. Lidar data is being collected throughout the United States of

America and other countries (60% of UK is covered by Lidar now), for accurate mapping and creation of better digital elevation models.

The elevation data for the earth's surface can be divided into a tiered data structure where the data sources are arranged by scale as shown in the Figure1. Accurate terrain data is required to understand and analyze various hydraulic and hydrological processes at micro terrain level.

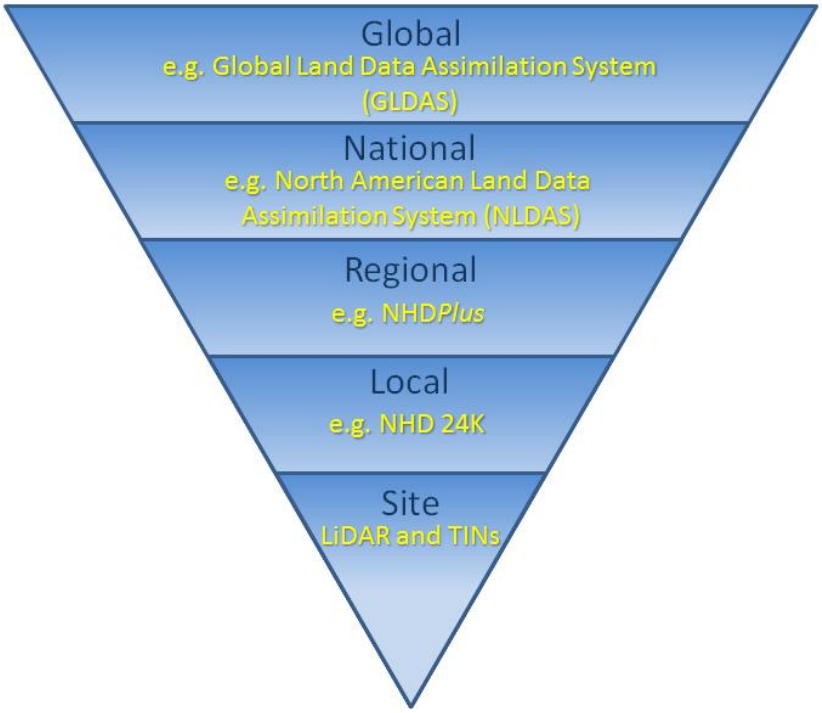

Figure 1: Tiered data sources by scale

Lidar high resolution data is collected and delivered as point cloud to the end customer. The point cloud data is a collection of the X, Y, Z coordinates and other Lidar attributes such as intensity and return number for each feature occurring in the area over which Lidar data is collected. The area and slope of channels is an important parameter in modeling hydraulic processes. Much work has been done on extracting streams and

channels using Lidar data from digital elevation models (Passalacqua, et al. 2010). Conventional digital elevation models and terrain analysis are based upon creating gridded datasets e.g. raster surfaces. Each pixel value is geo-referenced and contains the elevation value. These raster datasets are created by using points that are classified as bare earth points in the point cloud.

Feature identification, and classification of point cloud is an important step before we create high resolution surface models. It is necessary to identify bare earth points and differentiate them from points that belong to other features. Approaches such as object oriented classification of point cloud by using intensity and elevation values have been used in the past to classify point cloud (Antonarakis, Richards and Brasington 2008) mainly in regions dominated by thick vegetation and forested areas. This kind of classification has been also used to identify features such as roads (Clode, et al. 2007) , buildings (Sampath and Shan 2007) and power lines.

Identifying streams, floodplains and water surface points is a very challenging and difficult task. The errors associated with intensity and elevation measurements over water surface have kept researchers searching for better methods. Often, Lidar pre-processing methods filters out such points from the point cloud by either using breaklines or imagery data. These water gaps in the Lidar datasets are easily seen over larger water bodies since most streams and drains are dry for a major part of the year, measurements of Lidar intensity and elevation over such terrain are not removed during the pre-processing steps of the Lidar data and are classified as bare earth points.

For flood mapping and floodplain delineation it is important to identify the complete flow network of streams and rivers. Less research has been done in identifying points lying on streams and other water surfaces. These reasons motivated this research

towards finding methods to identify and classify points in the point cloud as water points and thus enable us to extract streams, water bodies and floodplains.

1.2 Objectives and Chapter Outline

The objective of this research is to identify streams, extract channels and floodplains from Lidar point cloud datasets. The research aims at

- identifying methods to classify points,
- create data structures for reading Lidar files and
- segmenting the points into features of interest

Since the primary goal of the research is identifying water bodies from Lidar datasets, methods concerning hydrological feature classification and segmentation are studied.

The thesis is divided into five chapters. The first chapter describes research and the objectives of the study. Chapter two reviews' related literature and research work done in the past. This chapter also explains the technology that drives Lidar and key concepts that are necessary for understanding the science of Lidar world. Chapter three outlines the methodologies used for classifying the data and segmenting the point cloud dataset. The standards followed for sharing Lidar files and the various attributes that can be useful in extracting the stream segments are analyzed. Chapter four describes a case study for test sites in Williamson County in Texas and talks about the model algorithm and feature extraction methods. And finally chapters five presents the results and conclusions.

Chapter 2: Literature and Technology Review

A brief review of related literature and research studies carried out by various researchers in the past is presented below.

2.1 LIDAR SYSTEMS

On a functional level Lidar integrates three different systems into a single system capable of acquiring data to produce accurate digital terrain models (DTM). These technologies are lasers, the global positioning system (GPS), and inertial navigation systems (INS). Combined together they support determining the footprint of the Lidar pulse as it hits the objects on ground with high degree of accuracy (Ambercore Terrapoint 2008).

Lidar technology has advanced significantly in the last four- five years. The ranging accuracy has improved substantially, now it is at the level of 1 – 2 cm point spacing for hard surfaces (Toth 2010). Besides obtaining elevation data, the interpretation of returned signal is now feasible with the appearance of full-waveform Lidar systems. Point density has improved as the pulse repetition frequency (PRF) advanced to the 100 – 200 KHz range. The introduction of multiple pulses in air (MPiA) technology, which allows the firing of a second laser pulse by the sensor before the reflected signal from the previous pulse has been received, boosts the PRF further till 400 KHz (the commercial Optech Pegazus system is a 400 KHz multi-channel system) (Toth 2010) .

A Lidar system consists of single narrow-beam laser with a laser pulse receiver system. The receiver receives the laser pulse and it measures the travel time of the pulse from the start to its return. Since the pulses are travelling at the speed of light the receiver is able to measure a pulse of light before it sends out another pulse. Knowing the speed of

light and the time of travel we can accurately detect the distance to the object from which the pulse was reflected back.

Combining the laser range, laser scan angle, laser position from GPS, and laser orientation from INS, accurate X, Y and Z ground coordinates can be determined for each pulse. Large volumes of data are collected since the number of pulses emitted per second can be anywhere between few hundreds to tens of thousands. In modern Lidar systems the typical pulse width is in the range of 4-10 ns, equivalent to 1.2-3.0 m distances. The great majority of airborne Lidar systems work with 1.064μ wavelengths. The separation of returns is practically defined by the emitted laser pulse width; for a 10ns pulse, the object surfaces should be at least 3 m apart to produce separate pulses (Toth 2010).

The ability to acquire multiple returns while flying over the land helps in capturing the bare earth points and also the top of leaves, buildings and various other features that could possibly be present on the surface of earth as shown in Figure 2.

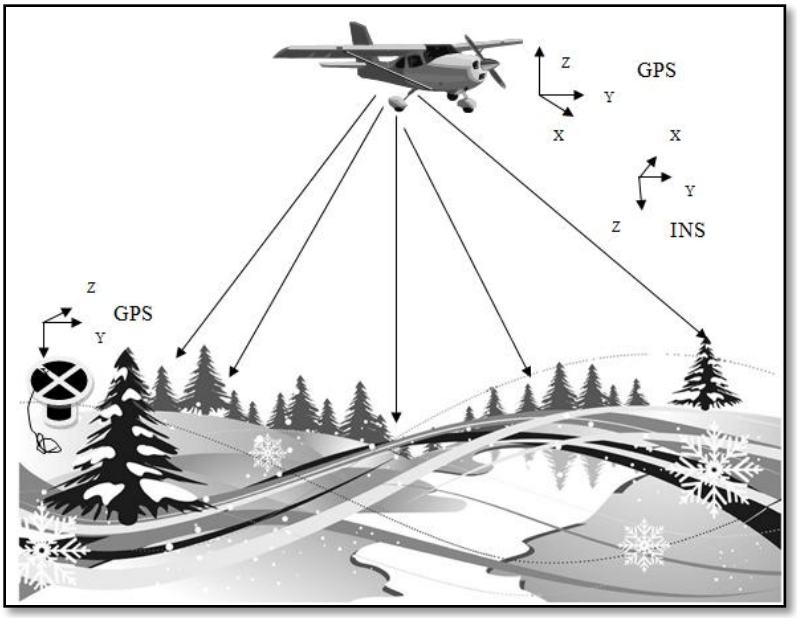

Figure 2: Lidar system

The resulting product is a densely spaced network of highly accurate georeferenced elevation points as shown in Figure 3 called point cloud that can be used to generate 3d surface representations of Earth's surface and its features.

Figure 3: Lidar point cloud

A discussion of Lidar often includes reference of technical terms that describe the level of accuracy, data collection and ensuing processing steps. (Schmid, Waters and Dingerson 2008)

- **RMSE –** The root mean square error, a measure of the accuracy of data and a check on the bias present in the data collected.
- **Fundamental Vertical Accuracy (FVA) –** a measure of the accuracy of the data in open areas at a high confidence level (95%). Calculated from RMSE using the formula

$$\text{RMSE} * 1.96 = \text{FVA} \quad (1)$$

- **Classification –** data that has been processed to define the type of object that the pulses have reflected off; can be as simple as unclassified (i.e. Point not defined) to buildings and high vegetation. The most common is to classify the data set for points that are termed as "bare earth" points while the rest are left unclassified.
- **Return Number (First/Last return) –** many Lidar systems are capable of capturing the first, second, third and ultimately the last returns from the objects on earth's surface. The return number can be used to classify the point correctly.
- **Point Spacing –** how close the laser points are to each other, analogous to the pixel size of aerial image, also called point density.
- **Intensity Density –** when laser return is recorded, the strength of the return is also tracked and this tells us how well the object on the ground reflected back the pulse of light. The intensity density is expressed in range of 0-100%.
- **DEM (Digital Elevation Model) -** the surface created from the point cloud elevation data to represent the 3d earth surface.

2.2 HYDROGRAPHIC LIDAR

Airborne Laser hydrography systems are different than normal terrain Lidar systems. They are able to determine water depths by measuring the time of flight of two laser pulses of different wavelengths (Populus 2009). Of the two pulses, an infrared wavelength is scattered back from the surface of water and the second light pulse typically a green light (wavelength of 532 nm) travels through the water interface and hits

the bottom. A sensor on the aircraft detects the time elapsed between receiving both the wave forms and hence it can determine the depth of water after accounting for the system's geometry, propagation induced biases, wave height and tide effects.

Some of the disadvantages of Hydrographic Lidar systems are they are only capable of measuring depths till 40-50 meters in very clear oligotrophic (environments that offer little to sustain life) water. Heavy bottom vegetation and fluid mud reduces the quality of Hydrographic Lidar systems. They are typically not employed in regions of chronic turbidity. The depth of water measured is usually about 3 times the Secchi depth (Smith, Irish and Smith 2000). Hence Hydrographic Lidar must generally collect data at lower altitudes than topographic Lidar systems. It is a unique capability of Hydrographic Lidar systems to survey shallow coastal areas. Most Hydrographic Lidar systems are government owned like the NASA EAARLS experimental Lidar system and the USACE/USN/NOAA CHARTS Lidar system, which have the capability to gather topographic and hydrographic data along shorelines (Heslin, Lillycrop and Pope 2003).

2.3 ASPRS STANDARDS

The American Society for Photogrammetry and Remote Sensing (ASPRS) is a scientific association with over 7000 professional members around the world advancing the knowledge of mapping sciences to promote responsible applications of Photogrammetry, remote sensing and geographical information systems (GIS). As a contributing organization to the Federal Geographic Data Committee (FGDC) ASPRS recommends a common Lidar Exchange Format Standard. The Lidar file format is a public binary file format for interchange of Lidar information between vendors and customers. This eliminates the use of proprietary systems for exchange of Lidar data such

as ASCII file interchange. The ASCII file format degrades processing performance and the file size is extremely large even for small amounts of data. Another problem is that all the raw data and information specific to the Lidar system is lost while attempting to keep the ASCII file size small, thus inhibiting troubleshooting and debugging of problem datasets and limit third party data analysis. The binary Lidar file format maintains all information specific to the Lidar systems while not being overly complex.

Files conforming to the ASPRS Lidar data exchange format standard are named with a LAS extension and are referred as LAS files. The LAS file contains Lidar point records which are a combination of GPS, INS and laser pulse range data as X, Y and Z coordinates. The format contains binary data consisting of a header block, variable length records and point data records. All data is in little-endian format (ASPRS Standards Committee for Lidar 2003). Over the years the ASPRS has revised the LAS format multiple times to accommodate the changes in Lidar systems as well as to enhance the performance of the binary structure proposed to store Lidar information. The first ASPRS version of LAS 1.0 was released in May 2003. The subsequent versions, LAS 1.1, LAS 1.2 and the current version LAS 1.3 were released subsequently by the ASPRS Standards committee. Since the data available for the study presented in this thesis came from a LAS 1.0 version file format, the sections below talk about items found in version 1.0 only. Detailed information about all LAS versions can be accessed on ASPRS website http://www.asprs.org/society/committees/standards/lidar_exchange_format.html.

2.3.1 Public Header Block

The information contained in the header block can be divided into various items; each item has a specific size and format. Not all the items are required, some are optional items. The list of items found in the header block is shown in Table 1.

Item	Format	Size	Required
File Signature	Char[4]	4 bytes	*
Reserved	Unsigned long	4 bytes	
GUID data 1	Unsigned long	4 bytes	
GUID data 2	Unsigned short	2 bytes	
GUID data 3	Unsigned short	2 bytes	
GUID data 4	Unsigned char[8]	8 bytes	
Version Major	Unsigned char	1 byte	*
Version Minor	Unsigned char	1 byte	*
System Identifier	Char[32]	32 bytes	*
Generating software	Char[32]	32 bytes	*
Flight Date Julian	Unsigned short	2 bytes	
Year	Unsigned short	2 bytes	
Header size	Unsigned short	2 bytes	*
Offset to data	Unsigned long	4 bytes	*
Number of Variable length records	Unsigned long	4 bytes	*
Point data format ID (0-99 for spec)	Unsigned char	1 byte	*
Point data record length	Unsigned short	2 bytes	*
Number of point records	Unsigned long	4 bytes	*
Number of points by return	Unsigned long [5]	20 bytes	*
X scale factor	Double	8 bytes	*
Y scale factor	Double	8 bytes	*

Table 1 : Header Block of a LAS file

Item	Format	Size	Required
Z scale factor	Double	8 bytes	*
X offset	Double	8 bytes	*
Y offset	Double	8 bytes	*
Z offset	Double	8 bytes	*
Max X	Double	8 bytes	*
Min X	Double	8 bytes	*
Max Y	Double	8 bytes	*
Min Y	Double	8 bytes	*
Max Z	Double	8 bytes	*
Min Z	Double	8 bytes	*

Table 2: Header Block of LAS file Continued

All the data that is not required and not filled with data must be set to zero. The file signature field must contain the characters LASF. A detailed explanation of each field can be found in ASPRS LAS 1.0 format standard (ASPRS Standards Committee for Lidar 2003).

2.3.2 Variable Length Records

The variable length records include variable type definition about projection information, metadata and user application data. The items in the variable records are shown in Table 3. Detailed information about the fields can be found in ASPRS LAS 1.0 format standard (ASPRS Standards Committee for Lidar 2003).

Item	Format	Size	Required
Record signature	Unsigned short	2 bytes	*
User ID	Char[16]	16 bytes	*
Record ID	Unsigned short	2 bytes	*
Record length after Header	Unsigned short	2 bytes	*
Description	Char[32]	32 bytes	

Table 3: Variable length records of a LAS file

2.3.3 Point Data Records

The point data record is the most important part of the LAS file. It contains the information collected at each point by the laser pulse. The original point data record format is 0 and all other formats are extended or derived from the "Point Data 0" structure with additional data added thereafter. The Point Data 0 format items are listed in Table 4.

Data from all the three parts of the LAS file are required to obtain the coordinates of each point. The offset fields in the header block are used to set offset values for the point records. So to get the X coordinate from the point record the X from the point record portion is multiplied by the X scale factor and then added to the X offset in the header block.

Thus we can derive the coordinates of each point as shown below.

$$X_{coordinate} = (X_{record} * X_{scale}) + X_{offset} \quad (2)$$

$$Y_{coordinate} = (Y_{record} * Y_{scale}) + Y_{offset} \quad (3)$$

$$Z_{coordinate} = (Z_{record} * Z_{scale}) + Z_{offset} \quad (4)$$

Item	Format	Size	Required
X	Long	4 bytes	*
Y	Long	4 bytes	*
Z	Long	4 bytes	*
Intensity	Unsigned short	2 bytes	
Return Number	3 bits	3 bits	*
Number of returns (given pulse)	3 bits	3 bits	*
Scan Direction Flag	1 bit	1 bit	*
Edge of Flight Line	1 bit	1 bit	*
Classification	Unsigned char	1 byte	
Scan Angle Rank (-90 to +90) –left side	Unsigned char	1 byte	*
File Marker	Unsigned char	1 byte	
User Bit Field	Unsigned short	2 bytes	
GPS Time	Double	8 bytes	*

Table 4: Point Data Records of a LAS file

Other attributes such as intensity, scan angle and return number can also be read from the point data record for each point.

2.4 LIDAR DATA PROCESSING METHODS

Lidar data collection results in large volumes of data. Processing of data involves reading the binary files, arranging them in defined data structures and processing them to derive meaningful information. This requires a lot of internal and external computer memory and processing power. A typical Lidar file (**.las*) may have nearly 10 million

points with an average point spacing of 1.2 meters. The Lidar tiles cover nearly an area of 16 Km square on earth surface and range from 25 to 180 MB in size.

