DIGITAL TERRAIN MODEL EXTRACTION FROM HIGH-RESOLUTION POINT CLOUDS BY USING A MULTI-RESOLUTION PLANARITY-BASED APPROACH

A, THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

YASİN KOÇAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
GEODETIC AND GEOGRAPHIC INFORMATION TECHNOLOGIES

SEPTEMBER 2021

Approval of the thesis:

DIGITAL TERRAIN MODEL EXTRACTION FROM HIGH-RESOLUTION POINT CLOUDS BY USING A MULTI-RESOLUTION PLANARITY-BASED APPROACH

submitted by **YASİN KOÇAN** in partial fulfillment of the requirements for the degree of **Master of Science in Geodetic and Geographic Information Technologies, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Zuhal Akyürek
Head of the Department, **Geodetic and Geographic Inf. Tech.**

Assoc. Prof. Dr. Uğur Murat Leloğlu
Supervisor, **Geodetic and Geographic Inf. Tech.**

Examining Committee Members:

Prof. Dr. Zuhal Akyürek
Dept. of Civil Eng., METU

Assoc. Prof. Dr. Uğur Murat Leloğlu
Geodetic and Geographic Information Technologies, METU

Assoc. Prof. Dr. Onur Pekcan
Dept. of Civil Eng., METU

Assoc. Prof. Dr. Fatih Nar
Dept. of Computer Eng., Ankara Yıldırım Beyazıt University

Assoc. Prof. Dr. Ali Özgün Ok
Dept. of Geomatics Eng., Hacettepe University

Date: 09.09.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Yasin Koçan

Signature :

ABSTRACT

DIGITAL TERRAIN MODEL EXTRACTION FROM HIGH-RESOLUTION POINT CLOUDS BY USING A MULTI-RESOLUTION PLANARITY-BASED APPROACH

Koçan, Yasin
Master of Science, Geodetic and Geographic Information Technologies
Supervisor: Assoc. Prof. Dr. Uğur Murat Leloğlu

September 2021, 128 pages

Digital Elevation Model (DEM) is a mathematical representation of the elevation of the Earth's surface. There are two types of DEM, namely Digital Surface Model (DSM) and Digital Terrain Model (DTM). DSM contains natural (bare-ground, trees, bushes, etc.) and artificial above-ground objects (buildings, vehicles, powerlines, etc.), while DTM covers only the bare earth without anything on it. Above-ground objects need to be removed to extract the DTM, which is a tedious task. This thesis proposes an algorithm that extracts DTM from aerial point clouds using a robust multi-resolution planarity-based divide-and-conquer algorithm. In this approach, the problem is handled in few simple steps rather than trying to solve the problem at once. The approach contains different planarity checks to get rid of nonplanar above-ground objects, segmentation step to find rough ground points, and an interpolation step to obtain the final DTM. In this thesis, ground points are assumed planar, and planar patches are detected as ground candidates. First, approximate planarity values are calculated by using neighboring points. This helps to eliminate most of the above-ground objects such as vehicles, trees, posts, etc. Nevertheless, since the building facades and roofs are also planar, a second planarity check is needed in different resolutions. For this purpose, the grid planarity values are checked. The grids that do not fit a plane within the given threshold are marked as nonplanar. The second planarity check helps to get rid of the building facades and the vertical planes. After removing building facades, getting benefit from the sparsity between ground candidates and the roof points, a region growing segmentation is utilized to segment the remaining ground candidates for rough ground surface calculation. The segments far from the rough ground surface are omitted. By doing so, the roof points can be eliminated. Lastly, the ground points are interpolated to obtain the resulting DTM raster. Although the input point cloud is already classified as ground and non-ground, it has some errors. The input point cloud is used to create a DTM; then, the resulting DTM is manually edited to use it as a ground truth. The accuracy assessment is done on interpolated DTM rasters. Using a manually corrected ground truth, Root Mean Square Error (RMSE) is calculated for two datasets with different characteristics having 1.00 m and 2.20 m spatial resolutions. The results are compared with two existing DTM extraction algorithms and RMSE values are close to these solutions. The RMSE values are 0.25 m and 0.70 m, respectively. Results indicate that an accurate DTM extraction is possible using a combination of only planarity values.

Keywords: Digital terrain model, digital elevation model, planarity, covariance features, Principal Component Analysis, point cloud

ÖZ

YÜKSEK ÇÖZÜNÜRLÜKLÜ NOKTA BULUTLARINDAN ÇOKLU ÇÖZÜNÜRLÜKLÜ DÜZLEMSELLİK TABANLI YAKLAŞIM KULLANARAK DİJİTAL ARAZİ MODELİ ÇIKARIMI

Koçan, Yasin
Yüksek Lisans, Jeodezi ve Coğrafi Bilgi Teknolojileri
Tez Yöneticisi: Doç. Dr. Uğur Murat Leloğlu

Eylül 2021, 128 sayfa

Sayısal Yükseklik Modeli, Dünya yüzeyinin yükseklik verilerinin matematiksel bir temsilidir. Sayısal Yüzey Modeli (SYM) ve Sayısal Arazi Modeli (SAM) olmak üzere iki tür sayısal yükseklik modeli vardır. SYM, doğal (zemin, ağaçlar, çalılar, vb.) ve yapay yer üstü nesneleri (binalar, araçlar, elektrik hatları vb.) içerirken, SAM, üzerinde hiçbir şey olmayan, sadece arazi modelini temsil eder. Meşakkatli bir görev olan SAM'ı çıkarmak için yer üstü nesnelerin kaldırılması gerekir. Bu tez, sağlam bir çoklu-çözünürlüklü düzlemsellik tabanlı böl ve yönet algoritması kullanarak SAM'ı havadan toplanmış nokta bulutlarından çıkaran bir algoritma önermektedir. Yaklaşım, düzlemsel olmayan yer üstü nesnelerden kurtulmak için farklı düzlemsellik kontrolleri, kaba zemin noktalarını bulmak için bölütleme adımı ve nihai DTM'yi elde etmek için bir enterpolasyon adımı içerir. Bu tezde, yer noktaları düzlemsel kabul edilmiş ve düzlemsel noktaların yer aday noktaları olduğu tespit edilmiştir. İlk olarak, komşu noktalar kullanılarak yaklaşık düzlemsellik değerleri hesaplanmaktadır. Bu, algoritmanın araçlar, ağaçlar, direkler vb. gibi yer üstü nesnelerin çoğunu ortadan kaldırmasına yardımcı olur. Fakat, bina cepheleri ve çatılar da düzlemsel olduğundan, farklı çözünürlükte ikinci bir düzlemsellik kontrolüne ihtiyaç duyulmaktadır. Bunun için ızgara düzlemsellik değerleri kontrol edilmektedir. Verilen eşik içinde bir düzleme uymayan ızgaralar, düzlemsel olmayan olarak işaretlenir. İkinci düzlemsellik kontrolü, bina cephelerinden ve düşey düzlemlerden kurtulmaya yardımcı olmaktadır. Bina cepheleri kaldırıldıktan sonra, zemin adayları ile çatı noktaları arasındaki seyreklikten yararlanılarak, kalan zemin adaylarının kaba zemin yüzeyi hesabı için bölütlenmesi için bölge büyüyen bir bölütleme kullanılır. Kaba zemin yüzeyinden uzaktaki parçalar atlanmıştır. Bu sayede çatı noktaları elenmektedir. Son olarak, elde edilen SAM ızgarasını elde etmek için zemin noktaları enterpolasyona tabi tutulur. Girdi olarak kullanılan nokta bulutu önceden sınıflandırılmış olmasına rağmen, yer noktalarını etiketlemede bazı hatalara sahiptir. Bir SAM oluşturmak için girdi olarak kullanılan nokta bulutu kullanılmıştır, ardından elde edilen SAM, referans olarak kullanılmak için el ile düzenlenmiştir. Doğruluk değerlendirmesi, enterpolasyonlu SAM ızgaraları üzerinde yapılmıştır. Elle düzeltilmiş bir referans kullanarak, 1.00 m ve 2.20 m uzamsal çözünürlüğe sahip farklı özelliklere sahip iki veri kümesi için Kök Ortalama Kare Hatası (RMSE) hesaplanmıştır. Sonuçlar, mevcut iki algoritma ile karşılaştırılmış ve yakın değerler elde edilmiştir. RMSE değerleri sırasıyla 0.25 m ve 0.70 m büyüklüğündedir. Sonuçlar, yalnızca düzlemsellik değerlerinin bir kombinasyonu kullanılarak doğru bir DTM çıkarımının mümkün olduğunu göstermektedir.

Anahtar Kelimeler: Sayısal yükseklik modeli, Sayısal arazi modeli, Düzlemsellik, Kovaryans özellikler, Temel bileşenler analizi

I dedicate this thesis to my family.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my supervisor, Assoc. Prof. Dr. Uğur Murat Leloğlu for his valuable insights, suggestions, guidance, and criticism throughout this research period. Without his support, I would not be where I am now.

I owe my deepest gratitude to my family for their endless support. Their trust in me made this thesis easier for me to finish.

I would also like to thank Pix4D team for making a public dataset for researchers to work with. I would also like to thank Zibumi R&D Center for their support. Without their support, it would be hard for me to finish my thesis while working there.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
5G	Fifth-generation wireless
AI	Artificial Intelligence
ALS	Airborne Laser Scanning
ASPRS	American Society for Photogrammetry and Remote Sensing
CNN	Convolutional Neural Network
CV	Computer Vision
DEM	Digital Elevation Model
DSM	Digital Surface Model
DTM	Digital Terrain Model
EO	Exterior Orientation
GHT	Generalized Hough Transform
GIS	Geographic Information Systems
GPS	Global Positioning System
GPU	Graphic Processor Unit
GSD	Ground Sampling Distance
HAG	Height Above Ground
IDW	Inverse Distance Weighting
IO	Interior Orientation
LIDAR	Light Detection and Ranging or Laser Imaging Detection and Ranging
NDVI	Normalized Difference Vegetation Index
PCA	Principal Component Analysis

PCL	Point Cloud Library
PDAL	Point Data Abstraction Library
PHM	Photogrammetry
RANSAC	Random Sample Consensus
RGB	Red Green Blue
RMSE	Root Mean Square Error
SfM	Structure-from-Motion
SAR	Synthetic Aperture Radar
TIN	Triangular Irregular Network
UAV	Unmanned Aerial Vehicle

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

Digital elevation model is a three-dimensional (3D) representation of elevation of the surface of the Earth, which is available in various formats such as raster format (consisting of a matrix of pixels where each pixel contains the elevation value) or Triangular Irregular Network (TIN) models. As a subset of the Digital Elevation Model, Digital Surface Model (DSM) and Digital Terrain Model (DTM) can be considered. DSM represents the Earth's surface and all the non-stationary objects above ground. These objects can be categorized as vegetation, building, city furniture, vehicles, power lines, etc. On the other hand, DTM covers only the bare earth without any object on it. However, since the sensing devices cannot capture the earth crust beneath these objects, the initial product as an elevation model from photogrammetric survey is actually a DSM. Some sensing methods can penetrate partially through the vegetation like LIDAR (Light Detection and Ranging or Laser Imaging Detection and Ranging) sensors. However, the buildings and some other objects still need to be removed from the DSM to obtain the DTM. Depending on the project's scope and desired spatial resolution, there are different types of remote sensing methods for DTM extraction, namely, Synthetic Aperture Radar (SAR), LIDAR, and photogrammetric methods.

Nevertheless, regardless of the sensing method, DTM extraction is a tedious task. There are different areas that DSMs, and DTMs are dominantly used. DSM is commonly used in telecommunication, fifth-generation wireless (5G) planning, urban planning, aviation, and viewshed analysis. On the other hand, DTM is frequently used in hydrology, soil research, land use planning, etc., where the bare-earth representation is crucial.

Different approaches in the literature extract DTM from DSM; however, due to massive variability in the earth surface, it is not easy to find a robust strategy that handles different terrain types. Although DTMs are commonly used, it is still hard to find a method that does not depend on user input, the terrain type (flat, steep, etc.), or object types (houses, trees, etc.). There is a need for a robust pipeline that works on photogrammetric points which can handle different terrain types with minimizing the user input need.

1.2 The objective of the Thesis

This thesis aims to detect the ground points from the point cloud generated from photogrammetric point clouds by exploiting the geometry of the coordinate values (X, Y, and Z). The study is based on Eigen-derived covariance features to obtain the ground points from which a Digital Terrain Model (DTM) can be extracted. Color information (R, G, B) may not be available for all point clouds, so the proposed algorithm does not depend on color information. We want this algorithm to work everywhere except in dense urban areas without having problems while filtering buildings, trees, vehicles, etc. To summarize, the main objectives of the thesis are:

1. To classify the ground points from the input point cloud as ground and non-ground using only geometric information.
2. To propose a robust algorithm that works in different terrain types (flat, steep, etc.)
3. To interpolate the gaps resulting from non-stationary objects (buildings, vehicles, vegetation, powerlines, etc.) to obtain a continuous DTM raster output.

1.3 Contribution of the Thesis

The following are the contributions of this thesis to the state-of-the-art:

1. Having a robust method that works in multi-resolution planarity approach,
2. Using only a planarity-based approach to get rid of the non-ground features without needing other covariance features,
3. Using divide-and-conquer strategy to solve the problem in simpler steps.

1.4 Structure of the Thesis

This thesis is organized as follows:

CHAPTER 1 introduces the significance and necessity of the digital terrain models. The research problem, objectives, contribution, and importance of the study are also described in this chapter.

CHAPTER 2 presents the background and literature review of the digital photogrammetry, digital elevation models, point clouds, description of principal component analysis and covariance features, the principles of Hough Transform and, common surface interpolation techniques. Various point cloud classification and digital terrain model extraction studies are also discussed in this chapter.

CHAPTER 3 gives information about the data characteristics and methodology used in this thesis. Ground point classification, terrain extraction processes and the proposed algorithm steps are explained in this chapter.

CHAPTER 4 evaluates the quantitative and qualitative results. The algorithm performance is compared with two available DTM extraction solution.

CHAPTER 5 presents a summary and results of the thesis are discussed in this chapter. Also, possible application of this algorithm and proposals for the future studies are given.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

DEMs are simply the digital representation of the Earth's surface. DEMs and their derivatives provide valuable information about the terrain. They are used widely in different studies such as hydrology, geomorphology, vegetation analysis, change detection, geology, robotics, city planning, solar irradiance calculation, archeology, volumetric calculation, etc., as shown in Table 1. DEMs can be categorized into two, namely, DSM and DTM. DSM is the DEM that includes objects such as buildings, vegetation. On the other hand, DTM represents the bare-earth surface (Gupta, 2018).

Table 1: DEM derivatives and their use (Gupta, 2018)

DEM Derivatives		
DEM Derivative	**Description**	**Use**
Shaded Relief	Represents illumination conditions of a region based on the sun location	Aesthetic visualization, provides hillshade visuals, etc.
Slope	Represents the slope map of the region	Hydrology, geomorphometry measurements, infrastructure planning (road planning etc.)
Aspect	Representation of the slope orientation. Each cell value indicates the slope direction	Hydrology, geo-morphometry measurements, infrastructure planning, vegetation analysis, solar irradiation analysis, etc.

Table 1 (continued)

DEM Derivative	**Description**	**Use**
Viewshed	Allows visibility analysis from specific points. There are different types of viewshed analysis: 1. Line of Sight 2. Area of Sight 3. Dome -shaped viewshed	Often used in archeology (Wheatley & Gillings, 2000), urban planning (Danese, Nolè, & Murgante, 2009), forestry (Domingo-Santos, de Villarán, Rapp-Arrarás, & de Provens, 2011), military (VanHorn & Mosurinjohn, 2010), and impact assessment (Howes & Gatrell, 1993)

The reconstruction of precise surfaces from unorganized point clouds obtained from LIDAR or photogrammetry is a complicated task that is not entirely solved and troublesome on inadequate, noisy, and sparse data (Remondino, 2003). Although it is a tedious task to obtain DTM, there are different approaches and solutions that extract DTM from DSM point clouds. Due to variations on the Earth's surface, some may perform better in specific terrain types, whereas some are fast in certain cases.

2.2 Digital Photogrammetry

Photogrammetry, as the name implies, is the "science of measuring via images." To compute distances, areas, or anything similar, one must first obtain the object (terrain) coordinates of any point in the shot, from which one can then calculate geometric data or construct maps (Linder, 2009).

Stereoscopic photogrammetry is based on trying to recreate, using two cameras, what we perceive with our eyes. The differential in the object positions in the photographs is the parallax of the item when the image from the left to the right camera are compared. Parallax causes the observer to perceive the distance from the object (Wang, Zhen, Zhang, & Sato, 2016).

Structure-from-Motion (SfM) is based on the same fundamentals as stereoscopic photogrammetry, in which 3D structure can be generated from a set of overlapping, offset images (Westoby, Brasington, Glasser, Hambrey, & Reynolds, 2012). First, these overlapped images are matched and connected with tie points, then sparse and dense 3D point clouds can be generated automatically by using photogrammetry software packages that offer different types of techniques such as intensity-based, feature-based, and relational image matching.

Thanks to enchantments in civilian UAV technology, Computer Vision (CV), image processing techniques, and Artificial Intelligence (AI), photogrammetry software suites allow users to create point clouds, orthophotos, and rasterized DEMs from UAV imagery (Woodget, Austrums, Maddock, & Habit, 2017). Camera pose and scene geometry are simultaneously rebuilt by automatically identifying the corresponding features for multiple images. These aspects are tracked from image to image to allow the camera locations and the object coordinates to be first estimated and repeatedly improved using non-linear least-squares minimization (Snavely, Seitz, & Szeliski, 2008). Most commercial UAVs have built-in GPS sensors that can be used to keep track of image coordinates. Two sets of factors must be considered while determining the interior orientation (IO). The first comprises the camera's geometric parameters, such as the principal distance and the principal point's coordinates. The parameters that characterize systematic errors are included in the second set (such as distortions or film deformations). The goal of the exterior orientation (EO) is to specify the camera's location (X, Y, Z, Omega, Phi, and Kappa) as illustrated in Figure 1 and attitude at the time of exposure (Grussenmeyer & Al Khalil, 2002).