A review of Lidar file naming convention followed in the United States of America along with various processing methods available for reading and analyzing these massive datasets is now presented.

2.4.1 Lidar File Naming Format

Typically the naming format for a Lidar file conforms to 1/16th USGS 7.5 minute quadrangle (1.875 minute by 1.875 minute) tiles. A USGS 7.5' X 7.5' Quadrangle and Quarter-Quarter Quad example is shown in Figure 4. For instance let's consider the file name "360125_3_C", this refers to Lidar file covering the quad C in 3rd quarter of the 7.5 min quarter of the USGS Quarter-Quarter-quad. Special suffix and prefix may be used to identify the Lidar files from a specific vendor, as well as to identify the type of data being delivered e.g. "fr" for first return, "lr" for last return, "la" for all laser points in LAS files and so on.

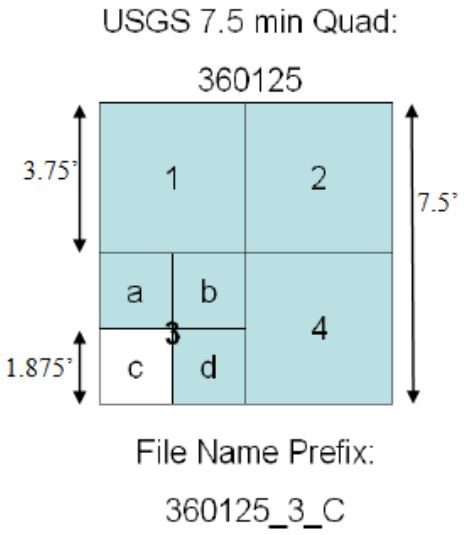

Figure 4: USGS 7.5 min Quad file naming format (TNRIS 2009)

An illustration of the Quarter-Quarter Quad file naming format is done in a later chapter for the LAS files used in the thesis. After successfully identifying the LAS files to be utilized for a study, the processing of each file can be accomplished via a batch process or an out of core process (by creating a tree data structure).

2.4.2 Batch Processing

Working with Lidar data requires large virtual memory. The Lidar data is typically stored and arranged in tiles. During a batch processing operation, the files are read one after the other into the memory, processed, the results are written to the disk and then the memory is cleaned. This operation is continued until all the files are read and the results are stored on disk. This method is very convenient for custom software and is often implemented by commercial software products. Scripting and batch execution scripts can be implemented for a huge number of Lidar files.

However batch processing may not be preferred solution. While sorting the Lidar datasets for global thresholds and for nearest neighbor implementation in point cloud or for feature extraction and boundary extraction, access to the full data is required which may not be possible during a batch process operation.

2.4.3 Out of Core Processing

Out of core algorithms operate on large volumes of data. These algorithms differ from batch processing methods in that they may or may not read the data in sequential order and the same data may be read and written more than once to complete the

operation. Since these processes require a lot of I/O operations, the algorithms are designed to minimize the I/O.

These algorithms typically consist of creating tree data structures that arrange the data spatially by branching them into nodes and subdividing them into groups that resemble tiles of batch processing. Tree based data structures have many applications in internal spatial indexing methods (Hanan 1990). These data structures are compact and depending on the nature of data, they save space as well as time and facilitate operations such as search.

2.4.4 Lidar Pre-Processing

While commercial Lidar has come a long way, processing of huge point clouds for the purpose of modeling systematic errors, filtering, feature extraction and thinning often requires a large amount of human intervention. For production of digital elevation models, the manual classification (filtering) and quality control pose the greatest challenges, consuming an estimated 60-80% of processing time (Flood 2001) .

Numerous algorithms have been developed for semi automatically extracting bare earth from point clouds obtained by Lidar systems. Important considerations when selecting a ground filter include the number and type of Lidar returns to use for ground filtering, input data format and definition of neighbors. Various ground filters have been devised to identify bare earth points in the literature. A class and methods table is prepared in Table 5 and Table 6 that illustrates the class of ground filters and the key methods used for filtering purpose by various authors in the past.

Class of ground filters	Key Methods	Authors
Segmentation/cluster	Segmentation based on smoothness constraint	(Rabbania, Heuvelb and Vosselmanc 2006)
	Segmentation based classification	(Filin 2002)
	Segment- based terrain interpolation	(Tóvári and Pfeifer 2005)
Morphology	Dual rank filter based dilation and erosion	(Lohmann, Koch and Schaeffer 2000)
	A morphological filter based on geodesic dilation	(Arefi and Hahn 2005)
	Progressive morphological filter	(Zhang, et al. 2003)
Directional Scanning	Bidirectional scanning	(Shan and Sampath 2005)
	Hybrid multi directional ground filtering	(Meng, Wang and Currit 2009)
Contour	Active contour and active shape model	(Elmqvist, et al. 2001)
	Active shape model based on energy function	(M. Elmqvist 2002)
TIN	Local curvatures of point measurements	(Haugerud and Harding 2001)
	The adaptive TIN model	(Axelsson 2000)

Table 5: Ground filtering methods and classes

Class of ground filters	Key Methods	Authors
Interpolation	The iterative robust implementation	(Briese and Pfeifer 2001)
	The multi-scale curvature algorithm based on TPS interpolation	(Evans and Hudak 2007)
	A facet model	(Zheng, et al. 2007)
	Linear prediction	(Lee and Younan 2003) (Pfeifer, et al. 1999)

Table 6: Ground filtering methods and classes continued

Short descriptions of the above mentioned filter types are included below:

Segmentation and clustering based filtering are the most popular techniques for land use and land cover classification. Many segmentation and clustering filtering techniques have been carried out on flat terrain and there is a need to evaluate these filters on rough terrain. **Mathematical morphology** deals with the shape measurement and experiments on Lidar data show that morphological filters have the ability to remove non ground objects such as buildings and trees. Typical processes include opening, closing, dilation and erosion based on kernel operators (Xuelian, Nate and Kaiguang 2010). Both classes of filters have been studied and used extensively in the literature.

Most ground filters define neighborhood in a two dimensional space but few methods calculate slope and elevation difference along a one dimensional scan line. Shan and Sampath (Shan and Sampath 2005) developed a **bi-directional algorithm** to remove non-ground points based on slope and elevation difference calculated along the scan line with additional reference to the nearest ground neighbor point and the label of the

previous point along the scan line. Instead of referring to estimated values based on surrounding points this bi-directional method compares the points to their immediate neighbors and has been proven to be efficient in removing low vegetation.

Contour based and TIN based filters are relatively new methods used in Lidar terrain preprocessing and they show good performance when compared to other methods. **Interpolation** based filters are based on the principle of comparing the elevation of points with the estimated values that are obtained through various interpolation techniques such as the linear least square interpolation technique. The Thin plate spline (TPS) model is also another interpolation technique considered to be very efficient in removing the vegetation and identifying bare earth points.

All of the methods specified above have been used to identify bare earth points, by filtering vegetation, buildings and roads. Most of the methods do not address the identification of hydrological features and hence these points either end up in the datasets as bare earth points or they remain unclassified.

2.5 LIDAR DATA MODELING

After pre-processing, the Lidar data is ready for post-processing to derive final products such as DEMs, DTMs and TINs for further analysis of data. With the help of Lidar data, 3D surface models can be easily generated by using visualization tools and methodologies. Tools like ArcView, 3d Analyst and QCoherent's LP360 extension for ArcGIS can be used to achieve this task. For generating 3D models, aerial photography, imagery and existing maps are required for a high level of accuracy. The methods of post-processing Lidar data are numerous and differ from each other depending on the final product required.

2.5.1 TINs

One of the most important aspects of post processing Lidar data is to identify topologically significant points to create a continuous terrain. A Triangulated Irregular Network (TIN) has been used a great deal in geographic information systems to represent terrain (Chen and Guevara 1987) (Vivoni, et al. 2004). The points in the point cloud are chosen as the nodes for the TINs and the Delaunay triangulation scheme is followed to construct triangles on these nodes. Construction of TINs on bare earth points creates a 3D earth surface that can be viewed as terrain. When the number of points is huge and the pre-processing is not accurate, Chen's Very Important Point method can be used to re sample points of interest from the point cloud. (Vivoni, et al. 2004) also considered using a hydrological similarity index to identify points of interest after creating DEMs from the Lidar data to create floodplains.

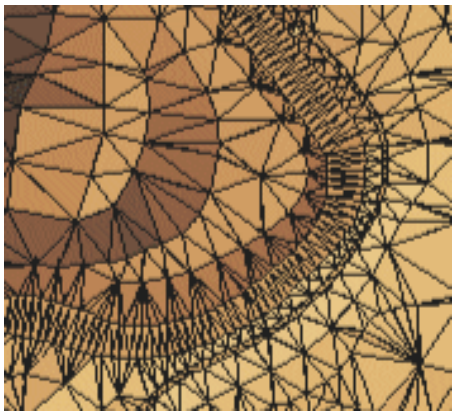

Figure 5: Lidar data viewed as TINs

TINs preserve the significant points on terrain and thus help in creating better terrain models. They are 3D in nature and do not suffer from interpolation issues like raster datasets. Uneven point density or water gaps could be some issues that might produce undesirable results. Inaccurate pre-processing might leave traces of trees,

building edges and power lines thus producing uneven terrain while triangulating on the point cloud.

2.5.2 Breaklines

Lidar pre-processing is a very important process. The extraction of bare earth points is the desired result. Comparing the Lidar bare earth points with field surveys is an important validation and quality control step. Questions like: does the Lidar and derived contours meet mapping applications, is the Lidar data on ground, are answered during the validation and quality control steps.

Breaklines are collected by the vendor to indicate various features like roads, boundaries of buildings and water bodies, streams and small ponds. These break lines are inserted into the terrain created by constructing TINs. This not only helps in identifying streams and boundaries, but it also helps in validations such as the difference in elevation obtained from Lidar and field surveys. The gap between Lidar derived bare earth points and field survey points defines the error in the terrain surface and aids in validation of the derived terrain surface.

2.5.3 Point cloud Classification

Since the pre-processing methods do not offer a perfect classification process, most third party Lidar point cloud visualization software, allows for classification of points by overlaying the point cloud and derived terrain model on base maps and high resolution imagery. Thus the process of point cloud classification can be carried out as a part of the post-processing of Lidar datasets.

To date most land cover classification methods work on raster datasets and intensity models derived from the Lidar point cloud. Recent studies show how object based point cloud analysis (Rutzinger, et al. 2008) following the concept of object based

image analysis can improve the classification process. This process offers a more robust laser point classification methodology with higher success for natural objects and surfaces than classification that does not consider information on geometry and attributes of larger spatial units. Points are segmented by using various techniques based on height, estimated normals to the point, curvature and slope or gradient differences within small neighborhoods.

These segments are classified based on their contextual information. A point is considered as the smallest unit of process. An Individual point is identified by defining its neighborhood relations e.g. height or slope differences between points. However for neighborhoods containing objects and bare-earth points, it is difficult to distinguish and classify the points on a single neighborhood (Sithole and Vosselman 2005), (Tóvári and Pfeifer 2005), (Brovelli, Cannata and Longoni 2004).

The review of literature related to Lidar technology shows that most of the research concentrates on identifying bare earth points and then creating a mesh surface to visualize these point cloud datasets. The literature also cites work in the field of forest inventory where Lidar data has been used for calculation of stem volume, mean or dominant height of canopy and leaf area (Säynäjoki, Maltamo and Korhonen 2008). Lidar has also been used for coastal studies such as extraction of shore lines (Mohammadzadeh and Zoej 2008) and determining mean high water level (MHWL) which is treated as the legal shoreline by many U.S. governmental agencies , including the US Army Corps of Engineers, FEMA and US Census Bureau (Graham, Sault and Bailey 2003) (Pajak and Leatherman 2002).

Chapter 3: Methodology

The objective of the research work is identifying water surfaces by using geometry and intensity attributes from the point cloud directly. Water surface mapping is the part of remote sensing techniques which aim at assessing the geometrical representation of water areas. Utilizing Lidar directly delivers geometrical information of the scanned surface by combining the GPS and INS resulting in a coordinate tuple (X, Y, and Z) for each recorded reflection. To date, only a few studies such as (Hofle, et al. 2009) and (French 2003)have investigated the potential of Lidar datasets to classify water points directly from the point cloud.

Lidar datasets not only give us information about the X, Y and Z of a point but they also have information about intensity, scan angle and return number for each pulse. Studies have been conducted to classify and segment point cloud datasets by using other non geometrical attributes of Lidar data. Lutz et al. 2003 concluded that the intensity values of a glacial lake are generally low when compared to land surface. Similarly it has been observed in the works of (Brennan and Webster 2006), (Antonarakis, Richards and Brasington 2008) and (Brzank, et al. 2008) that the intensity values observed over water surface are relatively lower than those of surrounding land surfaces. They all characterized water areas with very low intensity and low normalized height (i.e. DTM subtracted from DSM). The above approaches provided classification accuracy in the order of 90% to 100% when compared against reference plots which were selected partly in homogeneous areas.

Lidar intensities of water surfaces are low due to strong absorption of near-infrared light by water. Hence, in this standing, emphasis was laid upon extracting and

identifying potential water points by using geometrical analysis. Moreover the methods followed in the pre-processing of the LAS files used for the study were also unknown.

The methodology consists of three steps; first build the data structure to read the LAS files. Second, create local neighborhoods in the point cloud dataset and finally analyze these neighborhoods to identify possible water surface points.

3.1 READING LAS FILES

Acquiring, sorting and reading LAS file is tedious. Software like ArcMap, QCoherent and a few others provide some form of interface to sort and read LAS file information. The following sections describe about how LAS files can be read by using ArcGIS and also independently by using python outside ArGIS.

3.1.1 Reading Las Files in ArcGIS

ArcMap has offers a set of tools to read and store LAS file information in a geodatabase.

3.1.1.1 Point file Information toolbox

All Lidar files are referenced by a USGS quarter-quarter quad schema. The point file Information tool box generates a new output feature class containing statistical information about one or more LAS files. Figure 6 shows that each file is represented by a tile with attributes being file name, the point count, the average point spacing, Z min and Z max. The advantage of having a point file information feature is to select the LAS file to work with and just load the particular file. It also helps to select LAS files geographically which would not have been possible if it were to select the file by looking at the file name only.

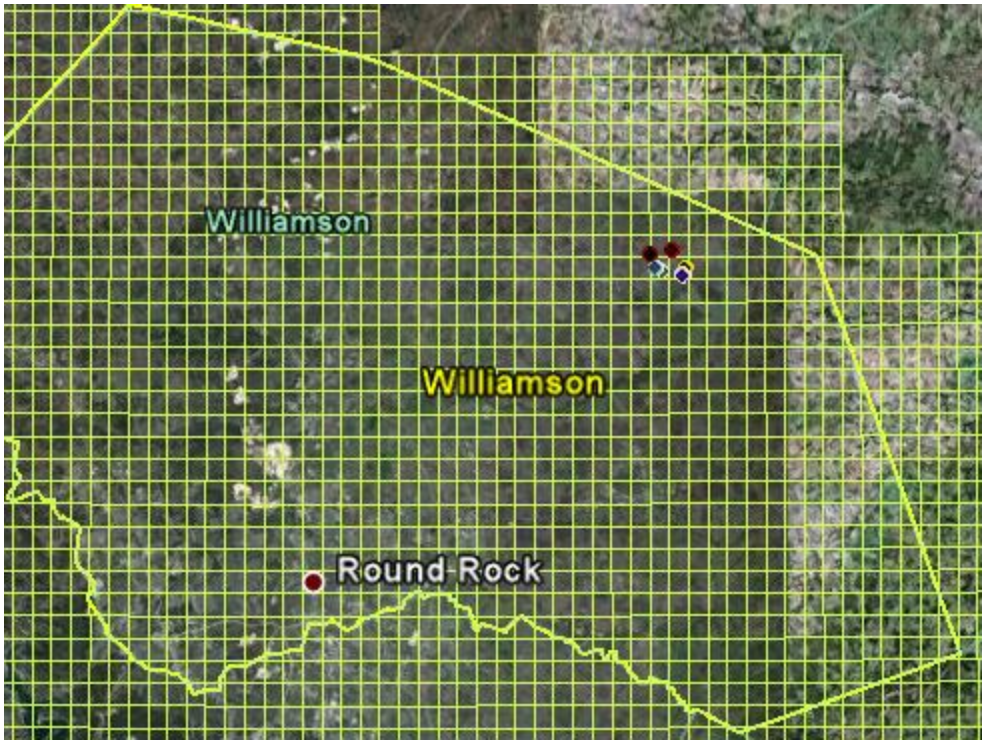

Figure 6: Point file information feature class for Williamson County, Texas in ArcGIS

The attributes like the Z max and Z min is read from the header block in the LAS file, where the average point spacing reported is not exact, it is an estimate that the tool makes by comparing the area of the file's bounding box with the total point count. The point file information toolbox can also be used to make feature class from a collection of point data files stored in a folder.

3.1.1.2 Las to Multipoint toolbox

Multipoint objects are collections of points. Features that are composed of more than a single point are called Multipoint. Multipoint objects are often used to manage arrays of very large point collections, such as Lidar point clusters, which can contain billions of points. Using a single row entry for such point geometry is not feasible (ESRI 2010). After creating the point file information feature class, the desired LAS file is

selected and the LAS to Multipoint toolbox is used to create a multipoint feature class from the LAS file. The toolbox supports reading of LAS versions 1.0, 1.1 and 1.2. It reads other Lidar attributes such as intensity and return number to form blob fields.

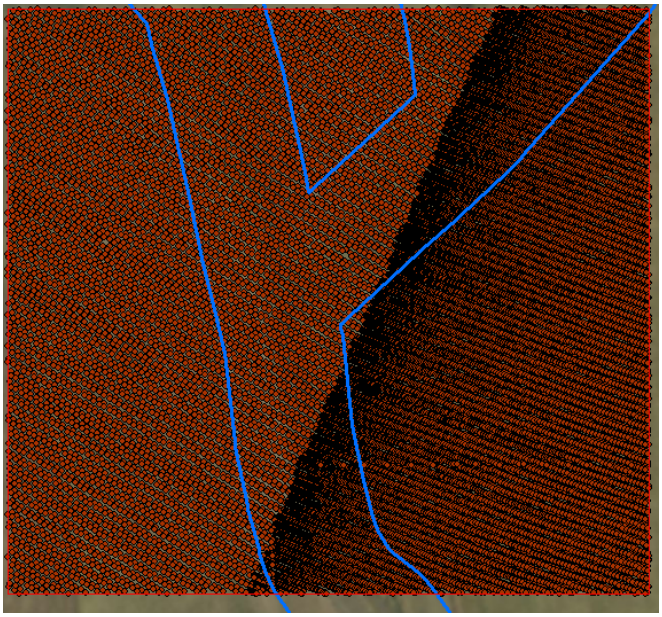

Figure 7: A multipoint feature class derived from Lidar data in ArcGIS

3.1.1.3 Terrain and TINs

Terrains and TINs are two possible ways of creating a continuous surface from multipoint feature classes. They help in determining patterns that are not readily apparent in the original multipoint features, such as slope and aspect.

Terrain datasets are TIN based multi-resolution surfaces made from elevation data stored in feature classes. Terrains reside inside a feature dataset and extend its capability to add and remove terrain points. Terrain datasets can be stored inside a geodatabase.

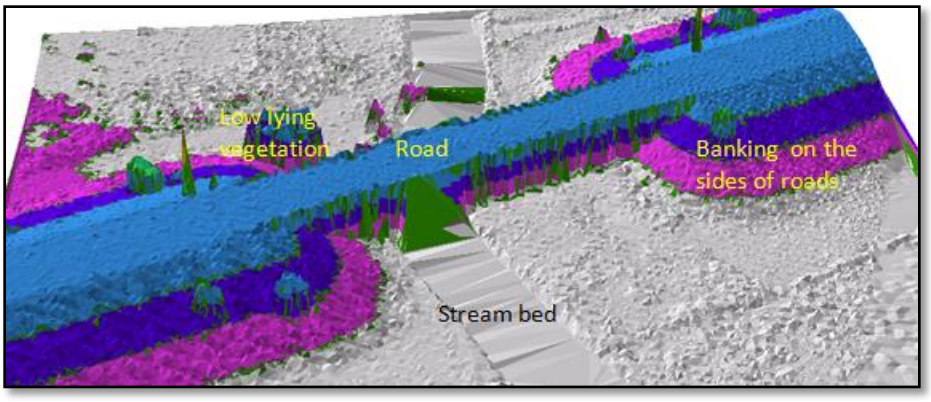

Figure 8: Terrain surface in ArcGIS

TINs are constructed by triangulating set of vertices; the vertices are connected with a series of edges to form a network of triangles. The resulting triangulation satisfies Delaunay triangulation which ensures that no vertex lies within the interior of any of the circum-circle of the triangles in network.

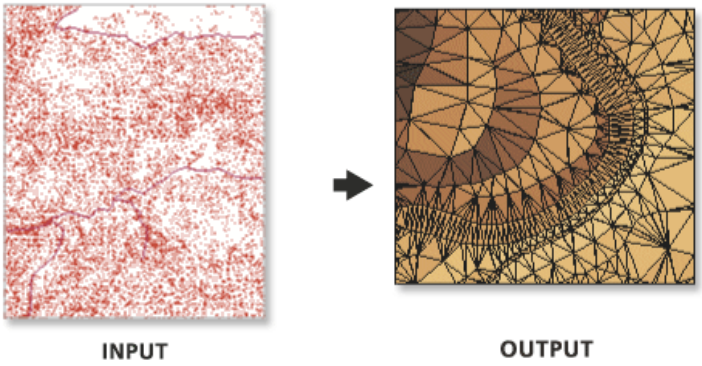

Figure 9: TIN surface created from a point cloud data

These methods are efficient tools to visualize the point cloud datasets in GIS environment. They rely on point cloud datasets that are classified into bare-earth and non bare-earth points. It is difficult to obtain perfectly classified datasets and differentiating between different features occurring on earth's surface is essential. Detailed information about the terrain is crucial for 2D hydrodynamic models. Identifying riverbed and dry

area of the river valley will be useful in facilitating simulation of river valley during the analysis of water flow.

3.1.2 Reading Las files using Python

Even though ArcGIS is able to read and visualize LAS files, the ability to read only a selected set of points based on intensity and elevation values is not available in ArcGIS. A LAS file is a binary format file defined by ASPRS structure explained in Chapter 2. Hence, any programming language that has ability to read binary files can be used for this purpose. The programming language Python was chosen because of its ability to interact with various ArcGIS functions and tools that could be used for other processing steps. Also the availability of a wide variety of functions within Python makes the analysis of LAS data very easy and efficient.

Two different data structures are defined in the python script to process the header block and the point record information. The data structure consists of the fields that were described in Chapter 2. First, the file header is read and the information about the number of point records, the start and end position of the point records, the maximum and minimum Z values, the X, Y, and Z offset values, X, Y and Z scale values are stored in memory. Once this information is read, the point record is read and the X, Y and Z coordinate of each point is built. The intensity values and the return number are also captured for each point record and the data is stored in arrays. Depending upon the minimum and maximum Z values the data is treated for elevation filtering.

3.1.2.1 Intensity Filtering

Initial filtering based on intensity helps in reducing the amount of data to be analyzed and hence improving the performance. Python enables us to enforce such filtering while reading data directly from the LAS files. As each point record is read from

the LAS file the intensity attribute is checked for the enforced filtering criteria and only points that pass the check are stored in arrays. Intensity filtering is not used for this study.

3.1.2.2 Elevation Filtering

While reading the LAS file, the header block can be used to identify the Z max and Z min in the LAS file. It could be possible that the Lidar file might contain points belonging to the cloud and the bare-earth. The points belonging to cloud (above the earth) could exist because of pre-processing errors and they may represent trees, light poles and buildings. These kinds of features are not of interest in the study and they are certainly above a threshold level which can be determined upon initial investigation of the LAS files. Elevation filtering can thus be used to remove such cloud data.

Upon elevation filtering of the dataset, the point cloud is spatially indexed by creating a tree data structure. The out-of core memory process is used here for holding and sorting the point cloud and finally searching for the neighborhood of each point.

3.2 CREATING POINT NEIGHBORHOOD

To create the neighborhood of each point we use the neighbors within a K radius. Space partitioning algorithms such as the kd-tree (K-dimensional tree) have good performance in organizing and retrieving neighborhood for point cloud.

3.2.1 KD-Tree Creation

A kd-tree is a data structure for sorting a set of points from a K dimensional space. It was initially proposed by (Bentley 1980) . A kd-tree is a special kind of binary tree. A tree data structure consists of a set of linked nodes. Mathematically a tree is a connected graph where each node has zero or more children nodes and at most one parent node. The contents and algorithm of the kd-tree are described briefly in the following sections. Table 7 describes the attributes stored at each node.

Field Name	Field Type	Description
Dom-elt	domain vector	A point from K_d-d space
Range-elt	Range vector	A point from K_r -d space
Split	Integer	The splitting dimension
Left	kd-tree	A kd-tree representing those points to the left of the splitting plane
Right	Kd-tree	A kd-tree representing those points to the right of the splitting plane

Table 7: The fields of the kd-tree

The *Dom-elt* field is the domain vector of the set of nodes in the tree. The *Range-elt* is the range vector of the nodes of the tree. The *Dom-elt* component is the index for the node and it splits the space into two subspaces according to the splitting hyperplane of the node. All the points in the subspace to the left are represented by the *left* subtree and points in the right subspace are represented by the *right* subtree. The splitting hyperplane is passes through the *Dom-elt* and is perpendicular to the direction specified by the *split* field. A kd-tree can be constructed by the algorithm described in Table 8 (Moore 1991)

Algorithm: Constructing kd-tree
Input: exset = The point cloud
Output: kd , of type kd-tree
Pre: Read the LAS file and create an array to hold X, Y, Z values after elevation and intensity filtering.
Code: 1. If point cloud is empty then return the empty kd-tree 2. **ex** := a member of the point cloud 3. **split** := the splitting dimension 4. **d** := domain of the **ex** 5. **exset'** := **exset** with **ex** removed 6. **r** := range vector of **ex** 7. **exsetleft** = $\{(d',r') \epsilon \text{ exset'} \mid d'_{split} =< d_{split} \}$ 8. **exesetright** = $\{(d',r') \epsilon \text{ exset'} \mid d'_{split} > d_{split} \}$ 9. **kdleft** := recursively construct kd-tree from the **exsetleft** **10. kdright** := recursively construct kd-tree from the **exsetright** 11. **kd** := **< d,r,split,kdleft,kdright >**

Table 8: Algorithm for constructing kd-tree

The point cloud is read as the input to the script that would create the kd-tree for a 3 dimensional dataset. The 3d array consisting of X, Y and Z are sorted and the point cloud is broken into two halves at the center having same number of points in each half. The center point is chosen as the parent node and the points whose X, Y and Z values are less than this root value are added to the left side of the tree and all other points whose X,

Y and Z values are greater than the root are added to the right side of the tree. The Tree index is maintained at each node. Subsequently, at each node, the points are sorted and the center point is chosen as the node value. All points in this branch of tree are compared with the node value at the current index and then arranged into left and right branches subsequently. This process is continued until a user specified leaf size is achieved. Once the leaf size is less than or equal to the specified size the branching of tree is stopped and the tree is created. A typical kd-tree structure is shown in Figure 10.

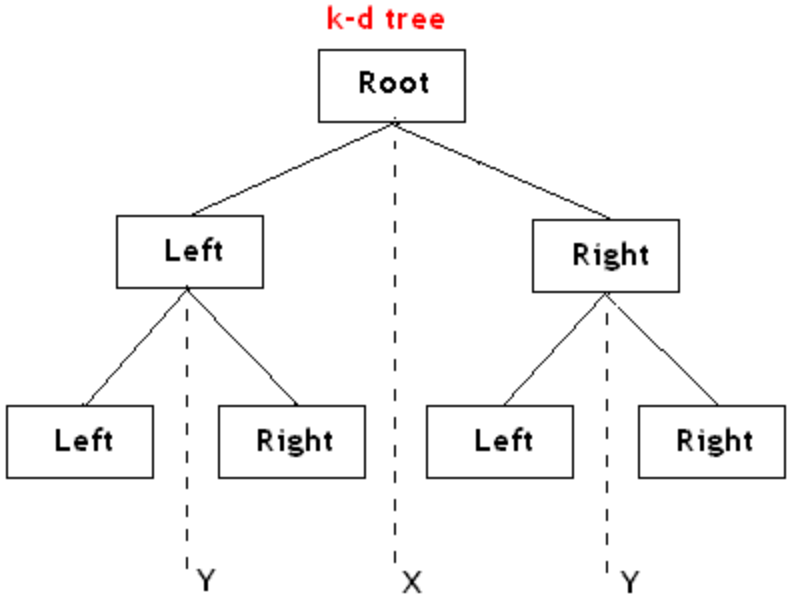

Figure 10: A typical kd-tree structure

The purpose of creating the kd-tree is not only to hold the point dataset in memory efficiently but also to enable easy and quick finding of the nearest neighbors.

3.2.2 Finding K nearest Neighbors

The nearest neighborhood algorithm aims to find the nearest point in the tree to the given point. The searching for the nearest neighbor proceeds as follows in a kd-tree.

1. Starting with the root node, the control moves down the tree recursively, i.e. it goes to the right or the left depending on whether the 2D distance (3D can also be used) of point is greater or less than a current node in the specified split dimension.
2. Once control reaches the leaf node, it saves the current node point as the current best fit.
3. The control unwinds the recursion of the tree, performing the following steps at each node:
    a. If the current node is closer than the current best, then it becomes the current best fit.
    b. The control checks whether there could be any points on the other side of the splitting plane that are closer to the search point than the current best. This is done by intersecting the splitting hyperplane with a hypersphere around the search point that has a radius equal to the nearest distance. This is implemented as a simple comparison to see whether the difference between the splitting coordinate of the search point and current node is less than the distance from the search point to the current best.
        i. If the hypersphere crosses the plane, there could be nearer points on the other side, so the control must move down the other branch of the tree from the current location or node, searching for closer points in space by following the same recursive technique.
        ii. If the hypersphere does not intersect the splitting plane, then the control moves up the tree and the entire branch on the other side is not inspected any more.

4. Once the process is finished for the root node the search returns the index and the distances of the points from the point of interest.

3.2.3 Finding Neighbors within K radius

The search is similar to finding the K nearest neighbors; the only difference being the Euclidean distance from the potential points is calculated at each step and compared if the distance is less than K.

3.3 NEIGHBORHOOD ANALYSIS

Having the neighborhood for each point within a K radius, the neighborhood is analyzed to decide if the sample point belongs to water surface or not. To this end a variety of neighborhood analysis methods are reviewed below, and the Gaussian curvature estimate is used to identify the possible water surface points.

3.3.1 Roughness

The roughness of the water surface is defined by the standard deviation of height values derived in a local neighborhood for each laser echo , water surfaces are assumed to be locally horizontal (Hofle, et al. 2009). The graph in Figure 11 from Hofle et.al 2009 illustrates how the roughness of the water surface is locally horizontal. They used a threshold of 0.2m to identify water surfaces for the River Inn near Innsbruck in Austria.

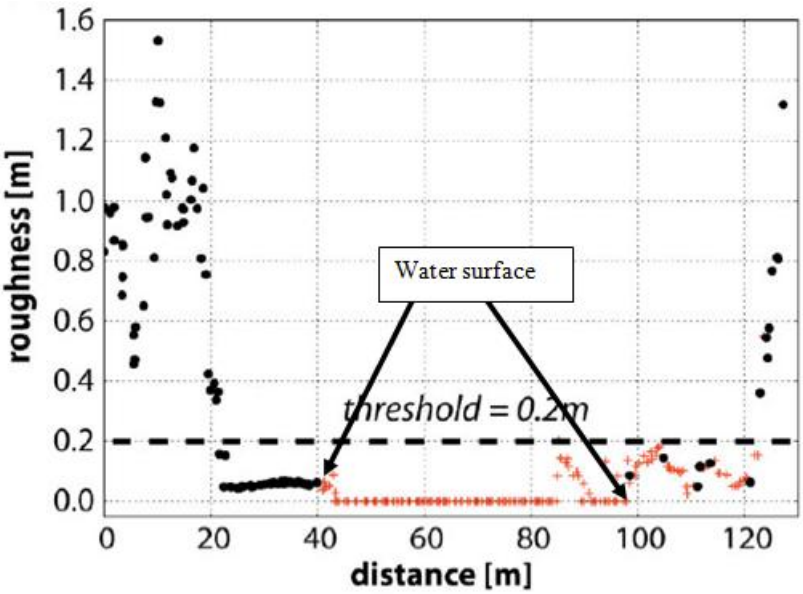

Figure 11: Cross-sections (2 m width) of Airborne Laser Swath (ALS) point attributes for River Inn: Roughness Hofle et al. 2009

3.3.2 Intensity Variation

The coefficient of variation of intensity in the neighborhood is called Intensity variation. When looking at neighborhood of point cloud it is not clear if the neighborhood contains a combination of different feature classes. Since the intensity of possible water points is different from bare earth points, it turns out that the intensity variation between land and water point clouds is high (Hofle, et al. 2009) .

The ground conditions during the time of collection of Lidar data, could lead to undesirable values of intensity. If the streams are dry during the time of collection of Lidar data, it could happen that the points which lie on the streams could be of the same intensity as those of the surrounding land surface points. It has been also observed in this study that the coefficient of variation of intensity of water surface points lies in the same range as that of roads and bridges. Hence the geometry of the point records is used in this

study to estimate curvature at each point weighted by the neighborhood and thus to help in determining if each point is a possible water surface point.

3.3.3 Curvature Estimation

In conventional terrain analysis using DEMs, three scalars are ordinarily computed at each site to characterize the topography of the surface: the local slope, the curvature of the topographic field of elevations and the total contributing area at each site. Mathematically slope of a line or surface describes the steepness, incline or grade. A higher slope value indicates a steeper incline.

$$slope = \frac{rise}{run} \qquad (5)$$

The curvature of a topographic field is defined by which a geometric object deviates from being flat. Curvature of a surface at a point generally varies with orientation and many definitions are possible (Schmidt, Evans and Brinkmann 2003). The total contributing area is a field representing drainage area upslope of each point. It is defined by tracing flow paths upslope from each point along the flow direction field to the drainage divide and measuring the area enclosed. Within a grid based GIS environment, contributing area is also referred to as the catchment area or the flow accumulation area.

Recently (Passalacqua, et al. 2010) proposed an approach which incorporates nonlinear diffusion for the processing of data to remove unwanted details and enhance the features that are relevant to the channel network extraction. They used a Perona-Malik filter (a nonlinear anisotropic diffusion filter) to reduce noise without affecting the boundaries localization (Perona and Malik 1990). After filtering the noise and enhancing the features they used energy minimization and geodesics to extract the channel.

Curvatures critically distinguish landscape sites. The curvature at the *i*th site, is defined by the second order derivative of a rectangular lattice involving all the eight nearest-neighbors. If the curvature is negative the site is topographically convex or topographically divergent and is conventionally defined as a ridge. On the contrary if the curvature is positive, the topography is concave and convergent typically defined as a valley (Tucker, et al. 2001).

There are multiple existing techniques for feature extraction from point clouds. Most of them have been developed in the field of robotics and computer vision and rely on polygon meshes, (Hubeli and Gross 2007), (Hildebrand, Polithier and Wardetzky 2005) created from a point cloud. A problem of the point based methods is the lack of any normal and connectivity information within the point cloud. Very few methods are dedicated to the point sampled geometry only. (Gumhold, Wang and McLeod 2001) presented a method that used the Riemannian tree to build the connectivity information in the point cloud. They then analyzed the neighborhood of each point via a principal component analysis method and the Eigen-values of the correlation matrix were used to determine a probability of a point belonging to a class of features. Also (Mrigot, Ovsjanikov and Guibas 2009) estimated the principal curvatures and normal directions of the underlying surface from a point cloud by using a so called Voronoi covariance measure and provided valuable theoretical measures.