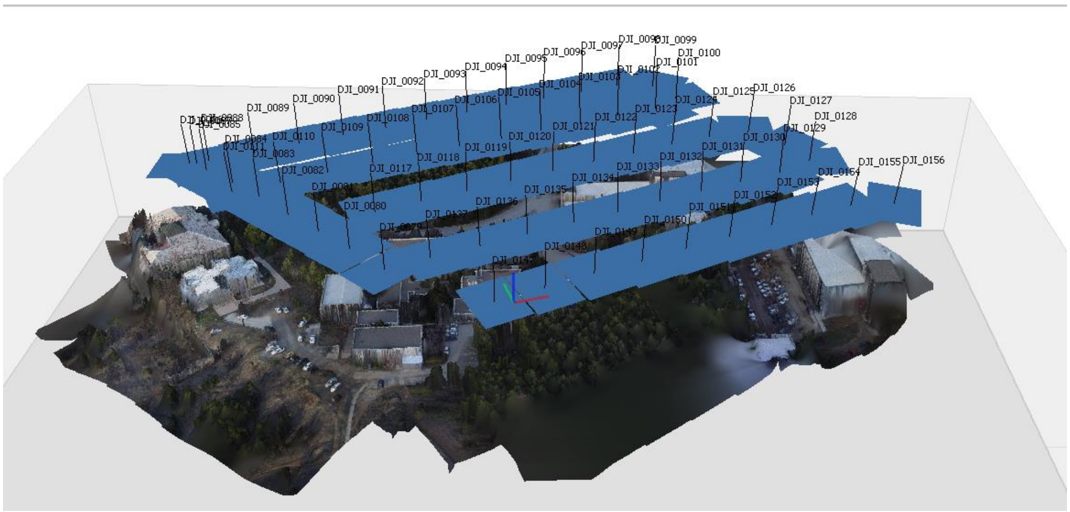


Figure 1: DSM and stereo aerial photographs showing a small portion of Middle East Technical University campus (Produced by author)

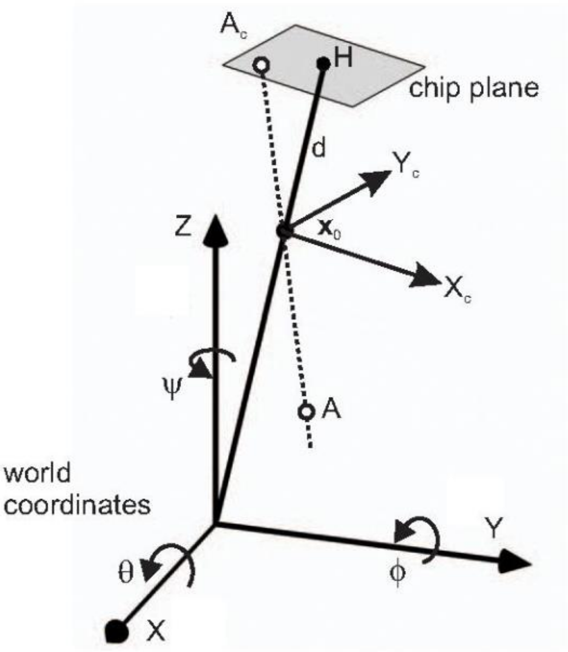


Figure 2: The perspective image of a point *A*, which produces an image point A_c, on the chip plane. The rotation is about the *x*, *y*, and *z*-axis of the world coordinate system is defined by θ, ϕ, and ψ. (Putz & Zagar, 2008) © 2008 IEEE

2.3 Digital Elevation Models

A DTM is a set of discrete points with unique height values across two-dimensional (2D) points that approximates a section or the entire continuous terrain surface (Hirt, 2014). Raster maps with a single elevation value allocated to each pixel are known as DEMs. Some vertical terrain characteristics cannot be described using DEMs, depending on the spatial resolution and terrain features. Although some applications can represent these raster maps as a 3D scene, they are not truly three-dimensional since they cannot represent cliff-like structures where sudden elevation change occurs. As a result, DEMs are classified as "2.5D" rather than "3D" (Longley, Goodchild, Maguire, & Rhind, 2005). DTM and DSM are commonly used together with DEMs to define the model characteristics. The term DTM refers to a DEM of the Earth's terrain in its purest form, i.e., the bare ground; conversely, DSM refers to ground items such as houses, trees, vehicles, and powerlines (Gupta, 2018). DEM is generally used as an inclusive term for elevation models from remote sensing, including DSM and DTM (Hirt, 2014). According to (Gupta, 2018), the acquisition of DTM can be in different ways and from various sources such as:

1. Ground surveys
2. Elevation contour maps
3. Conventional aerial photogrammetry
4. Digital aerial photogrammetry
5. UAV imager
6. SAR
7. LIDAR

This thesis focuses on the DTM generation from the photogrammetric point cloud. Thus, these methods are explained in detail in this chapter.

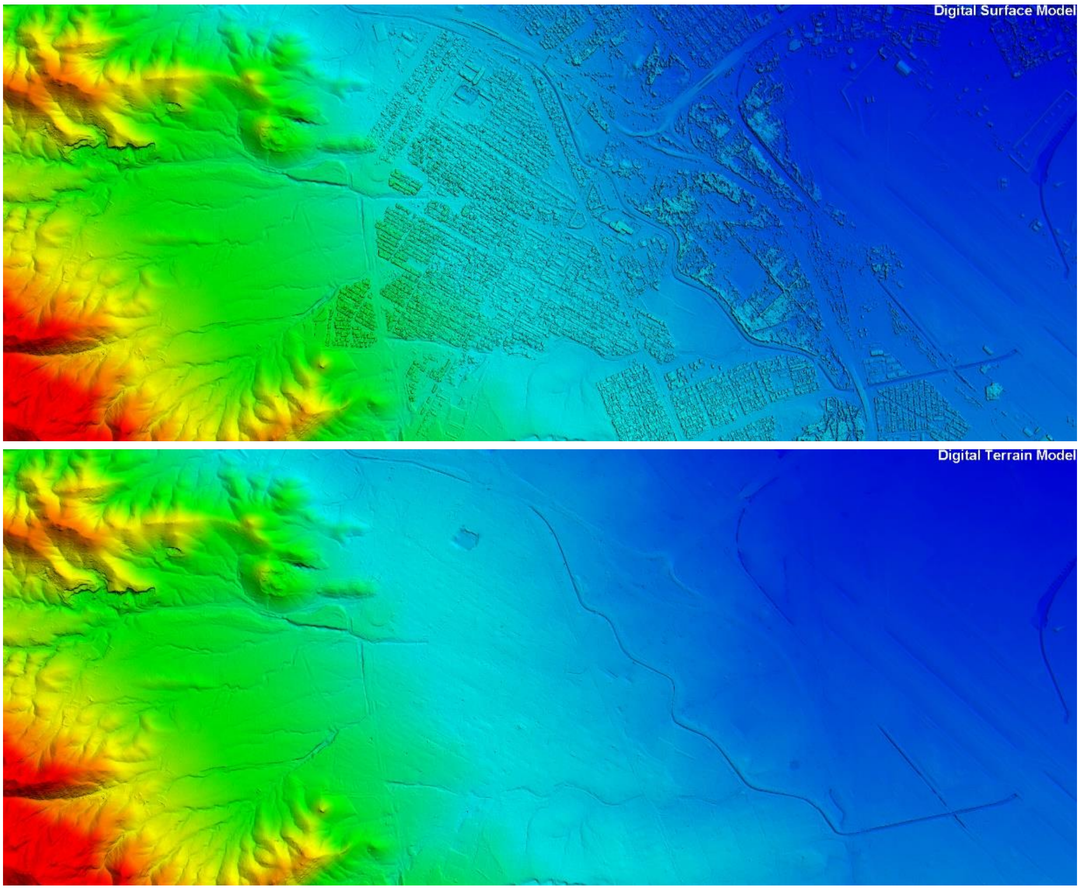


Figure 3: DTM (Up), DSM (Bottom) data acquired from Airborne Laser Scanning (Produced by author)

2.4 Point Cloud

A point cloud is a collection of data points in 3D space that represents an object or a surface. Each point has a set of Cartesian coordinates (X, Y, Z). Some additional fields such as color data (Red, Green, Blue, and Near-Infrared), Intensity, Return Number, Number of Returns (given pulse), Scan Direction Flag, Edge of Flight Line, Classification, etc. can be given as a set of features for each point in the cloud. In this thesis, only X, Y, Z values are used as input to extract the DTM.

LIDAR is a direct measurement in which a laser light pulse physically hits a surface, and the return time of the reflection is measured. Aerial photogrammetry uses overlapping images over a region and adequate ground control points to rebuild the landscape in an accurate 3D model using photos collected by airborne or space-borne platforms. These overlapping images need to be processed by a photogrammetry software to create surface models. Both approaches offer advantages and disadvantages, depending on the scale and scope of the project. Since some LIDAR sensors can penetrate through leaves which can provide ground observations, especially in forested areas, they are advantageous for DTM extraction. However, that does not mean that aerial photogrammetry is not applicable to these regions. Although some LIDAR solutions include cameras that can collect color information in addition to geometric data (X, Y, and Z values), this is expensive or uncommon in the remote sensing industry. On the other hand, aerial photogrammetry provides color information (multi-spectral) that may be utilized to infer information about a region. For example, when identifying vegetation, the Normalized Difference Vegetation Index (NDVI) provides valuable information for detecting vegetated area, as shown in Figure 4. Because of its accessibility and low cost, aerial photogrammetry is gaining popularity. In Table 2, the photogrammetric point cloud and LIDAR point cloud are compared in terms of cost, accuracy, process time, and additional information.