(Schmidt, Evans and Brinkmann 2003) reviewed a set of curvature measurement methods independent and dependent of slope. (Lashermes, Foufoula and Dietrich 2007) showed the use of wavelets to compute local gradients and curvatures and slope direction change at multiple scales on digital elevation models.

A wavelet approach is implemented and tested in this study with an exception that the method was tested on point cloud dataset directly, instead of DEMs as suggested by (Lashermes, Foufoula and Dietrich 2007).

3.3.3.1 Gaussian filtering

The Gaussian kernel is named after Carl Friedrich Gauss, a brilliant German mathematician. The Gaussian kernel can be defined in 1-D, 2-D and N-D respectively as

$$G_{1D}(x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \qquad (6)$$

$$G_{2D}(x,y;\sigma) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (7)$$

$$G_{ND}(\vec{X};\sigma) = \frac{1}{(\sqrt{2\pi}\sigma)^N} e^{-\frac{|x|^2}{2\sigma^2}} \qquad (8)$$

The σ (standard deviation) determines the width of the kernel and signifies the scale in spatial dimensions. The term $\frac{1}{\sigma\sqrt{2\pi}}$ in front of the kernel is the normalization constant, i.e. its integration over its full domain is unity for every σ.

In electronics and signal processing, a Gaussian filter is a filter which is designed to give no overshoot to a step function input while minimizing the rise and fall time of a signal. Mathematically a Gaussian filter modifies the input signal by convolution with a Gaussian function.

3.3.3.2 Convolution

Mathematically the convolution is a mathematical operator on two functions f and g, producing a third function that is a modified version of the two input functions. The convolution of f and g is written as

$$(\mathrm{h} * \mathrm{g})(\mathrm{t}) = \int_{-\infty}^{\infty} \mathrm{h}(\tau)\mathrm{g}(\mathrm{t}-\tau)\mathrm{d}\tau \qquad (9).$$

One of the well known properties of convolution product is that

$$\frac{\partial}{\partial x}(h * g) = \frac{\partial h}{\partial x} * g = h * \frac{\partial g}{\partial x} \qquad (10)$$

If the function h is elevation heights h(x, y) and the function g the Gaussian kernel, then by taking the advantage of the last term in equation 10 we can define the curvature at a point in x direction by

$$\nabla^2_{x,\sigma} h(x,y) = \left(h * g^x_{2,\sigma,x,y}\right)(x,y) \qquad (11)$$

where $g^x_{2,\sigma,x,y}$, is the second derivative of a 2D Gaussian function with standard deviation σ and centered at location (x, y):

$$G^x{}_{2,\sigma,x,y} = \frac{\partial^2}{\partial x} G_\sigma(x,y) = \frac{1}{\sqrt{2\pi}\sigma} \frac{x^2 - \sigma^2}{\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \qquad (12)$$

$$G^y{}_{2,\sigma,x,y} = \frac{\partial^2}{\partial y} G_\sigma(x,y) = \frac{1}{\sqrt{2\pi}\sigma} \frac{y^2 - \sigma^2}{\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \qquad (13)$$

$$G^{x,y}{}_{2,\sigma,x,y} = \frac{1}{\sqrt{2\pi}\sigma} \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \qquad (14)$$

Thus we can find the gradient and curvature of the sample points by convolving the elevation at each point in the neighborhood with the first and second derivative of the Gaussian kernel respectively.

3.3.3.3 Quantile-Quantile Plot

The statistical signature of the transitions from hill slope to valley based on the curvature estimate can be seen by plotting the quantile-quantile plot of the curvature values for the complete dataset. In a quantile-quantile plot, the curvature values are plotted against the standardized normal deviate of the same exceedance probability. The

deviation from the straight line indicates the deviation from the Gaussian probability density function. After computing the quantile-quantile plots for the curvature values of all the points the presence of positive curvature values suggests the presence of valley regions and we can determine the threshold of transition from hill slope to valley.

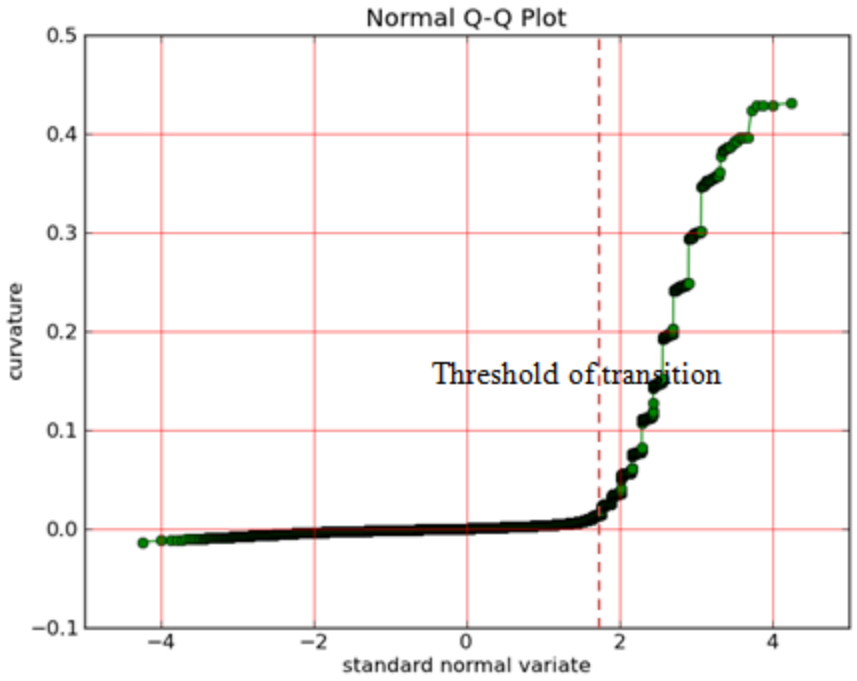

Figure 12: Quantile-Quantile plot for curvature

The complete end to end workflow of the developed methodology for water surface identification is shown in Figure 13.

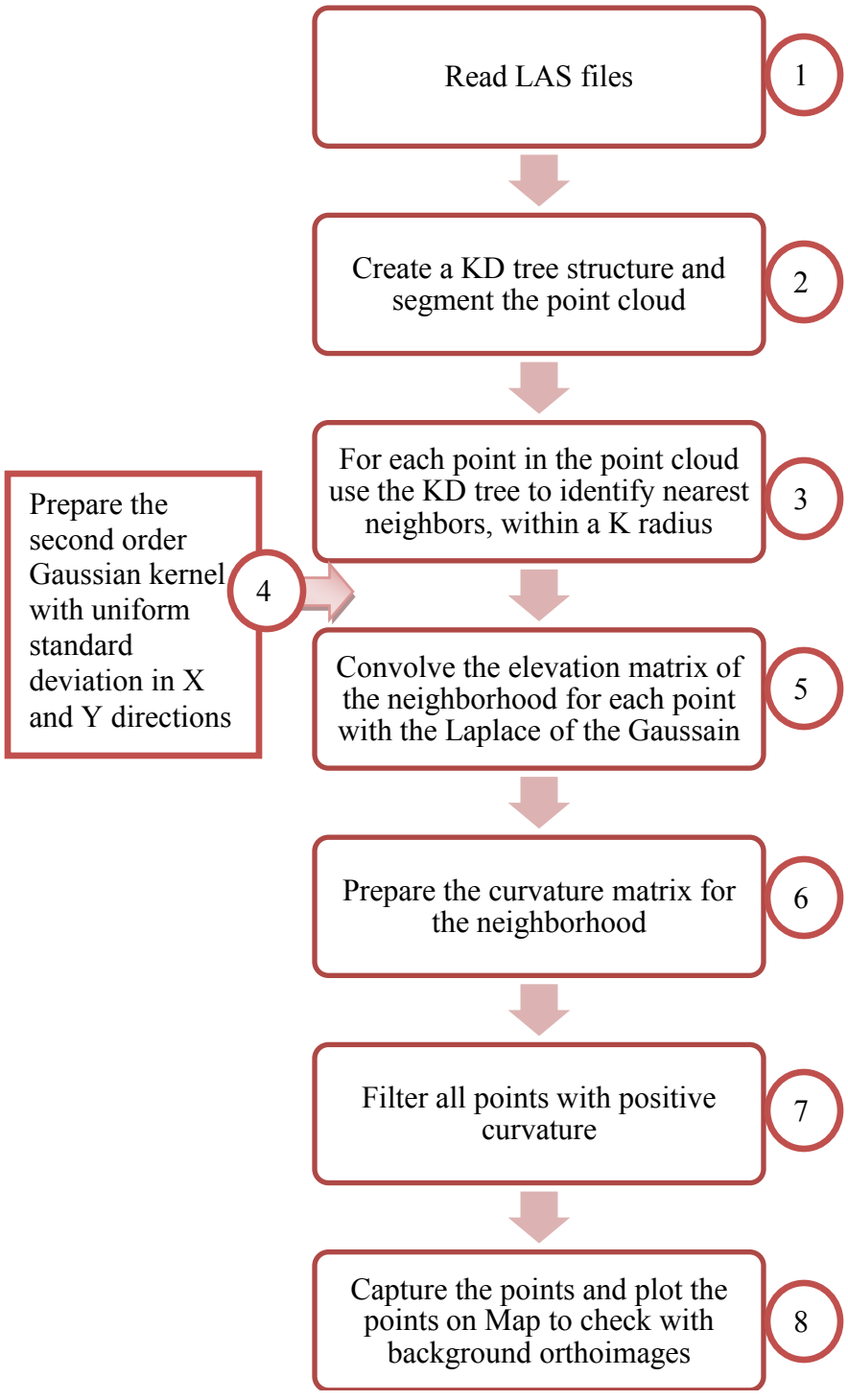

Figure 13: Workflow of the implemented methodology

Chapter 4: Water Points Identification

Central to this study is the identification of a workflow that can be used to separate water from land. The workflow shown in Figure 13 is implemented and the final result is a set of points that are identified as possible water surface points.

4.1 Study Area

In order to obtain the representative evaluation of the developed methods, six test sites were identified with specific characteristics regarding morphological settings.

4.1.1 Site Locations Point Cloud and Imagery

All the study locations are located in the Williamson County, in Texas. The geographical site locations are shown in Figure 14 and the test sites lie closer to the Granger Lake in the north east corner of the County. The Lidar data for the Williamson County is distributed as per the USGS Quarter-Quarter Quad schemas explained earlier. The test sites chosen are chosen such that the model algorithm is tested for a wide range of topological settings.

Figure 15 shows a closer look at the quarter quadrangle for Granger which is divided into four parts Granger SE, Granger SW, Granger NE and Granger NW. At the scale of Figure 15 it is seen that the test sites are located in the *Granger NE* quarter tile of the Granger quadrangle. Figure 16 shows that the tile Granger NE is divided as per 1.875 Q4 boundaries. Each of the Q4 squares is named as Granger–NEA1, Granger-NEA2, Granger-NEA3 and GrangerNEA4 and so on.

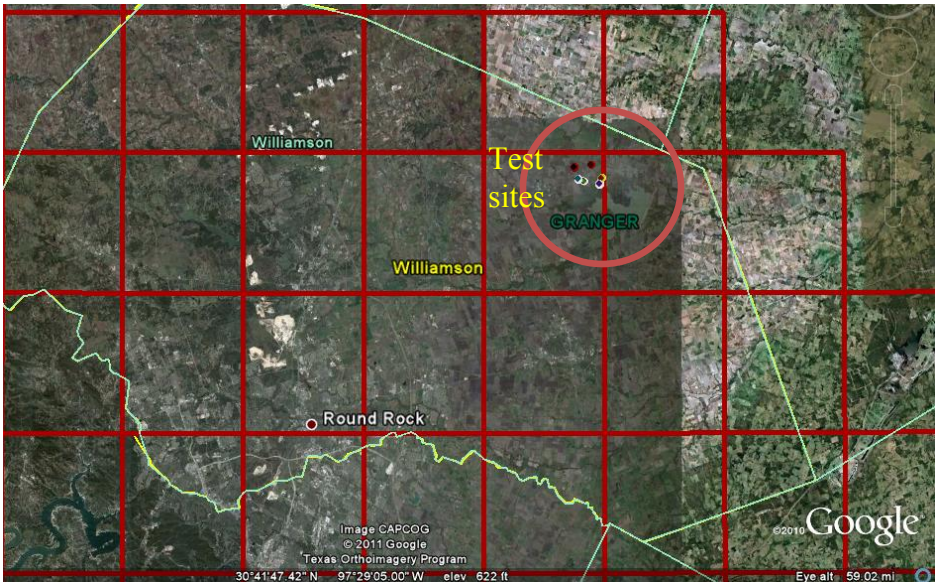

Figure 14: The 7.5 minute quadrangle boundaries for Williamson County on Google Earth by CAPCOG

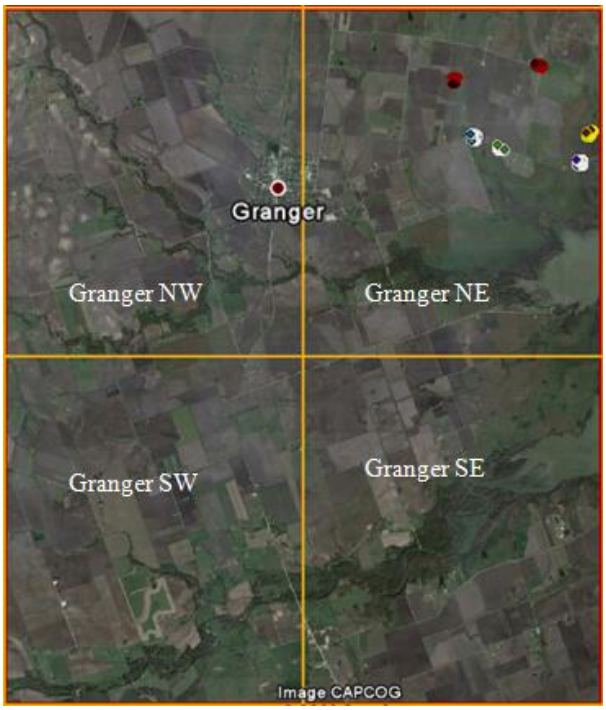

Figure 15: The 3.75 minute Quarter - Quadrangle Boundary for Granger on Google Earth by CAPCOG

Figure 16: The 1.875 minute Q4 tiles for Granger NE

The six test sites are located in the Granger-NEA2, Granger-NEB1, Granger-NEB2, Granger-NEB3, Granger-NEB4 Q4 boundaries respectively as shown in Figure 16. A single Lidar LAS file is obtained from Texas Natural Resources Information System (TNRIS 2009) for all the test sites, Granger-NEA2, Granger-NEB1, Granger-NEB2, Granger-NEB3 and Granger-NEB4 respectively. The X, Y and Z co-ordinates used the state plane coordinate system for Texas Central Zone (FIPS 4203) with units

being in feet. And the vertical datum defined for the LAS file was NAVD88 and units in feet.

LAS file name	Point Count	Point Spacing
u309721_2_a_la.las	12711947	3.040673

Table 9: Lidar file for the test sites

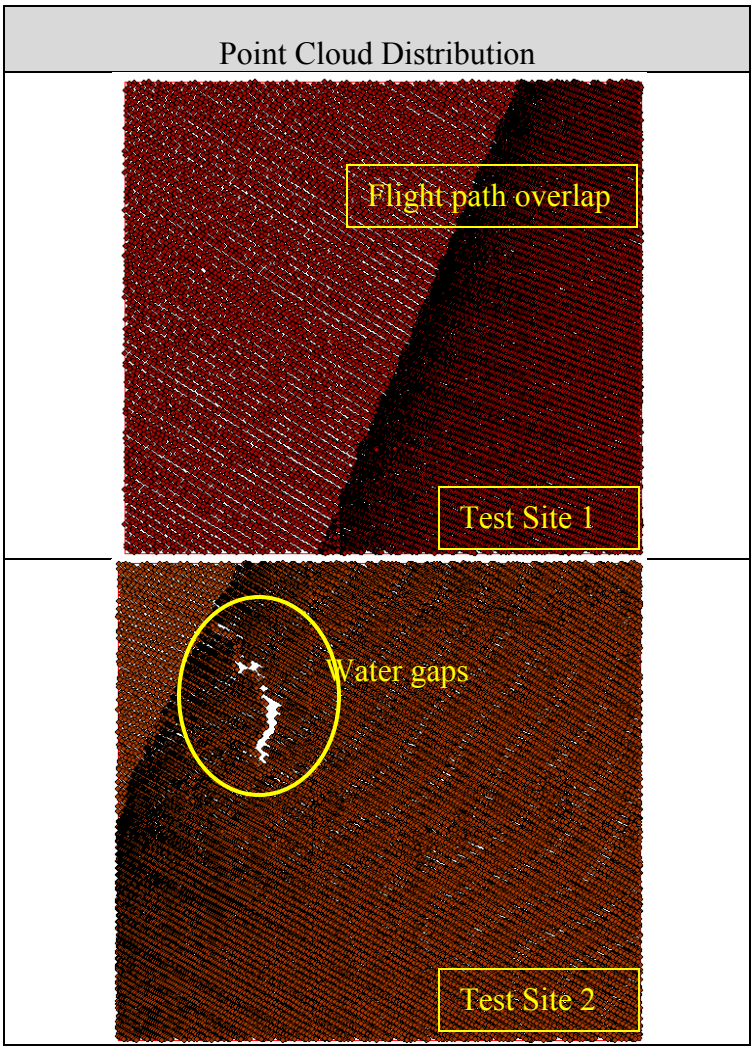

Table 10: Point cloud distribution of site 1 and site 2

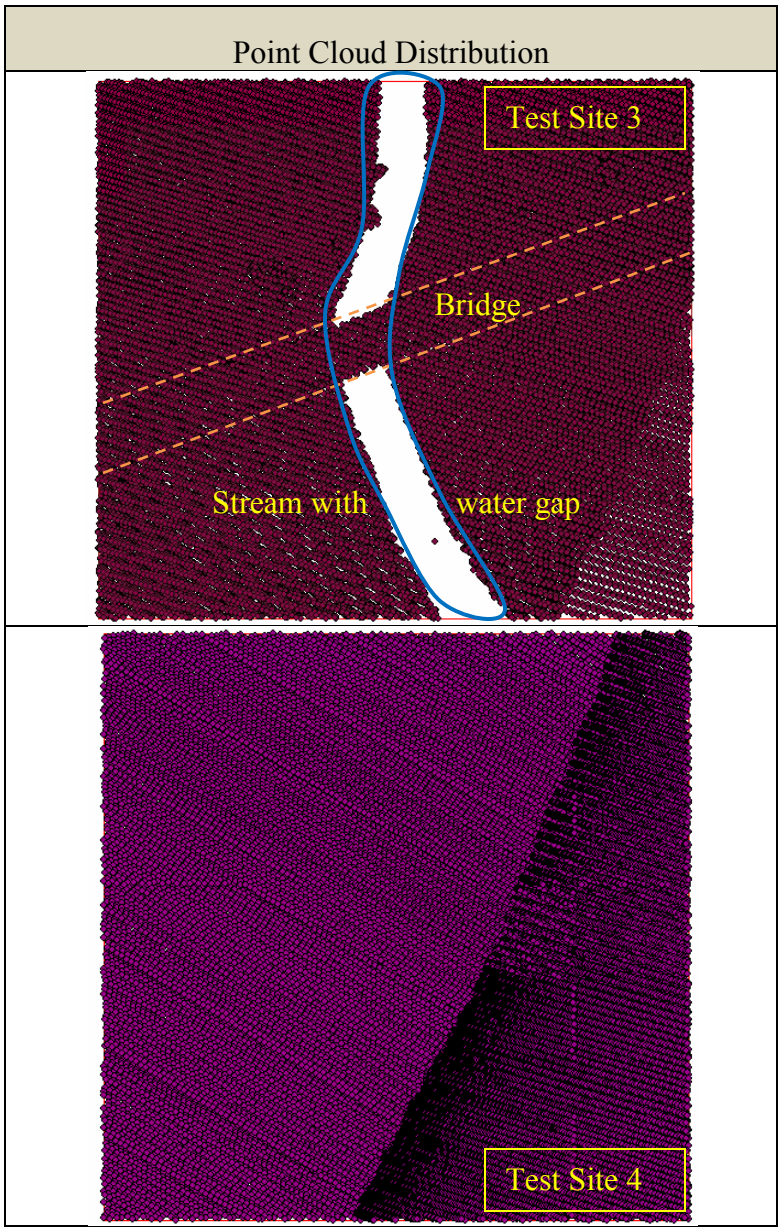

Table 11: Point cloud distribution of site 3 and site 4

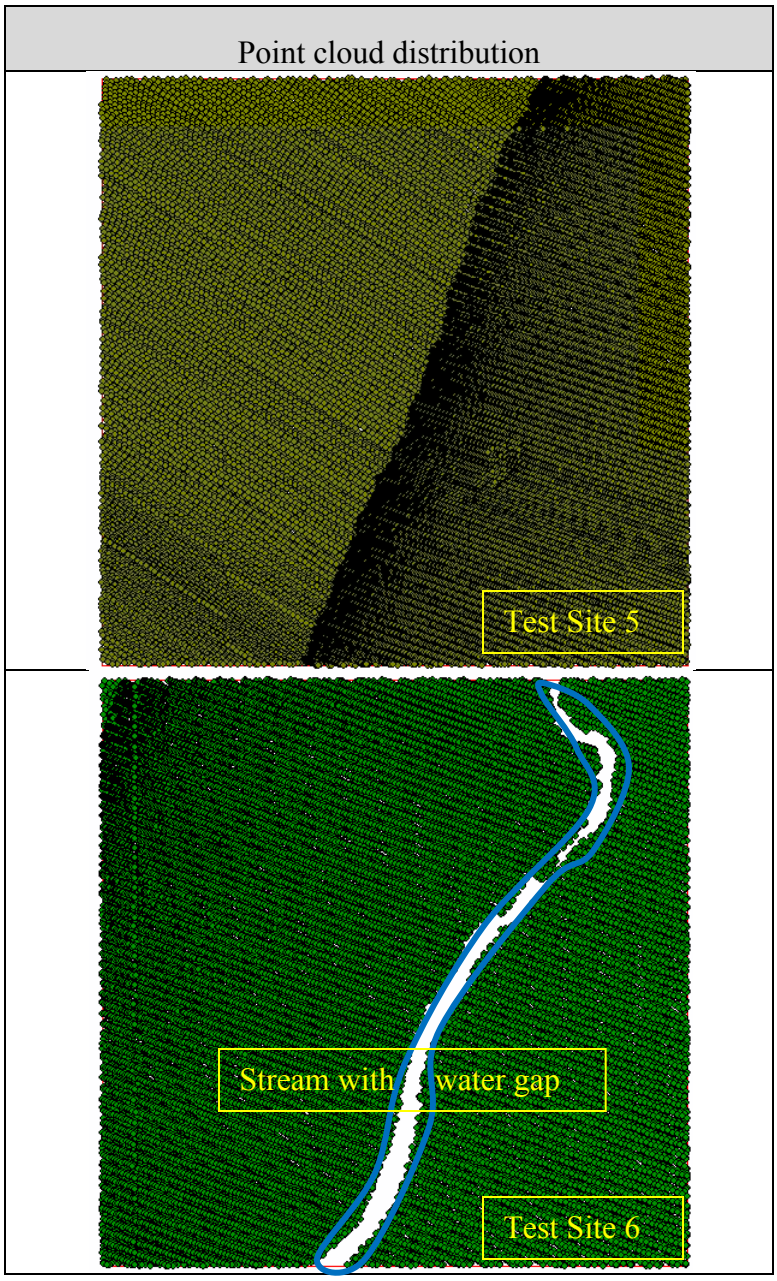

Table 12: Point cloud distribution of site 5 and site 6

From the point cloud distributions shown in Table 10, Table 11 and Table 12 it is clear that the point density and distribution of raw LAS files are not uniform across locations. The bands of high point density are due to the overlap of two different flight paths as seen in Table 10. The gaps in the point cloud in sites 2, 3 and 6 is due to the fact that there was no recorded reflection from the surface on earth and these could have resulted in water gaps (Kraus 2007) . Absence of water gaps in other site locations such as in site 1 could be due to the fact, that most of the streams were dry during the time of Lidar data collection and hence the recorded measurements of intensity and elevation at these locations matched the bare earth measurements.

High resolution imagery was used to validate the mapping accuracy of the points identified as probable water points. The following section describes each site in detail and shows the imagery for each study location accompanied by the small statistics table for each site.

1. **Site1**

   The first test site covers an area of 500 ft X 500 ft approximately. The test site lies on the boundary of the Granger-NEA2 and Granger-NEB1.

Site-1 Statistics	Statistic Value
Point count	36653
Mean elevation	532.322167435 ft
Maximum Elevation	540.367 ft
Minimum Elevation	525.268 ft
Standard deviation in Elevation	1.94568 ft

Table 13: Site 1 Statistics

Topographically the site is characterized by two small streams converging into single branch. The surrounding areas are relatively flat and have no roads crossing the study area.

Figure 17: Test Site1 high resolution imagery from Google Earth

2. Site 2

The study area lied in Granger-NEB3 tile. Site 2 has a standard deviation in elevation lot higher than site 1. Figure 18 shows the selected site location. With a study area of 500 ft X 500 ft this site, has a small channel or drain that appears to be cut by an elevated road surface. A small pond on the right hand side of the study area and some trees in the middle of the study location were other features of interest observed in the site settings.

Site-2 Statistics	Statistic value
Point count	48638
Mean elevation	523.618196595 ft
Maximum elevation	564.8 ft
Minimum elevation	508.663 ft
Standard deviation of elevation	7.04909723163 ft

Table 14: Site 2 Statistics

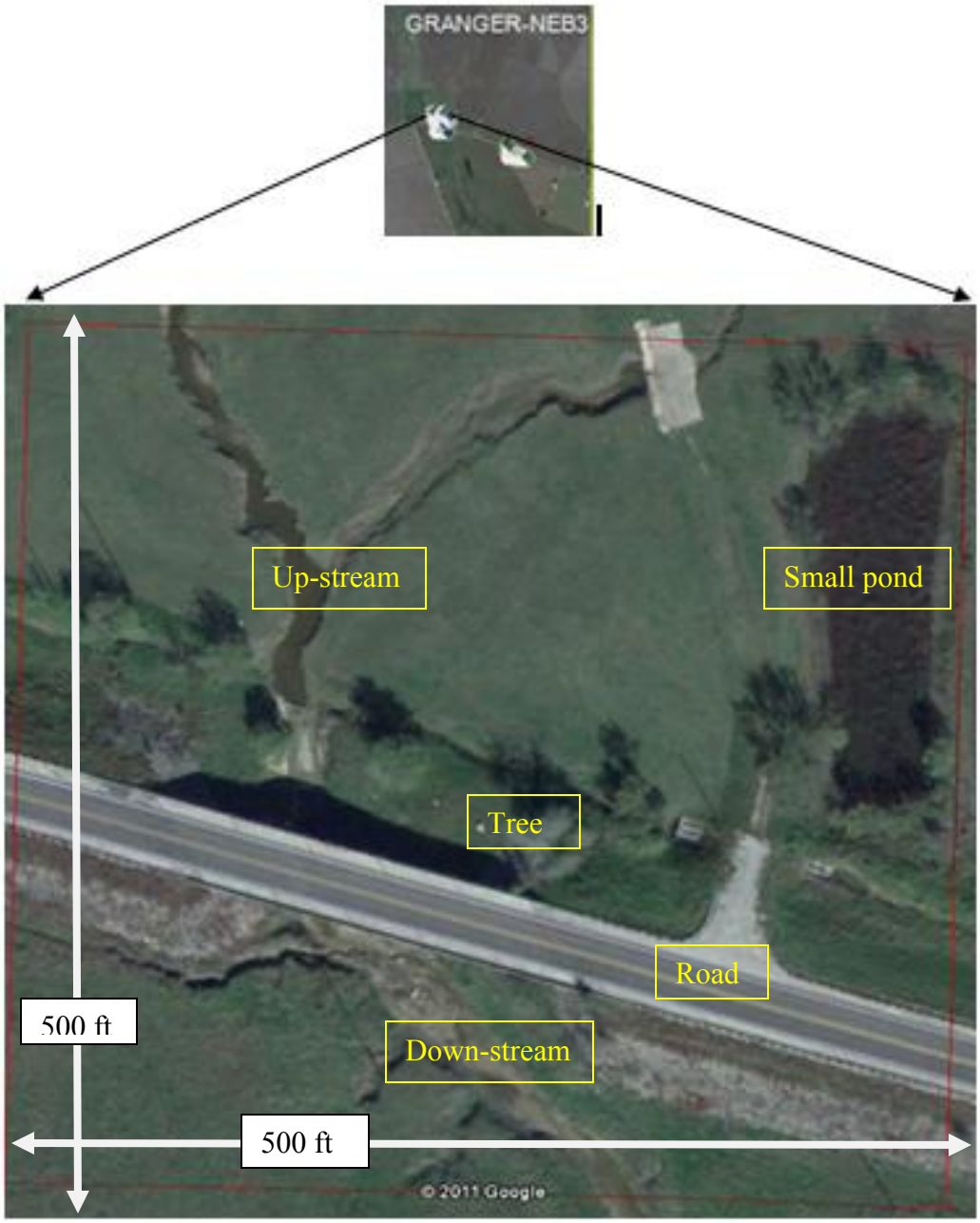

Figure 18: Test Site 2 high resolution imagery from Google Earth

3. Site 3

The third site is located neared to the mouth of the Granger Lake, with flat topography all around the site. The site featured a stream draining into the lake and an elevated road surface crossing the stream. The areas around the stream appear to be part of the flood plain of the stream. The site is located in the Granger-NEB4 tile. The average area is around 500 ft X 500 ft. Apart from streams and elevated road surface, trees and gravel bed along the road were other features of interest.

Site-3 Statistics	Statistic value
Point count	46253
Mean elevation	515.878739066 ft
Maximum elevation	578.655 ft
Minimum elevation	503.652 ft
Standard deviation of elevation	10.4220530291 ft

Table 15: Site 3 statistics

4. Sites 4 and Site 5

The fourth and fifth sites are similar in the topographic setting. Both the study locations have roads, trees and farm lands. The only difference between them is that site 5 has a small stream cutting on the top of the study location. The site 4 is located in the tile Granger-NEB3 and site 5 is located in the tile Granger-NEB2. Table 16 and Table 17 contains the summarized site statistics.

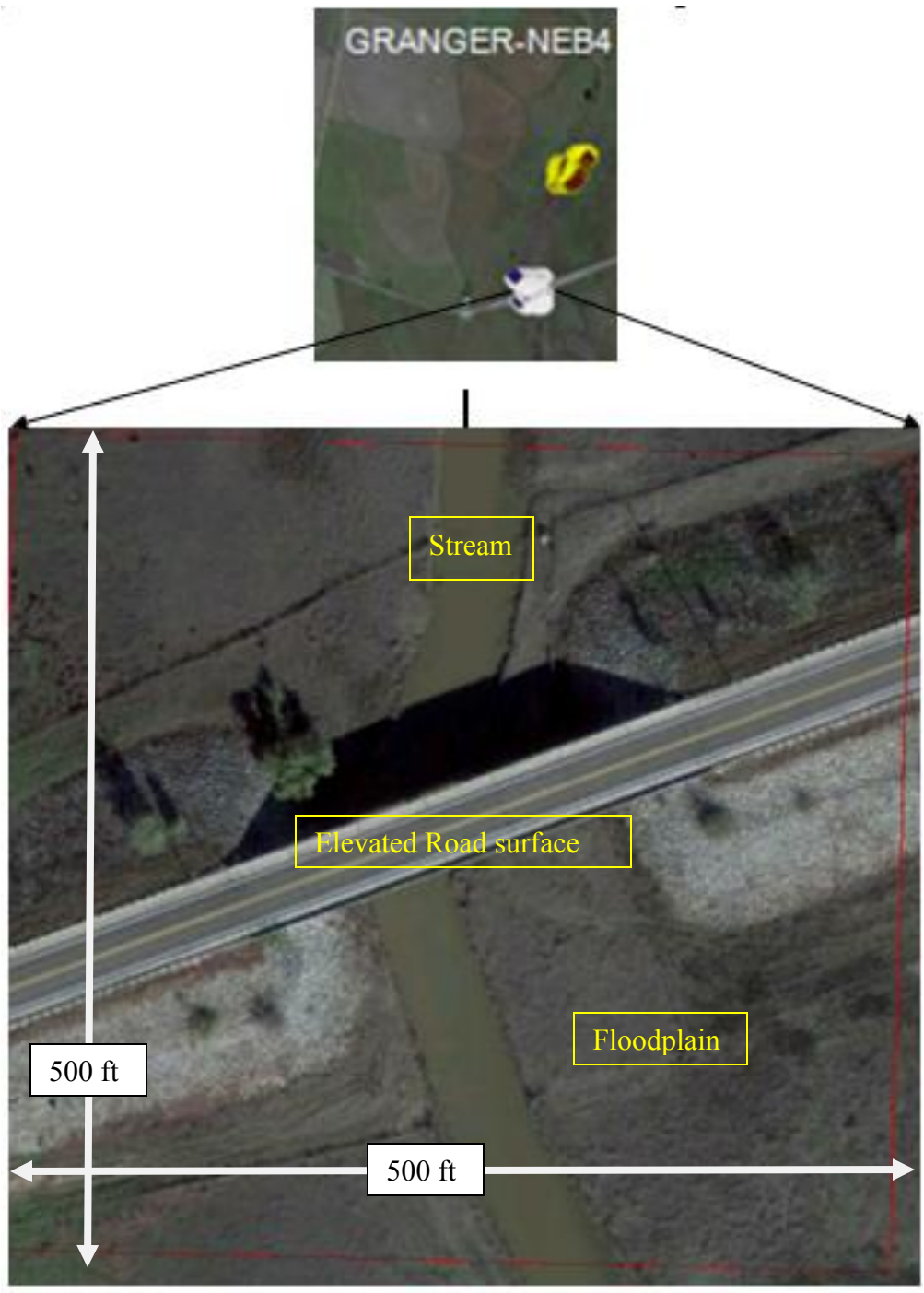

Figure 19: Test Site 3 high resolution imagery from Google Earth

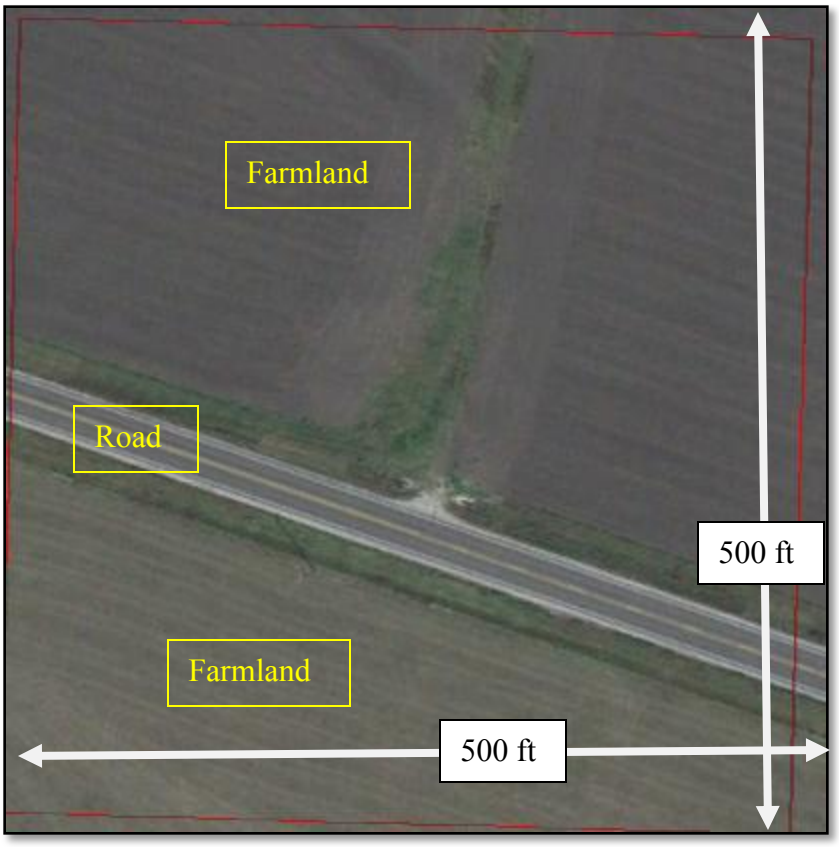

Figure 20: Test site 4 high resolution imagery from Google Earth

Site 4 – Statistics	**Statistic Value**
Point count	38467
Mean elevation	542.544101022 ft
Maximum elevation	547.86 ft
Minimum elevation	536.83 ft
Standard deviation of elevation	2.34610159831 ft