Figure 4: RGB and NDVI Colored Satellite Image (Produced by author)

Table 2: Photogrammetry and ALS comparison (Höhle & Potucková, 2011)

DTM collection method	**Vertical accuracy**			**Terrain type**			
	$< \mathbf{0.1}$ **m**	$(\mathbf{0.1} - \mathbf{1.0})$**m**	$> \mathbf{1.0}$ **m**	**Open terrain**	**Low vegetation**	**Forested**	**Built-up areas**
PHM	$+ *$	$+$	$+$	$+$	$-$	$-$	$+/-$
ALS	$+$	$+$	$+$	$+$	$+/- **$	$+$	$+$
DTM collection method	Strongly influenced by atmospheric effects		Holes in data		Large area (> 10 km^2)		
PHM	+		+		+		
ALS	-		+		+		

* Achievable in low flying heights and small pixel size (GSD)

** Vegetation density-dependent

*** Wavelength dependent

ALS - airborne laser scanning PHM – photogrammetry

2.5 Principal Component Analysis and Covariance Features

Large datasets are becoming more prevalent, yet they might be challenging to interpret. PCA is a way to reduce the dimensionality of such datasets, enhancing interpretability while minimizing information loss. It accomplishes this by generating new uncorrelated variables with maximized variance. After finding the covariance matrix of the data, the problem becomes solving an eigenvalue/eigenvector problem (Hubert, Rousseeuw, & Vanden Branden, 2005). The principal components (PCs) are linear combinations of the original variables classified by the variability of data. The PCs, therefore, describe the data variances in multiple orthogonal directions (Nurunnabi, Belton, & West, 2012). With a principal component analysis, the geometric properties of an area, such as linearity and planarity describing the characteristics of the regions, can be estimated as well (Rovers, de Vreede, Rook, Psomadaki, & Nagelkerke, 2015). A variety of 3D features can be determined easily by describing fundamental geometric features of a considered 3D neighborhood (specified radius and points inside this sphere or given number of nearest neighbors (k-NN)) (Weinmann, Urban, Hinz, Jutzi, & Mallet, 2015). For a given point $\boldsymbol{p}$ and its neighborhood $\boldsymbol{P}$, the covariance features shown in Table 3 can be calculated based on eigenvalues $\boldsymbol{\lambda_1}$, $\boldsymbol{\lambda_2}$, $\boldsymbol{\lambda_3}$ and eigenvectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$ of the covariance matrix $C = \frac{1}{k}\sum_{i \in P}(p_i - \bar{p})(p_i - \bar{p})^T$ (Hackel, Wegner, & Schindler, 2016).

Table 3: Geometric features based on eigenvalues of the local structure covariance matrix and corresponding eigenvectors (Hackel, Wegner, & Schindler, 2016)

Covariance Feature Name	**Covariance Feature Formula**
Sum	$\lambda_1 + \lambda_2 + \lambda_3$
Omnivariance	$(\lambda_1 \cdot \lambda_2 \cdot \lambda_3)^{\frac{1}{3}}$
Eigenentropy	$-\sum_{i=1}^{3} \lambda_i \cdot \ln(\lambda_i)$
Anisotropy	$(\lambda_1 - \lambda_3)/\lambda_1$
Planarity	$(\lambda_2 - \lambda_3)/\lambda_1$

Table 3 (continued)

Covariance Feature Name	**Covariance Feature Formula**
Linearity	$(\lambda_1 - \lambda_2)/\lambda_1$
Surface Variation	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$
Sphericity	λ_3/λ_1
Verticality	$1 - n_Z$

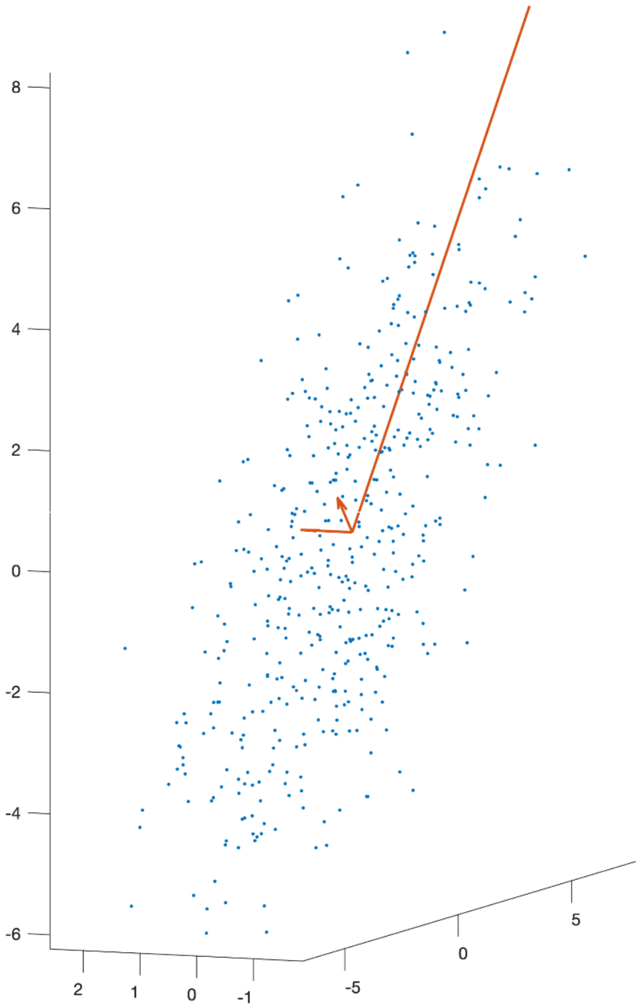


Figure 5: PCA on linear point distribution

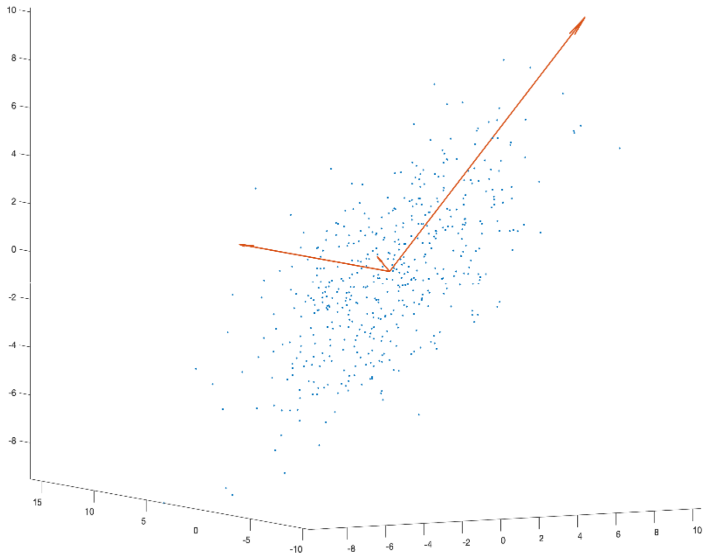


Figure 6: PCA on planar point distribution

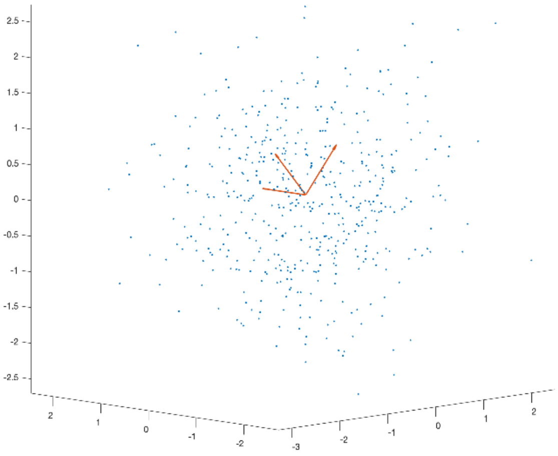


Figure 7: PCA on spherical point distribution

Linearity, planarity, and sphericity are used to determine whether a point is labeled as 1D (linear), 2D (planar), or 3D (volume). For example, if the linearity value of a point is high, it most likely belongs to a linear feature such as a building edge or a pole, and hence the points nearby are mostly distributed along a line. Similarly, the planarity value describes the smoothness of the surfaces and is high for features like building facades (Waldhauser, et al., 2014). Omnivariance is used to explain how the points spread inhomogeneously for a given 3D volume and high for inhomogeneous point distributions in this volume (Waldhauser, et al., 2014). Anisotropy is helpful to distinguish oriented and non-oriented objects and is higher if the eigenvalues differ a lot (Oude Elberink & Maas, 2000). Surface variation defines how closely a smooth planar patch forms a local neighborhood and is low for planar regions (Wang & Feng, 2015). The eigenentropy measures the order or disorder of the 3D points within the 3D volume (Weinmann, Urban, Hinz, Jutzi, & Mallet, 2015). Verticality value is high both for vertical planes and vertical linear features such as building facades and poles. Verticality (V) is calculated as $1 - |\langle [001], \boldsymbol{e}_3 \rangle|$, in other words, $V = 1 - n_Z$ where n_Z represents the third component of the normal vector n (Demantké, et al., 2011). Lastly, sum of eigenvalues defines the total variance. If the corresponding eigenvalue is small compared to the summation, it may not be essential (Mark, 2010). For demonstration purposed, these features are calculated for a building point cloud for a 0.5 m radius neighborhood; for each point, the points in its 0.5 m neighborhood are used for covariance matrix estimation. In Figure 8, an RGB-colored point cloud is illustrated. Verticality (Figure 9), sphericity (Figure 10), surface variation (Figure 11), linearity (Figure 12), planarity (Figure 13), anisotropy (Figure 14), eigenentropy (Figure 15), omnivariance (Figure 16), the sum of eigenvalues (Figure 17), and values are shown to illustrate how these features change on a single building example. For example, in Figure 9, it is observed that the building facades have high verticality values. Because the surface normal is horizontal, their inner product with the vertical vector is small, so the verticality values are significant. In Figure 10, it is noticed that the sphericity value is high for complex 3D structures and close to zero in planar and linear regions. Like sphericity, Figure 11 shows a similar behavior because the planar regions have low surface deviation, whereas the complex structures show the opposite. In Figure 12, the linearity value is significant for linear formations such as building edges, roof edges, post-like structures, and object boundaries. In Figure 13, large planarity values are observed at ground points, planar regions, building roofs, and smooth building facades. In Figure 14, the anisotropy value is considerable for complex structures and almost zero around the planar areas. In Figure 15, since the eigenentropy is a measure of order/disorder of the points, it is significant for the complex 3D structures and relatively low for planar regions. Similarly, in Figure 16, the omnivariance value is high along the lines where the roof planes intersect because the point distribution is relatively inhomogeneous in these regions that contain multiple planes. In Figure 17, the higher sum values represent the significant variance.