Table 16: Site 4 statistics

Site 5 – Statistics	Statistic Value
Point count	46148
Mean elevation	547.94382866 ft
Maximum elevation	585.791 ft
Minimum elevation	536.75 ft
Standard deviation of elevation	5.62358201177 ft

Table 17: Site 5 Statistics

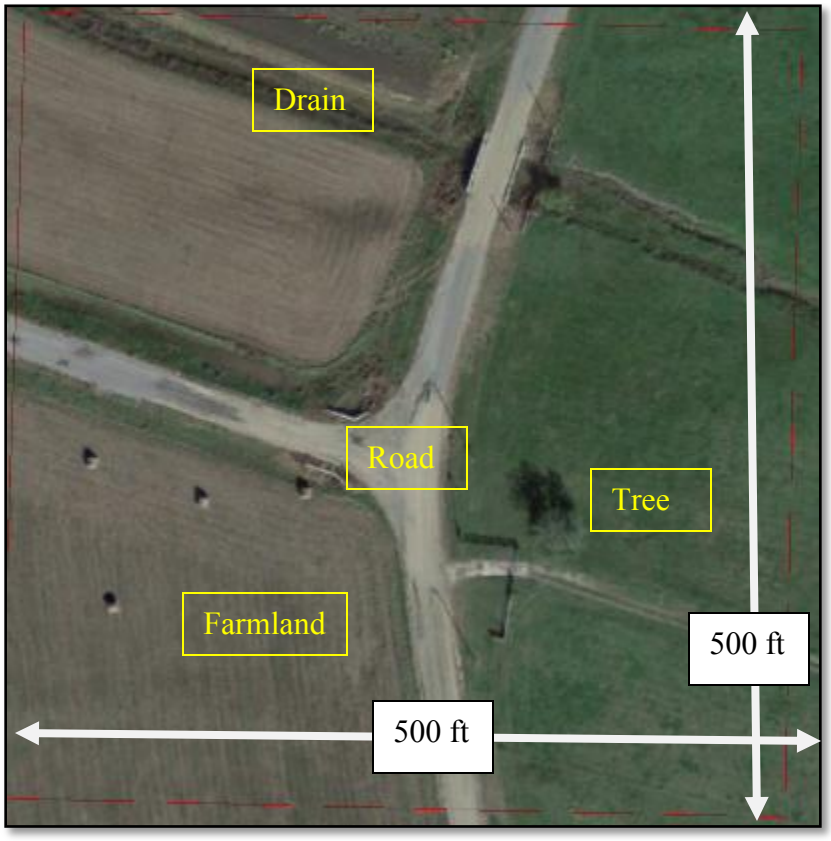

Figure 21: Test site 5 high resolution imagery from Google Earth

5. Site 6

Site 6 is located upstream of site 3. This is a simple location with few features. The study location primarily consists of a stream running along the length of the site and all the surrounding areas are low lying flood plains of the stream. The site 6 is located in tile Granger-NEB4.

Site 6 – Statistics	Statistic Value
Point count	42752
Mean elevation	511.884561377 ft
Maximum elevation	525.308 ft
Minimum elevation	503.679 ft
Standard deviation of elevation	1.80629614728 ft

Table 18: Site 6 statistics

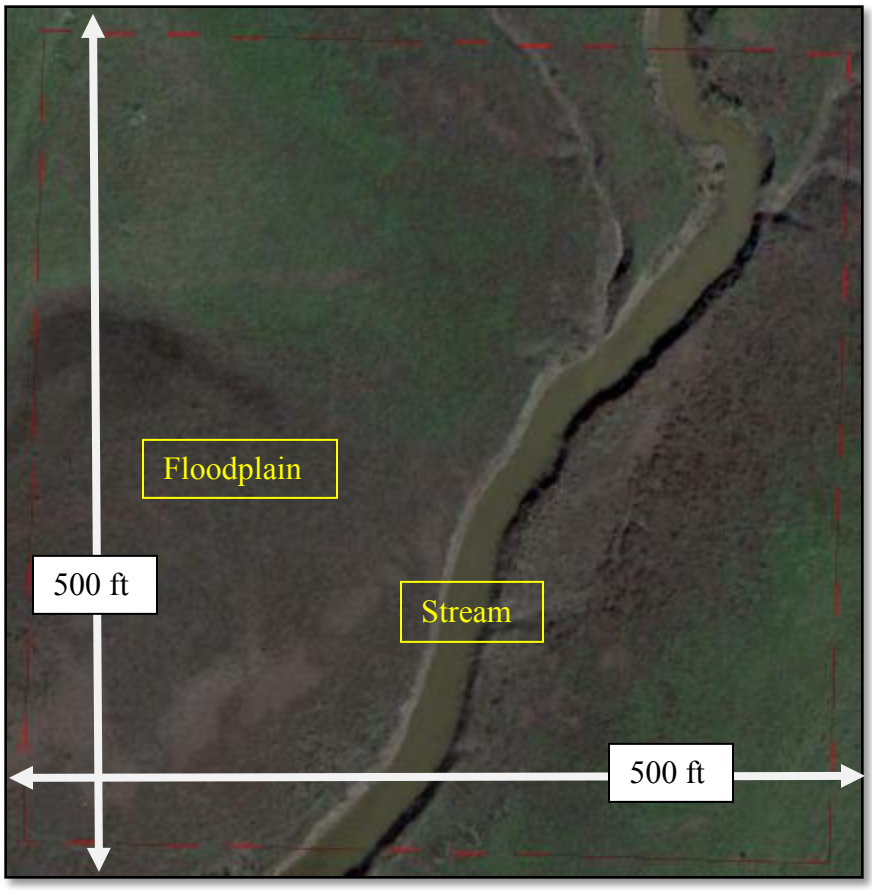

Figure 22: Test Site 6 high resolution imagery from Google Earth

4.2.1 Algorithm Settings

Before discussing the final results of the model it is important to discuss the algorithm settings which are used to achieve the results. Three different settings are discussed below which are crucial for each of the model runs.

4.2.1.1 Suitable Choice of Local Neighborhood

Point density and uneven pre-processing of the Lidar files is a major concern. The bands of high point density in areas of flight path overlap challenged the selection of suitable neighborhood for each point. The average width of streams in the study area is around 15 ft. Thus a neighborhood of points within a radius of 15 ft is sought for each

point record. This measure also determines the standard deviation of the kernel which is used for estimating the curvature. The second order derivative of a Gaussian kernel has a standard deviation of 3σ; hence the σ used for building the Gaussian kernel is 5ft.

4.2.1.2 Threshold Selection

The statistics of elevation values for each study area are calculated. The mean, maximum, minimum and standard deviation of the elevation values are estimated once for each study location and are used as a filtering criteria for eliminating outliers if any.

Features such as roads, bridges and trees are also areas of positive curvature and could be identified as water points. The difference in values of intensity for roads, low lying vegetation and water surface points is very small. Thus identifying only water surface points in the presence of all other features requires a combination of elevation and curvature filtering.

The elevation threshold defined as $\left(mean_{elev} - \left(\frac{Standarddevaition_{elev}}{2}\right)\right)$ is kept constant for all the study areas. And since no proper distinguishing limits are found in intensities between roads, trees and water surface points, the intensity filtering is not considered for this study. The final filtering criteria used for all the test sites is

$$curv_{value} > 0 \; and \; elevation < \left(mean_{elev} - \left(\frac{Standarddevaition_{elev}}{2}\right)\right) \qquad (15)$$

where, $curv_{value}$ is the curvature at each point, $mean_{elev}$ is the mean elevation of the study location and $Standarddevaition_{elev}$ is the standard deviation of elevation values in the study location.

The Q-Q plots of curvature are plotted for each site to look at the distribution of curvature values over the landscape. The distribution of the curvature values is observed to be the similar over all the test sites.

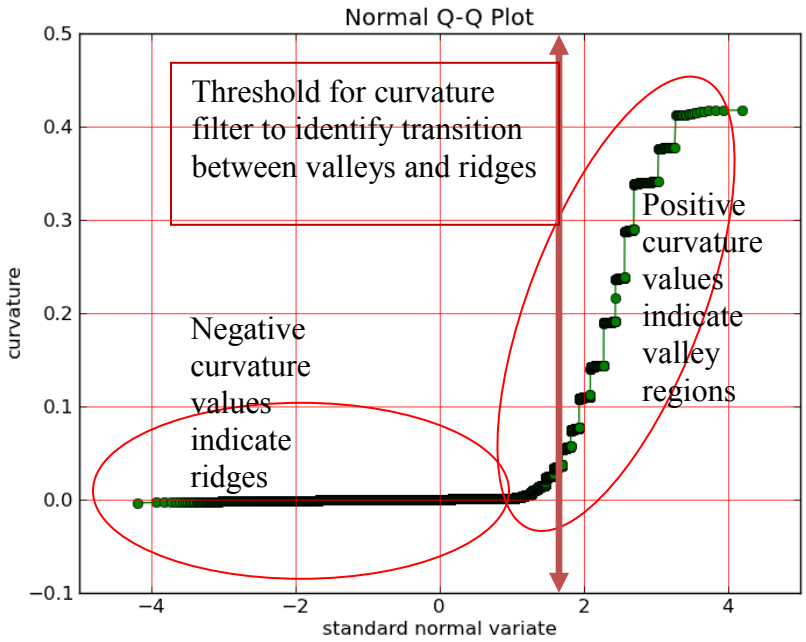

Figure 23: Normal Q-Q plot for Site 2

4.3 Model Results and Assessment

The model algorithm is applied to all the six sites in Williamson County. The resulting probable water points identified are converted into features for visualization and plotted on top of high resolution imagery to access the accuracy of the identified points.

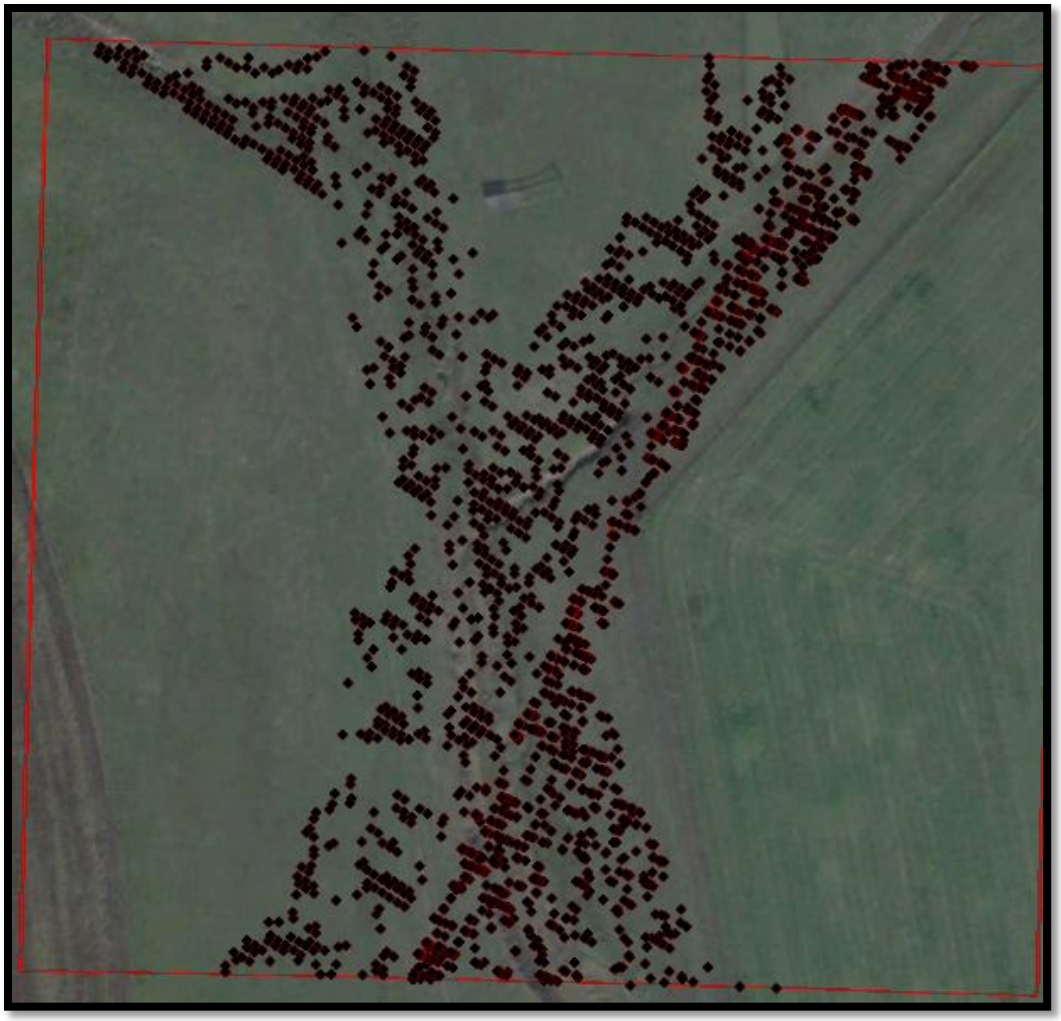

Figure 24: Identified water points at Site 1

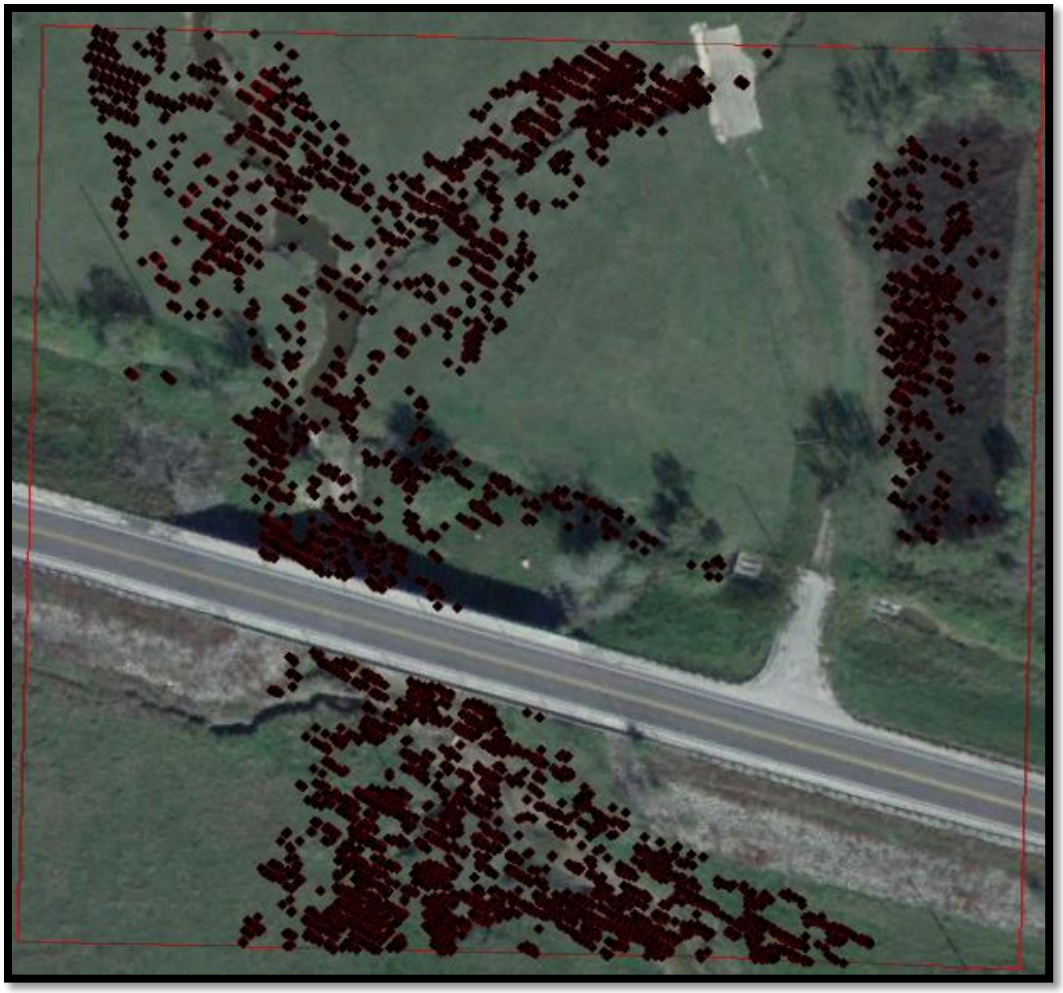

Figure 25: Identified water points at Site 2

The results from Site-1 and Site-2 are shown in Figures 24 and Figure 25 respectively. It is clear that the model is able to identify water surface points accurately. The resulting points were identified around the streams within a buffer of 15 ft from the center lines of stream lines obtained from National Hydrography Dataset Plus (NHD Plus). The results from Site-2 indicate that the model not only identifies streams as in Site-1 but it also identifies small marsh lands, drains and all points of low elevation, that could possibly be flooded in the event of a heavy storm. The additional filtering criteria

based on elevation helps in omitting roads and bridges as being identified points of positive curvature.

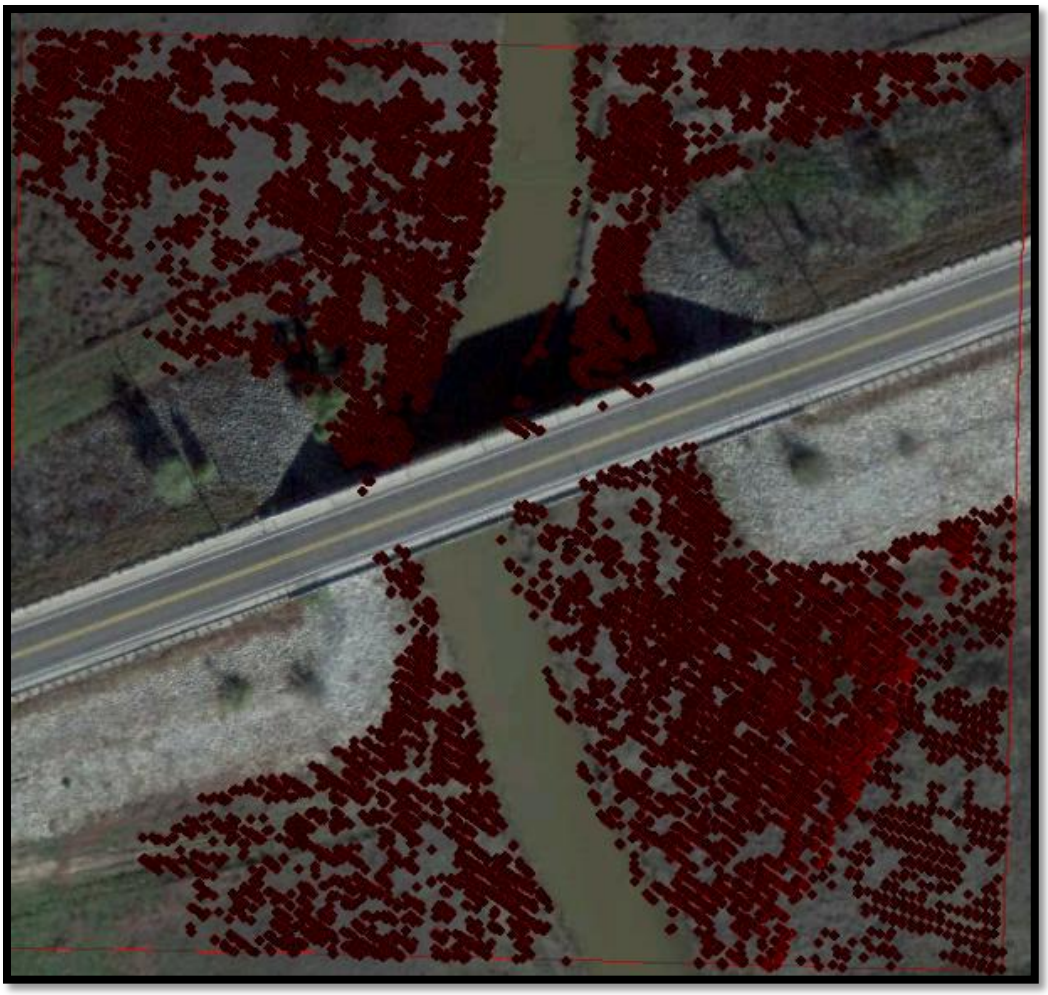

Figure 26: Identified water points at Site 3

The results of Site-3 suggest that the model is able to omit roads and only identify streams and flood plains around them. Since the point cloud for Site-3 had all points removed from the water surface during the pre-processing steps, the model did not detect any points on the water surface. The model does very good in detecting low lying areas around the stream and thus identifies a major part of the study location as flood plain. It

can also be seen from the imagery that these areas are darker in color suggesting wet soil and possible flood plains.

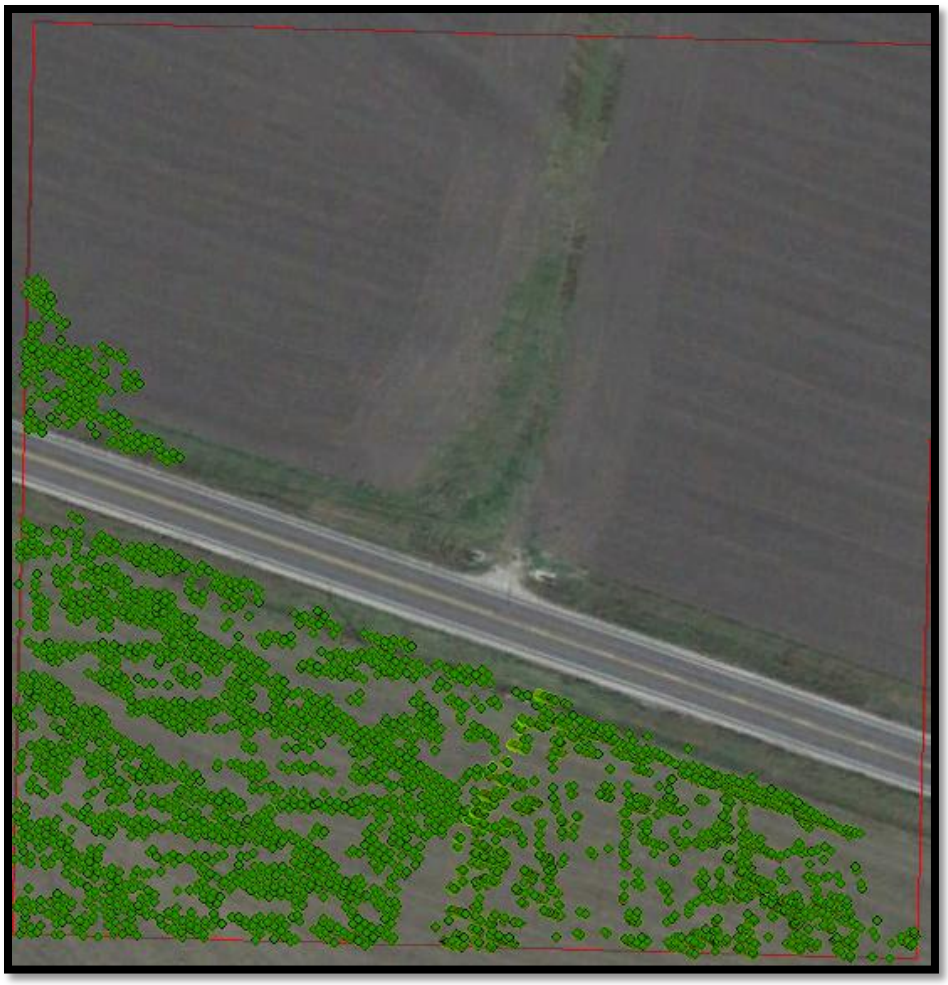

Figure 27: Identified water points at Site 4

The results for Site-4 identify the farmlands as possible water surface points. The study area has neither streams nor drains and hence ideally the model should have identified none of the points. On the contrary the model finds all points of positive curvature. This suggests that the neighborhood that is used for estimating the curvature should be a global subset and not a local subset of the point cloud. And also the

corresponding elevation and intensity thresholds should be determined globally and not for smaller subsets of the point cloud.

Figure 28: Identified water points at Site 5

The identified subset of point cloud for Site-5 also suggests ideas similar to those originating from analysis of Site-4. In this case the model identifies the stream and surrounding low lying areas, thus showing potential in delineating floodplains for a stream.

The results for Site-6 shown below are very promising and show that the model is able to pick and select points around streams and other depressions on the earth surface.

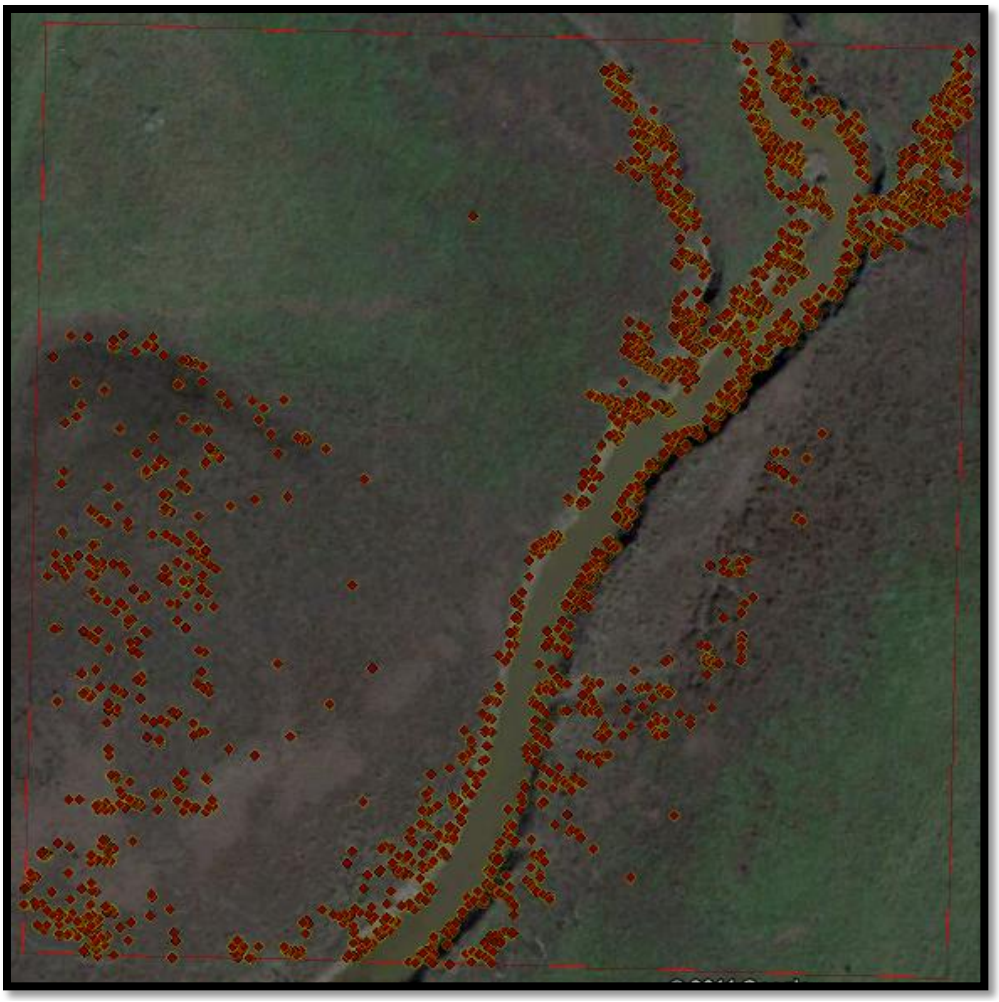

Figure 29: Identified water points at Site 6

Once the points are filtered, a TIN surface was fitted to site 2 and site 3. Figure 30 shows site 2 in 3D created from the unfiltered point cloud. The 3D representation shows that the stream is cut by a road; there are some trees and low lying vegetation present in the bare-earth point cloud. Figure 31 shows the TIN created form the points identified as possible water surfaces in site 2. It is interesting to note that the roads which appeared as

dams in Figure 30 are not present any more. The low lying vegetation and trees also do not appear any more.

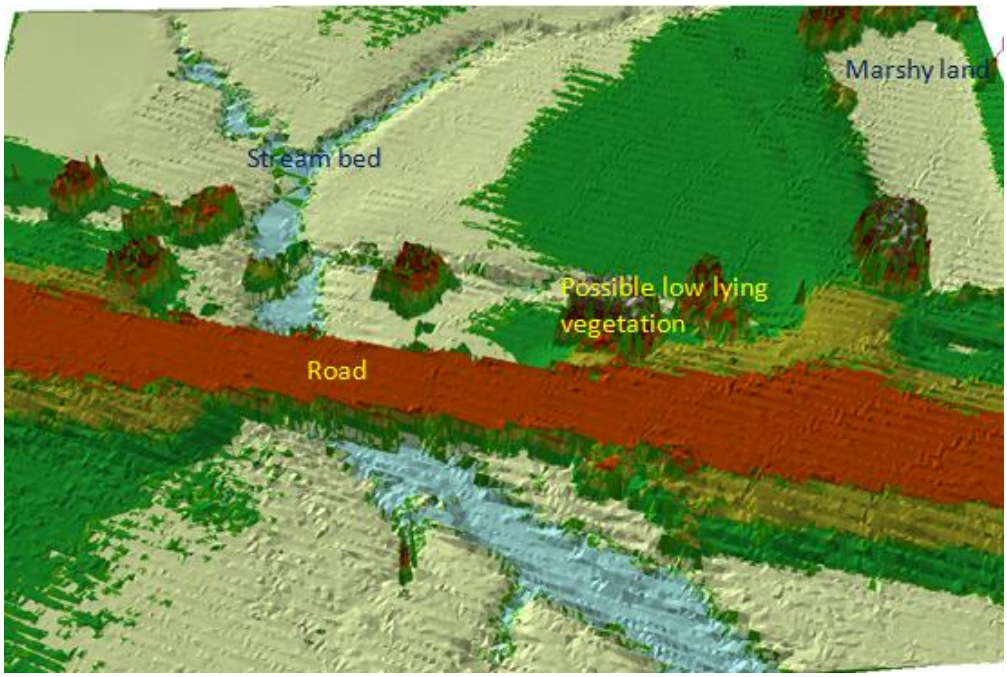

Figure 30: TIN surface created from point cloud for site 2

Figure 32 and Figure 33 show the 3D TIN surface created for site 3 from unfiltered and filtered point cloud dataset. Figure 32 shows the 3D terrain for site 3 consisting of an elevated road surface and embankments of the road, a stream bed and floodplains of the stream and some low lying vegetation. An artificial dam is created at location where the road crosses the stream.

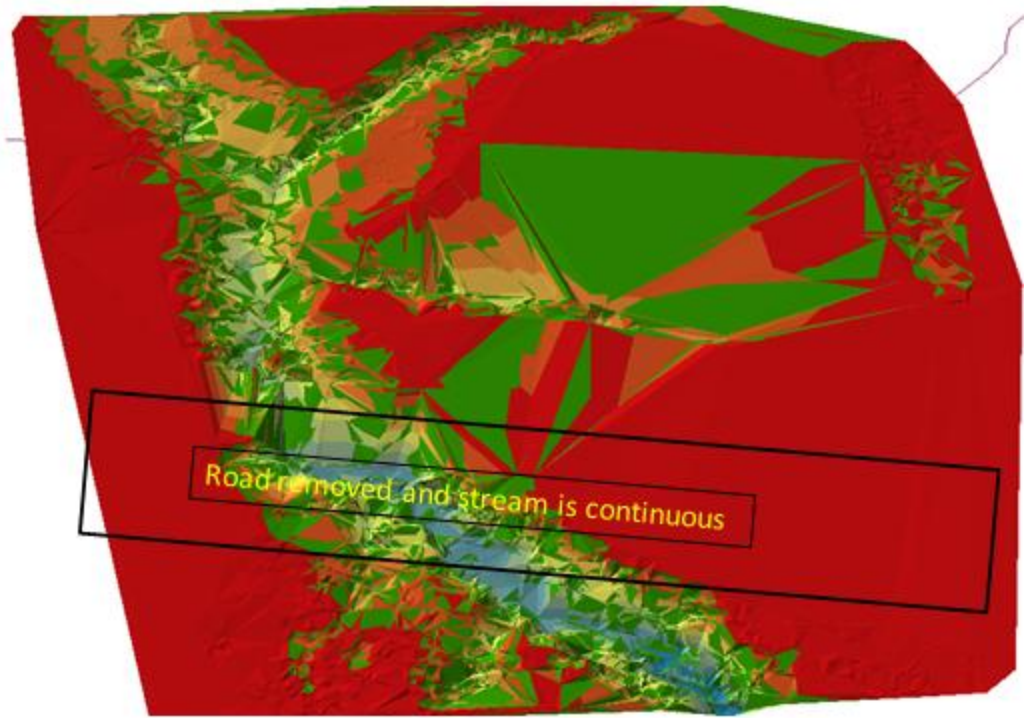

Figure 31: TIN surface created from the filtered point cloud for site 2

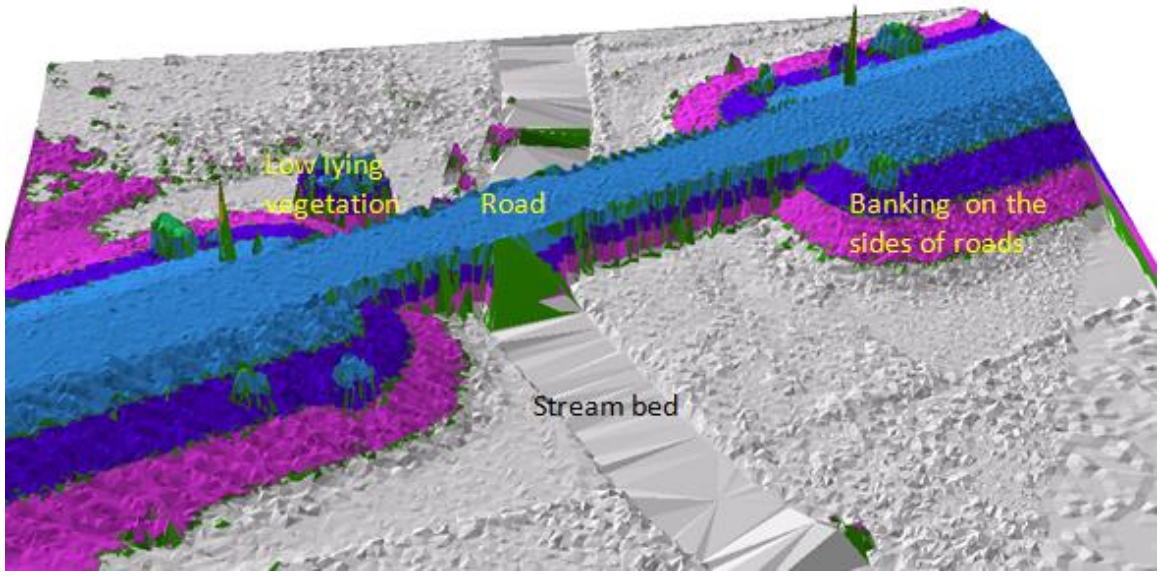

Figure 32: TIN surface created from the point cloud for site 3

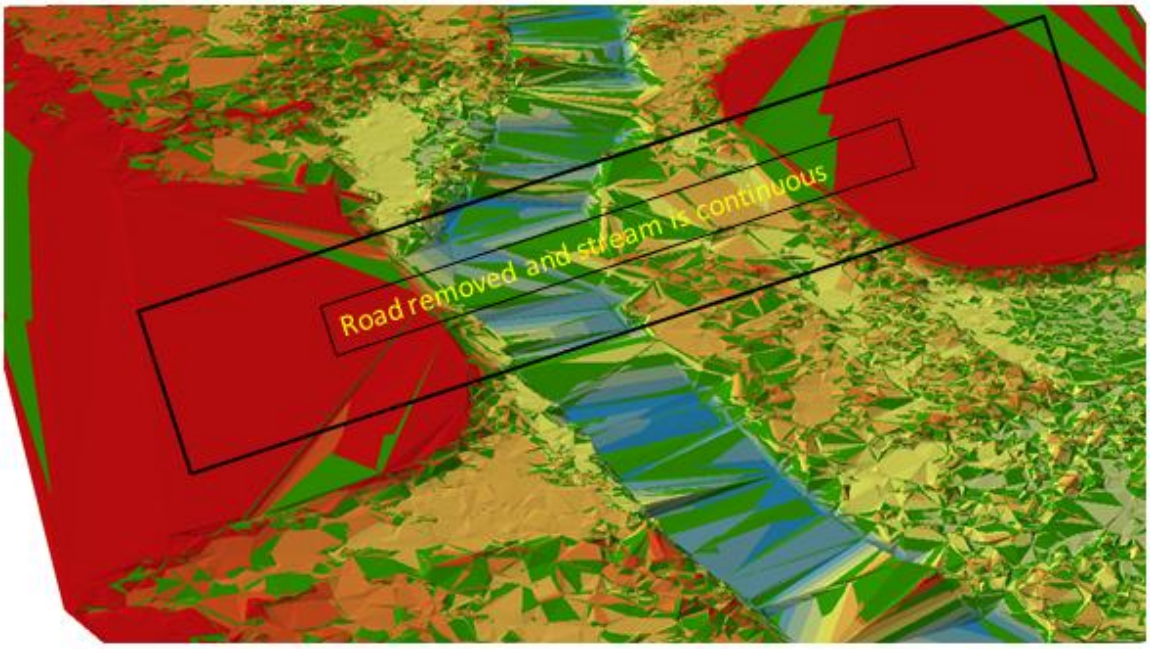

Figure 33: TIN surface created from filtered point cloud for site 3

Figure 31 and Figure 33 shows the 3D TIN surface created from a filtered point cloud dataset obtained after the classification procedure outlined in this study. Both the figures show that the TIN surface created from the filtered point cloud dataset are able to remove the roads and other low lying vegetation from the final point cloud dataset. This prevents the creation of artificial dams due to the presence of roads and other features that can cross the streams.