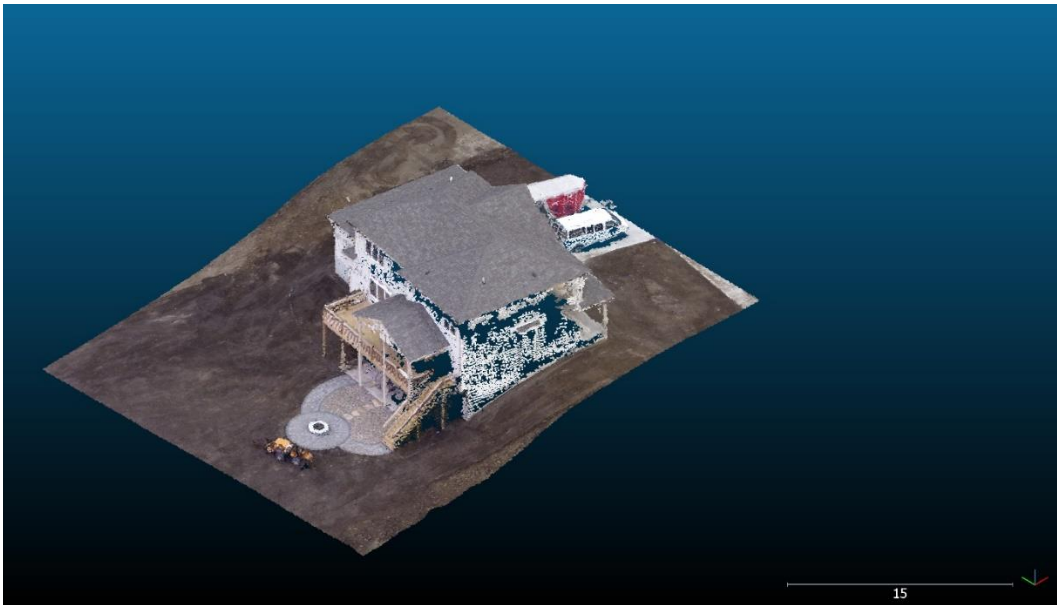


Figure 8: RGB-colored point cloud of a building

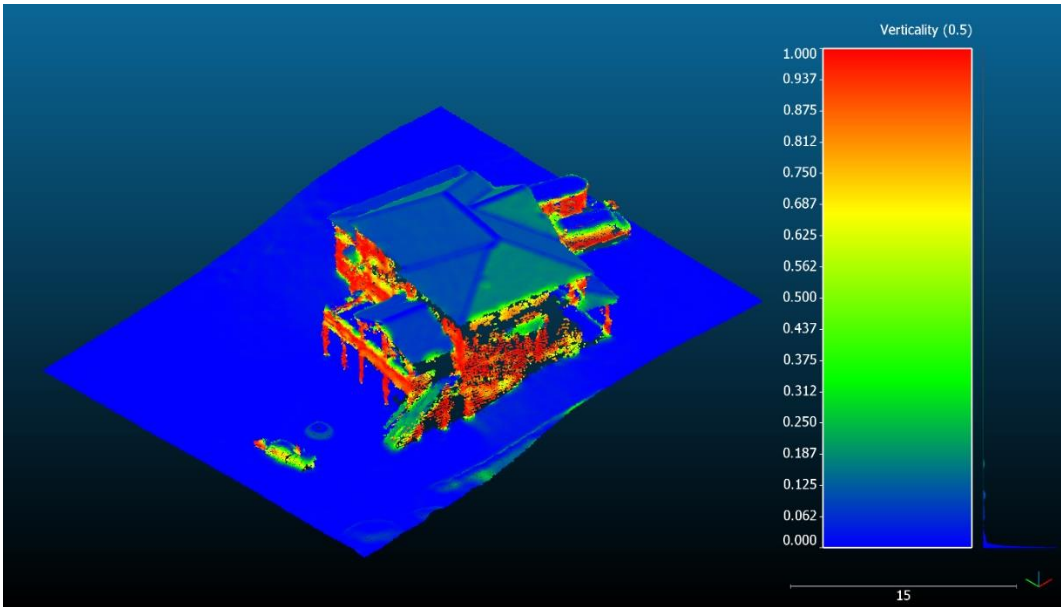


Figure 9: Verticality of a building point cloud in 0.5 m radius

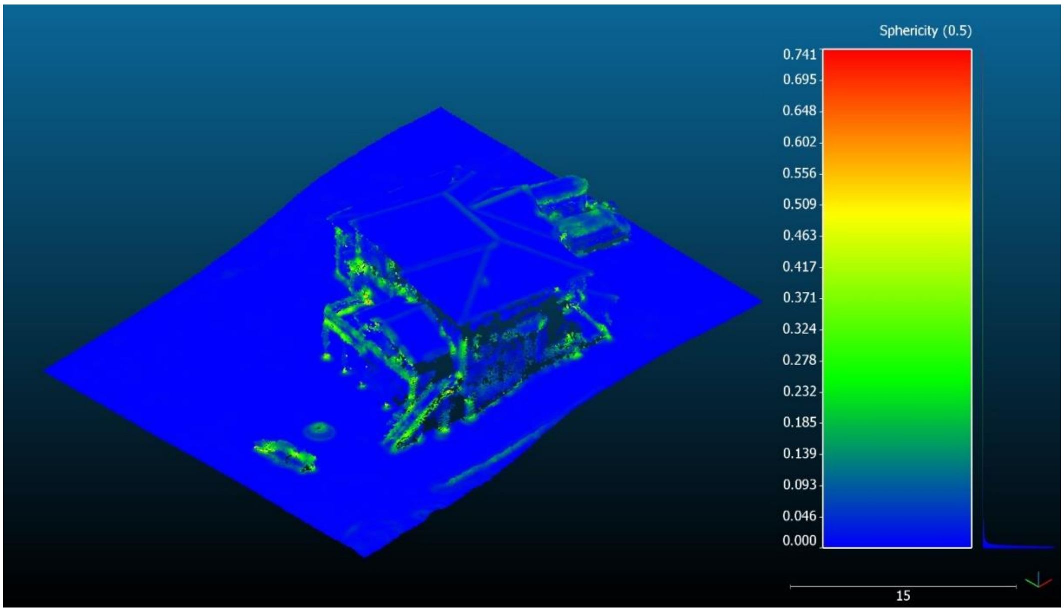


Figure 10: Sphericity of a building point cloud in 0.5 m radius

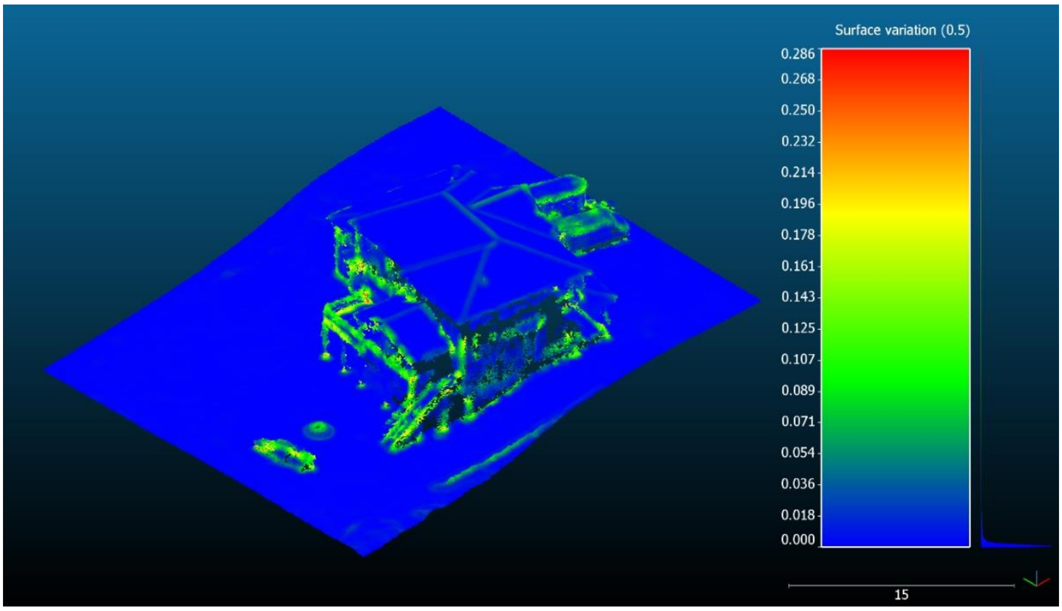


Figure 11: Surface variation of a building point cloud in 0.5 m radius

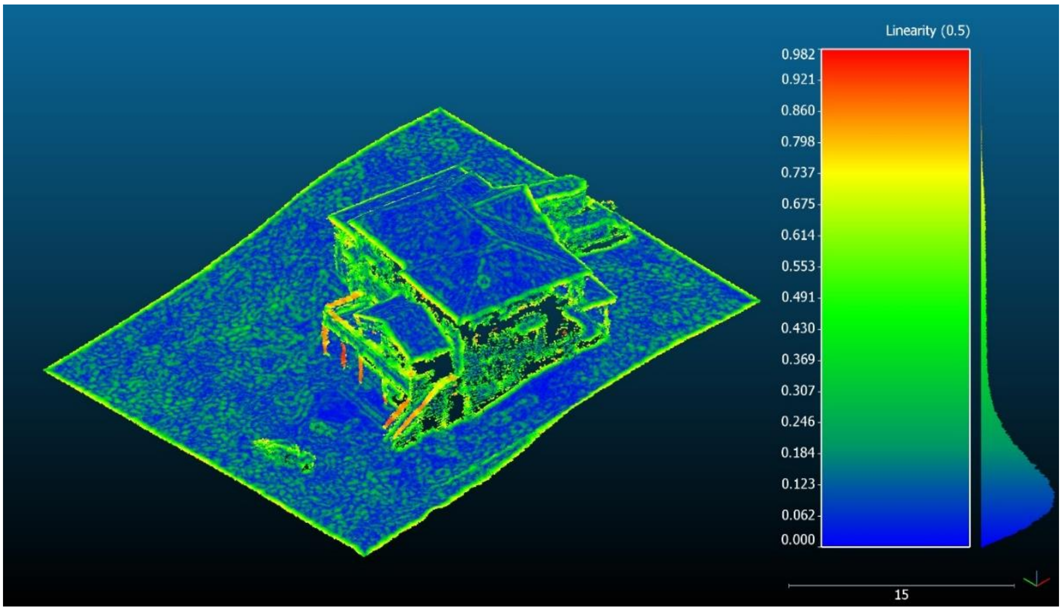


Figure 12: Linearity of a building point cloud in 0.5 m radius

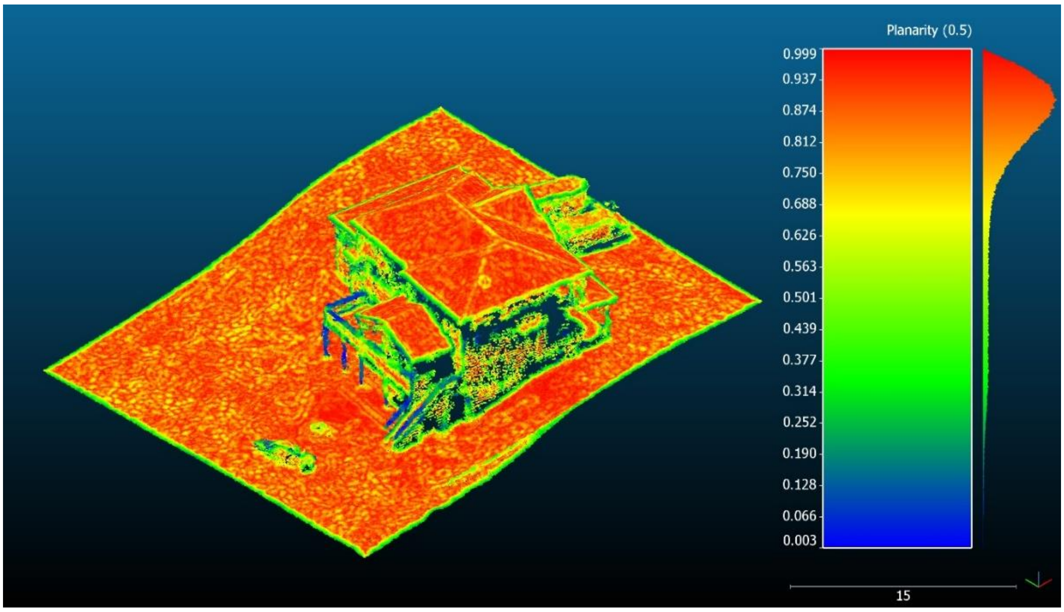


Figure 13: Planarity of a building point cloud in 0.5 m radius

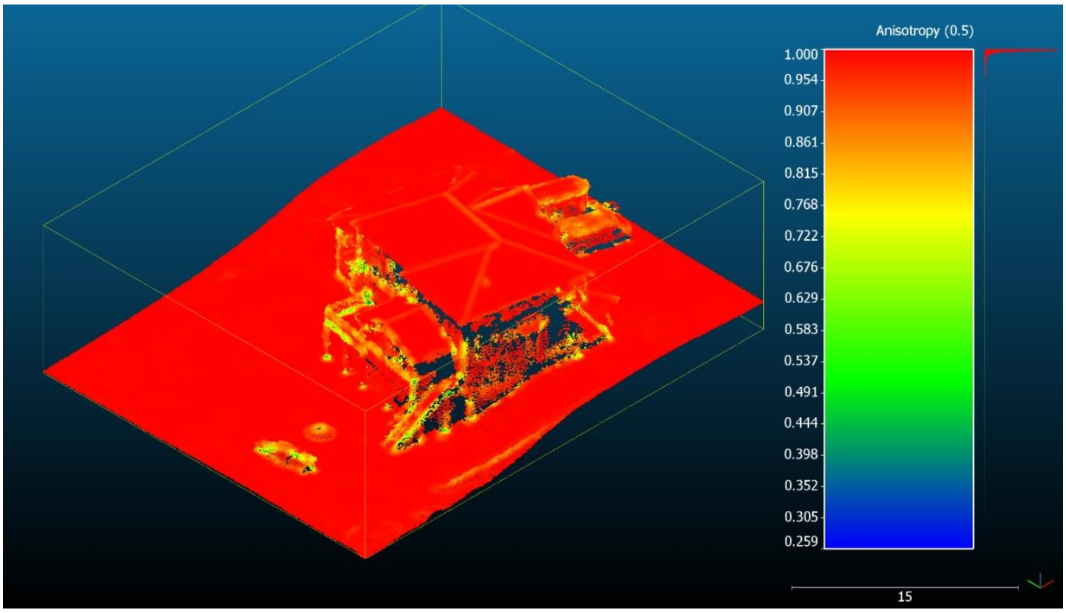


Figure 14: Anisotropy of a building point cloud in 0.5 m radius

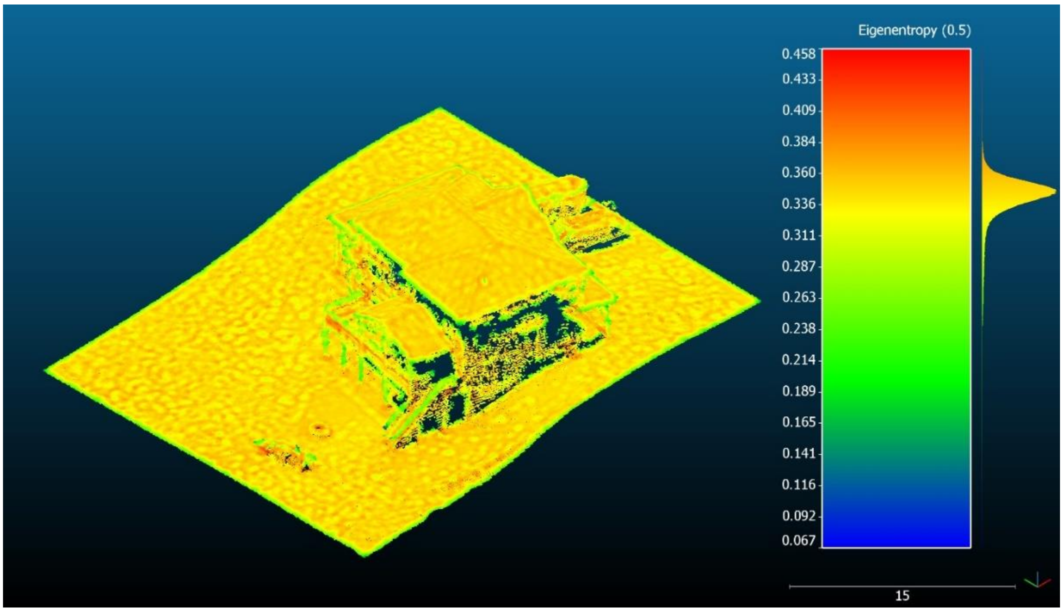


Figure 15: Eigenentropy of a building point cloud in 0.5 m radius

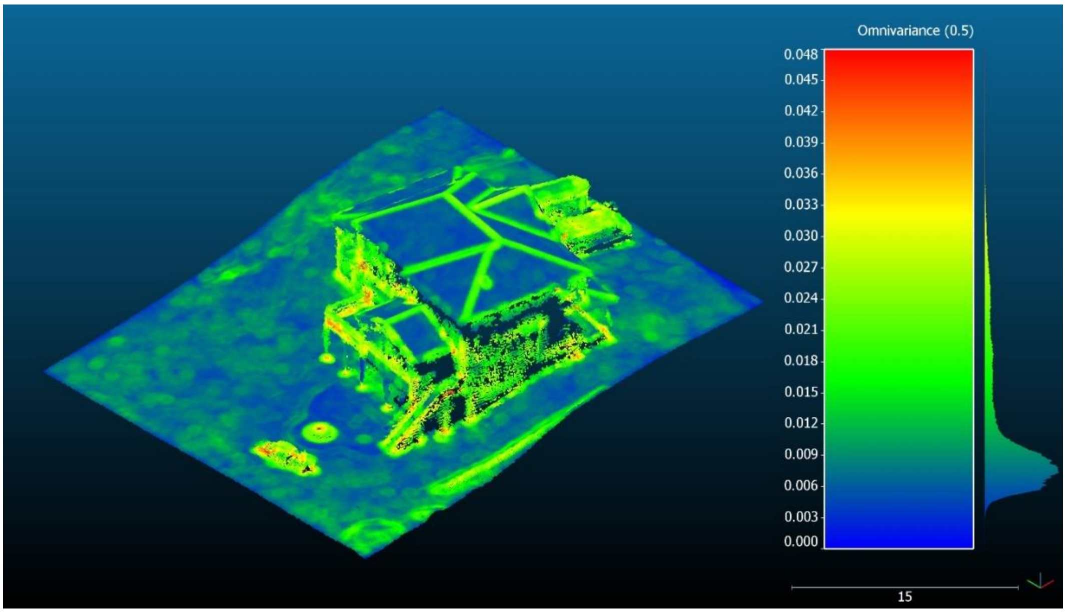


Figure 16: Omnivariance of a building point cloud in 0.5 m radius

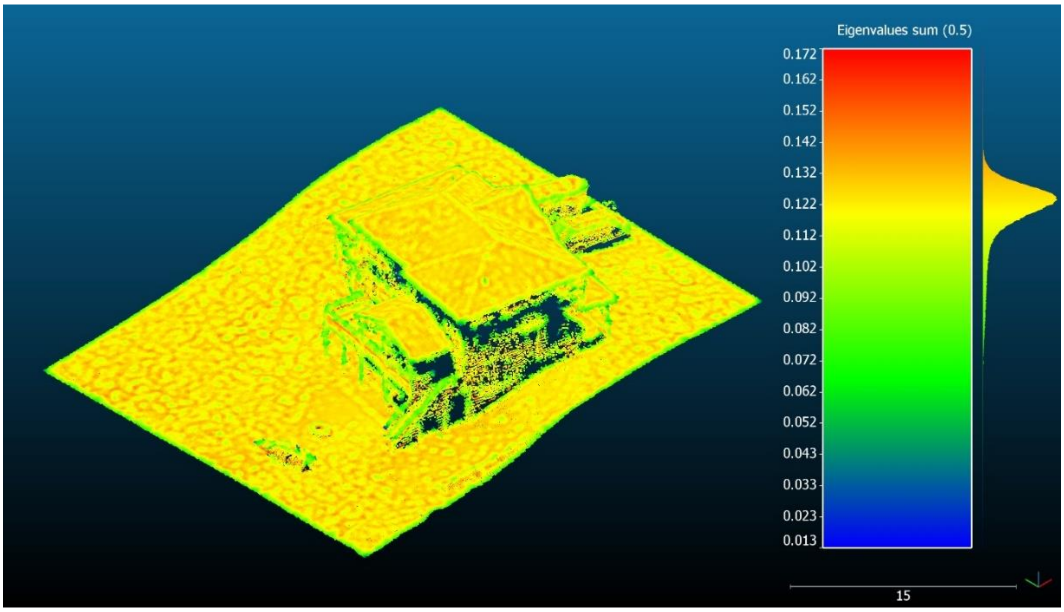


Figure 17: Sum of eigenvalues of a building point cloud in 0.5 m radius.

2.6 Hough Transform

This thesis aims to extract DTM by using a planarity-based approach and to measure planarity, Hough transform is utilized. The principles of the Hough transform are explained in Section 2.6, and its utilization in plane detection is discussed in Section 2.7.