The model algorithm was tested on six different topographic settings and the results in the previous section show that the model is consistently able to identify water points. The results also suggest the use of a global threshold instead of a local threshold for building the neighborhood subset. It is important to identify the scales at which such a global threshold can be defined. The model results also suggest that the spatial extent of the identified points from the center line of the streams is a function of the standard

deviation of the kernel. The average stream width is used as the radius of neighborhood search for each point and is also used for defining the standard deviation of the Gaussian kernel. Since the width of streams varies from location to location, the determination of standard deviation remains a function of the stream width specific to the study location and needs further study and research.

The model results also suggest that the identified points not only indicate the stream beds but also identify the flood plains. The use of positive curvature points can provide us with a set of valley points and can help in identifying all locations which are susceptible to water logging. On the contrary the points having negative curvature values can also be identified as hills and ridges and can be used to identify watershed lines.

Chapter 5: Conclusions

6.1 What Has Been Learned

The motivation behind conducting this research is to properly segment and classify point cloud into water and non water points. Identifying floodplains and stream beds is a very challenging and difficult task. The massive size of the point cloud dataset, the absence of connectivity and the three dimensional distribution of the datasets make the task of analyzing the datasets very complicated. The National Research Council's committee on Floodplain Mapping Technologies (NRC, 2007) concluded that new elevation data should be collected and should employ Lidar as the primary technology for digital elevation data acquisition and the FEMA floodplain maps should be updated with consistent elevation data. Lidar data is being collected all over the United States of America.

For hydrological modeling, the high resolution of Lidar helps in creating true representation of terrain, and hence correct filtering or classification of terrain to the highest level of detail, is crucial. For hydrological modeling the terrain data should reflect the ability of water to run under objects. Topological issues such as water gaps are left to the end users.

During the course of study and research it is observed that a lot of work and effort has been spent on extracting channels and streams by creating digital elevation models from Lidar data and some form of filtering technique is used to extract streams. Another approach of detecting features is by using training datasets to determine parameters and thresholds for elevation and intensity values and after numerous runs, these thresholds are used to classify the point cloud into various features. Most of the later techniques are

used in studies to identify tree and canopy heights in forest regions. Identification of vegetation type and differentiating ground points from non-ground points is the major motive behind all classification work done in the literature.

Recently some research was done in identifying rivers and streams from point cloud by using intensity correction and drop out modeling (Hofle, et al. 2009). To the knowledge of the author classification of point cloud to identify possible water points by using wavelet analysis has not been done before.

This research has shown that it is possible to classify point cloud datasets as water. By reading the elevation values, and by using the Laplacian curvature via wavelets, we can assign curvature values to each point based on the neighborhood of each point without the need to generate surfaces such as raster, terrains and TINs. The research has also shown that we can classify points not only those on the streams, but also those on the floodplains of these streams.

The point cloud datasets are not used as frequently by analysts as one would expect. The main reason for not using of what is actually the original data, is that most data users are far more familiar with grids, the way grids are loaded, presented and analyzed in common GIS software. A second reason is that many standard installations of GIS systems do not contain modules for handling point cloud data. Another reason is that, in particular because of the distribution of data in ASCII format, the sheer size of the files posse's obstacles to users or system administrators to load the data. Furthermore it cannot be denied that also the lack of knowledge, communication and documentation hampers the potential use of point cloud data.

6.2 AREAS OF FUTURE RESEARCH

The current model and algorithm has been tested on smaller datasets and in rural areas which are relatively flat. Future work will focus on adding functionality to improve the application of the model to larger datasets. The threshold and results from some sites indicate the need to develop global parameters and threshold levels to avoid detecting low lying farmlands as potential water points. There is a need to employ data structures that will support reading huge point cloud datasets and store them for efficient neighborhood search.

Appropriate statistical error estimates should be devised to measure the inaccuracies in the model. The visual inspection is by far the best method to validate and check results, but this method won't be suitable for larger areas and datasets. Even though the literature review about intensity attributes and their use, few of the Lidar dataset attributes have been used for classification. After extracting and classifying points in Lidar datasets TINs and Terrains cab be built on these sampled points and thus help in reducing the number of nodes in such data structures, which is crucial for numerical hydrological modeling.

Bibliography

Ambercore Terrapoint. A white paper on Lidar Mapping. White paper, Woodlands Texas: Ambercore Terrapoint Division Offices, 2008.

Antonarakis, A. S., K. S. Richards, and J. Brasington. "Objects-based land cover classification using airborne LiDAR." Remote Sensing of Environment, 2008: 2988–2998.

Arefi, H, and M Hahn. "A morphological reconstruction algorithm for separating off-terrain points from terrain points in laser scanning data." Proceedings of the ISPRS Workshop Laser Scanning. Enschede, 2005.

ASPRS Standards Committee for Lidar. ASPRS LIDAR Data Exchange Format Standard Version 1.0. ASPRS, 2003.

Axelsson, P. "DEM Generation from Laser Scanner Data Using Adaptive TIN Models." International Archive of Photogrammetry and Remote Sensing, 2000: 110-117.

Bentley, J. L. "Multidimensional Divide and Conquer." Communications of the ACM. 1980. 214-229.

Brennan, R., and T. L. Webster. "Object-oriented land cover classification of lidar-derived surfaces." Canadian Journal of Remote Sensing, 2006: 162–172.

Briese, C, and N Pfeifer. "Airborne laser scanning and derivation of digital terrain models." Proceedings of Fifth Conference on Optical 3-D Measurement Techniques. Vienna, Austria, 2001.

Brovelli, M A, M Cannata, and U Longoni. "Lidar data filtering and DTM interpolation within grass." Transactions in GIS, 2004: 155-174.

Brzank, A, C Heipke, J Goepfert, and U Soergel. "Aspects of generating precise digital terrain models in the Wadden Sea from lidar-water classification and structure line extraction." ISPRS Journal of Photogrammetry and Remote Sensing, 2008: 510–528.

Chen, Z T, and J A Guevara. "Systematic Selection of Very Important Points (VIP) for Digital Terrain Model for Construction Triangular Irregular Networks." AutoCarto 8 ASPRS/ACSM. Baltimore, MD, 1987. 50-56.

Clode, Simon, Franz Rottensteiner, Peter Kootsookos, and Emanuel Zelniker. "Detection and vectorization of roads from lidar data." Photogrammetric Engineering and Remote Sensing, 2007: 517-535.

Clode., SP, F Rottensteiner., and P Kootsookos. "Improving city model determination by using road detection from lidar data." International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2005: 159–164.

Elmqvist, M, E Jungert, F Lantz, A Persson, and U Soderman. "Terrain modelling and analysis using laser scanner data." International Archive of Photogrammetry and Remote Sensing, 2001: 211-218.

Elmqvist, M. "Ground surface estimation from airborne laser scanner data using active shape models." Proceedings of ISPRS Commission III Symposium, Photogrammetric and Computer Vision. Graz, Austria, 2002. 114-118.

ESRI. http://help.arcgis.com/en/arcgisdesktop/. 2010.

Evans, J.S., and A.T. Hudak. "A multiscale curvature algorithm for classifying discrete return LiDAR in forested environments." IEEE Transactions on Geosciences and Remote Sensing, 2007: 1029-1038.

Filin, S. "Surface clustering from airborne laser scanning data." International Archive of Photogrammetry and Remote Sensing and Spatial Information Science, 2002: 119-124.

Flood, M. "LIDAR activities and research priorities in the commercial sector." International Archives of the Photogrammetry,Remote Sensing and Spatial Information Sciences, 2001: 3-8.

French, J.R. "Airborne LiDAR in support of geomorphological and hydraulic modelling." Earth Surface Processes and Landforms, 2003: 321-335.

Graham, D, M Sault, and C Bailey. "National ocean service shoreline: past, present, and future." Journal of Coastal Research, 2003: 14-32.

Gumhold, S., X. Wang, and R. McLeod. "Feature extraction from Point clouds." Proceedings of 10th International Meshing Roundtable, 2001.

Hanan, S. Applications of Spatial Data Structures. College Park, MD: Institute of Advanced Computer Studies and Center for Automation Research, 1990.

Haugerud, R.A., and D.J. Harding. "Some algorithms for virtual deforestation (VDF) of LiDAR topographic survey data." International Archive of Photogrammetry and Remote Sensing, 2001: 219-226.

Heslin, J, W.J. Lillycrop, and R. Pope. "CHARTS: An evolution in airborne hydrography." U.S.Hydro 2003. Biloxi MS, 2003.

Hildebrand, K., K. Polithier, and M. Wardetzky. "Smooth feature lines on surface meshes." Proceedings of Symposium on Geometric Processing, 2005.

Hofle, B, M Vetter, N Pfeifer, G Mandlburger, and J Stotter. "Water surface mapping from airborne laser scanning using signal intensity and elevation data." Eart Surface Processes and Landforms, 2009: 1635–1649.

Hubeli, A., and M. Gross. "Multiresolution feature extraction for unstructured meshes." Proccedings of IEEE Visualization, 2007: 287–294.

Isenburg M, Lindstrom P. "Streaming Meshes." Proceedings of the 16th IEEE Visualization 05. Minneapolis, 2005. 231-238.

Kraus, K. Photogrammetry. Berlin, 2007.

Lashermes, B., E. Georgiou. Foufoula, and W. E. Dietrich. Channel network extraction from high-resolution topography using wavelets. Geo physical Res letters, 2007.

Lee, H.S., and N.H. Younan. "DTM extraction of LiDAR returns via adaptive processing." IEEE Transactions on Geoscience and Remote Sensing, 2003: 2063-2069.

Lohmann, P, A Koch, and M Schaeffer. "Approaches to the filtering of laser scanner data." International Archive of Photogrammetry and Remote Sensing, 2000: 540-547.

Lutz, E., T. Geist, and J. Stötter. "Investigations of airborne laser scanning signal intensity on glacial surfaces – utilizing comprehensive laser geometry modeling and orthophoto surface modeling." International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2003: 143-148.

Mallat., S. A Wavelet Tour of Signal Processing. San Diego , California: 2nd edition Academic, 1999.

Marcus, and Fonstad. "Optical remote mapping of rivers at sub-meter resolutions and watershed extents." Earth Surface Processes and Landforms, 2008: 4-24.

Meng, X, L Wang, and N Currit. "Morphology-based building detection from airborne LIDAR data." Photogrammetric Engineering and Remote Sensing, 2009: 427-442.

Mohammadzadeh, A, and V M J Zoej. "A State of Art on Airborne Lidar Application in Hydrology and Oceanography: A Comprehensive Overview." The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2008.

Moore, Andrew. W. Efficient Memory based Learning for Robot Control. Computer Laboratory, University of Cambridge: Carnegie Mellon University, 1991.

Moore., I D, R Grayson., and A LADSON. "Digital terrain modelling: a review of hydrological, geomorphological, and biological applications." Hydrological Processes, 1991: 3-30.

Mrigot, Q., M. Ovsjanikov, and L. J. Guibas. "Robust voronoi-based curvature and feature estimation." In. Symposium on Solid and Physical Modeling, 2009.

NRC. Elevation Data for Floodplain Mapping, Committee on Floodplain Mapping Technologies,National Research Council. National Academies Press, 2007.

Pajak, M J, and S Leatherman. "The high water line as shoreline indicator." Journal of Coastal Research, 2002: 329-337.

Passalacqua, Paola, Do Trung Tien, Foufoula Georgiou Efi, Sapiro Guillermo, and Dietrich William E. "A geometric framework for channel network extraction from lidar Nonlinear diffusion and geodesic paths." Journal of Geophysical Research, 2010.

Perona, P., and J. Malik. "Scale-space and edge detection using anisotropic diffusion." IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990: 629–639.

Pfeifer, N., T. Reiter, C. Briese, and W. Rieger. "Interpolation of high quality ground models from laser scanner data in forested areas." International Archive of Photogrammtery, Remote Sensing and Spatial Information Science, 1999: 31-36.

Populus, Jacques. Coastal Wiki: Airborne Laser Hydrography. March 31, 2009. http://www.coastalwiki.org/coastalwiki/Use_of_Lidar_for_coastal_habitat_mapping.

Puetz, Angela M., R. Chris Olsen, and Brian Anderson. Effects of LIDAR point density on bare earth extraction and DEM creation. Florida: SPIE, 2009.

Rabbania, T, van den F.A. Heuvelb, and G Vosselmanc. "Segmentation of point clouds using smoothness constraint." International Archive of Photogrammetry, Remote Sensing and Spatial Information Science, 2006: 248-253.

Rutzinger, M, B Höfle, M Hollaus, and N Pfeifer. "Object-based point cloud analysis of full-waveform airborne laser scanning data for urban vegetation classification." Sensors, 2008.

Sampath, Aparajithan, and Jie Shan. "Building boundary tracing and regularization from airborne lidar point clouds." Photogrammetric Engineering and Remote Sensing, 2007: 805-812.

Säynäjoki, R, M Maltamo, and T K Korhonen. Forest inventory with sparse resolution Airborne Laser Scanning data a literature review. Vantaa, Finland: Finnish Forest Research Institute, 2008.

Schimdt., J, and R Dikau. "Extracting geomorphometric attributes and objects from digital elevation models semantics, methods, future needs." GIS for Earth Surface Systems—Analysis and Modelling of the Natural Environment, 1999: 153-173.

Schmid, Keil, Kirk Waters, and Lindy Dingerson. Lidar 101: An Introduction Lidar Technology, Data, and Applications. Charleston, SC: NOAA Coastal Services Center, 2008.

Schmidt, J., S. I. Evans, and J. Brinkmann. "Comparison of polynomial models for land surface curvature calculation." INTernational Journal of Geographical Information Science, 2003: 797–814.

Shan, J., and A. Sampath. "Urban DEM generation from raw LiDAR data: a labeling algorithm and its performance." Photogrammetric Engineering and Remote Sensing (Photogramm. Eng. Remote Sens.), 2005: 217-226.

Sithole, G, and G Vosselman. "Filtering of airborne laser scanner data based on segmented point clouds." Laser scanning, 2005: 66-71.

Smith, Rebecca.A, J.L Irish, and Michael.Q Smith. "Airborne Lidar and airborne hyperspectral imagery: a fusion of two proven sensors for improved hydrographic surveying." Proceedings of Canadian Hydrographic Conference 2000. Montreal Canada, 2000. 10.

TNRIS. TNRIS. 2009. http://www.tnris.org/DataCatalog/Lidar.aspx.

Toth, C. "Airborne Lidar Technology: The State-of-art and future trends." Latin American Remote Sensing Week Regional ISPRS Conference. Chile, 2010.

Tóvári, D, and N Pfeifer. "Segmentation based robust interpolation–a new approach to laser filtering." International Archive of Photogrammetry, Remote Sensing and Spatial Information Science, 2005: 79-84.

Tucker, E. Gregory., Filippo. Catani, Andrea Rinaldo, and L. Rafael Bras. "Statistical analysis of drainage density from digital terrain data." Geomorphology, 2001: 187–202.

Vivoni, E.R., V.Y. Ivanov, R.L. Bras, and D. Entekhabi. "Generation of triangulated irregular networks based on hydrological similarity." Journal of Hydrologic Engineering, 2004: 288.

Xuelian, Nate, and Kaiguang. "Ground Filtering Algorithms for Airborne LiDAR Data: A Review of Critical Issues." Remote Sensing, 2010: 833-860.

Zhang, K, S Chen, D Whitman, and M Shyu. "A progressive morphological filter for removing nonground measurements from airborne LiDAR data." IEEE Transactions on Geoscience and Remote Sensing, 2003: 872-882.

Zheng, S., W. Shi, J. Liu, and G. Zhu. "Facet-based airborne light detection and ranging data filtering method." Optical Engineering, 2007.