The Hough transform is a method for detecting lines in visual representations (US Patent No. 3,069,654, 1962). To identify straight lines, curves, and ellipses, an improved version, "Generalized Hough Transform" (GHT) (Duda & Hart, 1972), is frequently applied in the digital image processing area (Tarsha-Kurdi, et al., 2007). Detecting straight lines is the most basic application of the GHT. It is based on a voting mechanism after conversion from image space to a parameter space. In Figure 18, the conversion from (x,y) to (ρ,θ) parameter space is illustrated for lines. (x,y) space can be converted into (ρ,θ) parameter space where ρ is the orthogonal distance to the straight line from (x,y) to the origin, and θ is the angle between the x-axis and the line segment AB, and the slope of AB can be found by the following equation (Chandrasekar & Durga, 2014):

$$m = -\frac{\sin\theta}{\cos\theta} = -\cot\theta \quad (1)$$

The two parameters that form a straight line are estimated using the linear Hough transform method. However, there are infinitely many lines that can pass through a single point. Hence, for a single point, different ρ values can be calculated by changing θ values (Chandrasekar & Durga, 2014). Some points in *(x,y)* space and their equivalents in (ρ,θ) parameter space are shown in Figure 19 and Figure 20, respectively (Chandrasekar & Durga, 2014). As seen in Figure 20, when there is a line in the image, the corresponding point in the parameter space will receive many votes from the image. As a result, the lines in the image will show themselves as peaks in the parameter space if an accumulator is implemented.

In the case of line detection, there might be a problem. When the line approaches the vertical direction, m (the slope of the line) approaches infinity. Every point in the parameter space is used as an accumulator to find or identify a line defined by $\rho = x\cos\theta + y\sin\theta$ to overcome this problem (Gonzalez & Woods, 2018).

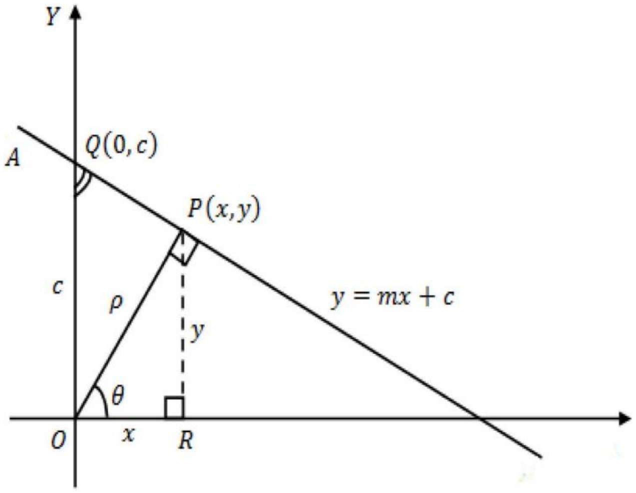


Figure 18: xy-space converted into (ρ,θ) parameter space (Chandrasekar & Durga, 2014) © 2014 IEEE

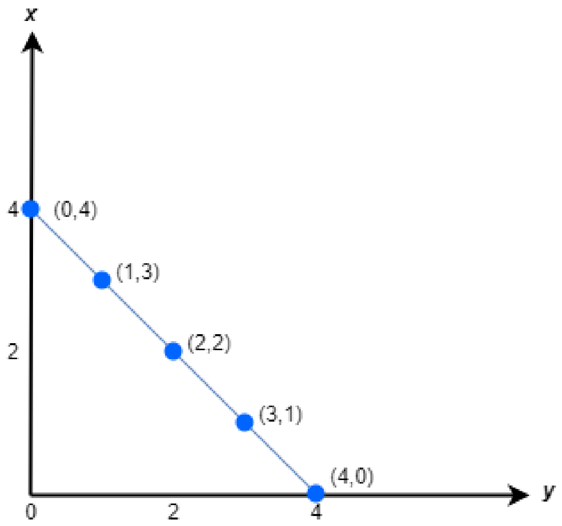


Figure 19: Line graph in (x,y) space. Adapted from (Chandrasekar & Durga, 2014) © 2014 IEEE

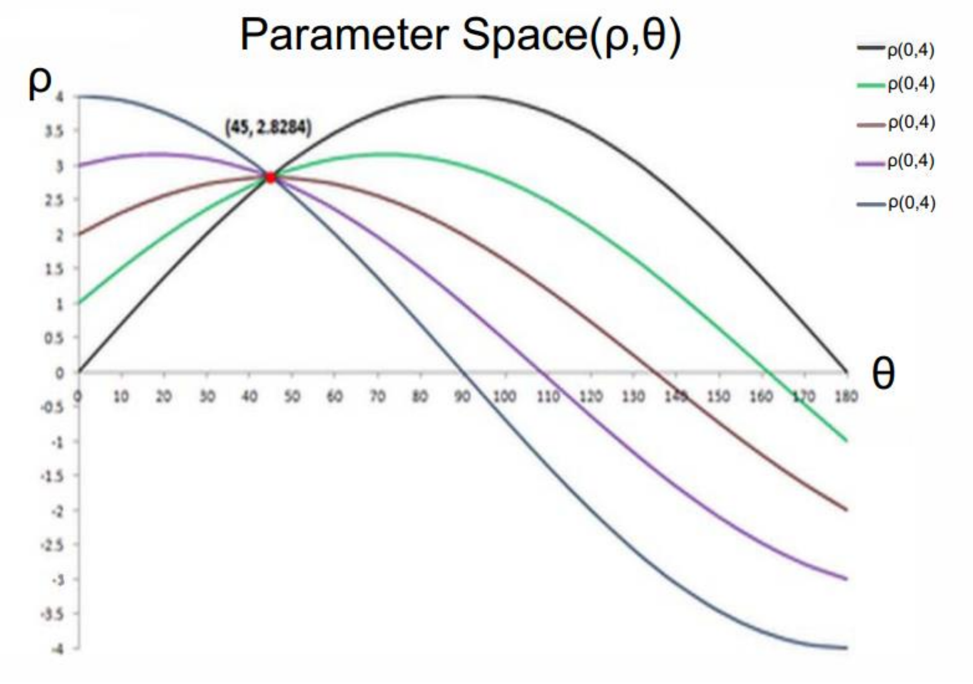


Figure 20: Line representation in (ρ,θ) parameter space (Chandrasekar & Durga, 2014) © 2014 IEEE

2.7 Approximate Coplanarity

According to (Limberger & Oliveira, 2015), existing plane detection algorithms are computationally costly and have a scale problem in big datasets; therefore, for performance enhancements, they usually benefit from non-deterministic strategies such as randomly chosen subgroup of the original point cloud. To overcome this problem, they proposed an efficient technique for plane detection in point cloud data that can work in real-time and faster than the plane detection algorithms such as Random Sample Consensus (RANSAC) and Hough Transform algorithms. Their approach is based on the segmentation of planar patches using a subdivision operation to refine an octree and approximate coplanar samples. They leveraged the detected clusters to achieve an efficient Hough-transform voting arrangement instead of casting votes for each cluster on a spherical accumulator for each sample. They utilized a Gaussian kernel centered on the cluster's best-fitting plane, which considers the variances of the cluster. In this regard, their technique enhances the kernel-based voting method described by (Fernandes & Oliveira, 2008), which uses a Gaussian distribution that has three variables over spherical coordinates (θ, φ, ρ). For better understanding, the voting procedure on spherical accumulator is illustrated in Figure 21. They first segment approximate planar clusters of samples then they use this information for their Hough-transform voting procedure where the votes are casted by cluster rather than individual samples on a spherical accumulator. For voting, they use a trivariate Gaussian distribution over spherical coordinates. In Figure 21 (a), it is shown that the number of votes cast by a plane on the spherical accumulator cell increases substantially to the equatorial regions where the color scale represents the number of votes cast by a cluster. In Figure 21 (b), it is possible that the plane is oriented incorrectly because of the noise in the point cloud, which is represented by a cone of normals around a normal of the best-fitting plane's orientation shown in red. In Figure 21 (c), for a fixed value of ρ, on the equator, these uncertainity cases result an isotropic Gaussian kernel. Nevertheless, an isotropic Gaussian kernel is observed near a pole.

Since the eigenvalues of the covariance matrix describe the proportions of the variances of the sample distribution in the given cell, similar to covariance-based feature studies discussed in Section 2.5, they are used to filter out the non-planar clusters. Two conditions have been inspected to determine whether a series of samples are approximately coplanar:

1. The cluster thicknesses
2. The degree of isotropy to eliminate misclassification of lines and thin extended clusters as planes

To achieve that, they defined two scaling factors s_α and s_β. s_α defines the relative off-plane displacement, and s_β defines the degree of sample anisotropy of the cluster. Note that $0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3$ and a point is marked as approximate coplanar if $(\lambda_2 > s_\alpha \lambda_1)$ and $(s_\beta \lambda_2 > \lambda_3)$ is satisfied. In the study $s_\alpha = 25$ and $s_\beta = 6$ scale factors produced satisfactory results. Thus, the same values are used in this thesis study as well.

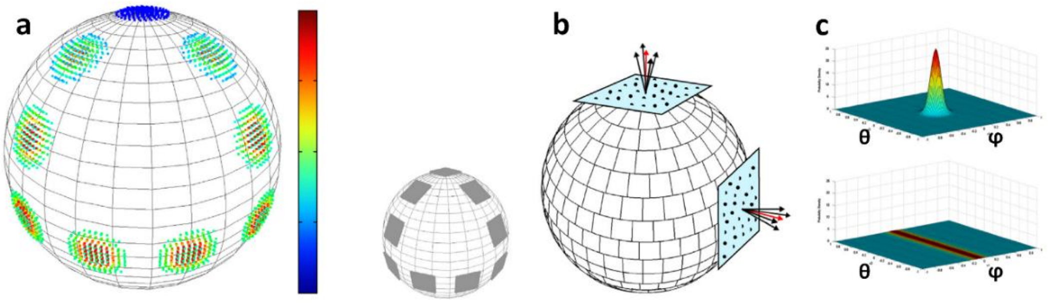


Figure 21: (a). Number of the votes casted and best-fitting planes (b) Cones of normals (c) isotropic (top) and sliced anisotropic (bottom) are the Gaussian kernels in the (θ, ϕ). (Limberger & Oliveira, 2015)

The robustness of the proposed algorithm was tested on a synthetically created cubic box dataset where inclined planes and missing data cases occur. It performed well both in these cases, and the results are shown in Figure 22.

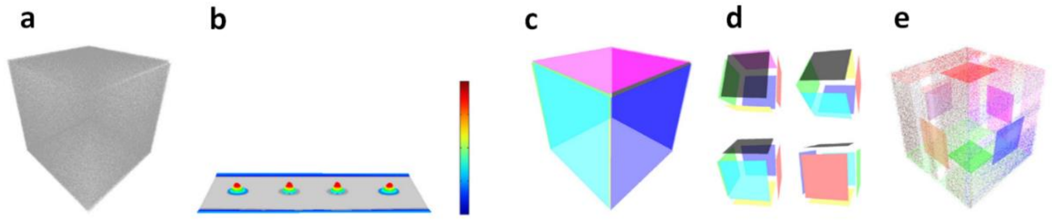


Figure 22: Cube dataset. (a) A point cloud showing a cube that evenly distributed noise on each face. (b) A 3D view of a flattened accumulator slice (c) Reconstructed planes (d) Inclined reconstructed planes with 20°, 40°, 60°, and 80° around the x-axis, respectively. (e) Thinned version of the cube that contains around 60% of the initial area for each face. (Limberger & Oliveira, 2015)

2.8 Region Growing Segmentation

Region growing segmentation is used in this thesis to divide the point cloud into sub-groups to estimate a rough ground surface from these segments. This section explains the principles of the algorithm.

Segmentation is a process of dividing the point cloud measurements into such groups that the points belonging to the same surface or region have the same label (Rabbani, Heuvel, & Vosselman, 2006). (Vo, Truong-Hong, Laefer, & Bertolotto, 2015), evaluate point cloud segmentation algorithms in three main groups: model-fitting, region growing, and clustering feature-based methods. Region growing segmentation is proposed by (Besl & Jain, 1988) because it was hard to interpret the extensive amounts of raw data by computers. Although the algorithm was initially built for image segmentation, it is developed by other researchers for 3D point cloud applications. (Vieira & Shimada, 2005), extended the existing algorithm to handle noisy point clouds by focusing on seed point selection. (Nurunnabi, Belton, & West, 2012) proposed an algorithm using PCA-based region growing segmentation, (Vo, Truong-Hong, Laefer, & Bertolotto, 2015) introduced an octree-based region growing segmentation.

Briefly, in region growing segmentation, the aim is to merge the sufficiently close points with the smoothness condition. First, a seed point is selected, and the neighbor points are found. Then, these neighboring points are tested for the curvature value to check whether it satisfies the smoothness criteria. If so, the points are added to the same cluster. The algorithm steps are given as the following steps:

1. All the points are sorted by their curvature value, and the region starts to grow from the minimum curvature value (flattest area) to reduce the total number of segments.
2. The minimum curvature point is added to a seed point set.
3. For each seed point, the neighboring points are found.
4. For each neighboring point, a normal is calculated.
5. If the angle between the point normal and the seed point set normal is less than the given threshold, the point is added to the current region.
6. Current seed is removed from seed point set.
7. If there is no point left in the seed set, the process is repeated from the beginning. (Rusu & Cousins, 2011)

2.9 Common Surface Interpolation Techniques

2.9.1 Introduction

Spatial interpolation is a technique to logically estimate a continuous field's values that are not actually measured by using existing observations (Longley, Goodchild, Maguire, & Rhind, 2005). Most of the quantities such as elevation, climate data, soil properties, population density, precipitation, etc. are described by measured or sampled point data, which are often irregularly distributed in Geographic Information System (GIS) applications, so there is a need for interpolation techniques to derive information about the missing points (Mitas & Mitasova, 2005).

For example, in practice, it is impossible to cover a whole country with weather stations. Climate values around the areas without these stations can be interpolated from existing weather stations. Despite the availability of different spatial interpolation techniques, some of them are used in DEM interpolation. Within the scope of this thesis, only the common interpolation methods for DEM interpolation are covered. Different spatial interpolation methods are covered in this thesis with their advantages and disadvantages to explain why Inverse Distance Weighting is used over the other spatial interpolation techniques.

2.9.2 Inverse-Distance Weighting (IDW)

IDW is one of the most straightforward local neighborhood interpolation methods in which it is assumed that the value of an unsampled point can be estimated as a weighted average of the values at the points within a certain cut-off distance or at a specified m of the nearest points (Longley, Goodchild, Maguire, & Rhind, 2005). The value of the unsampled point can be estimated by using the following formula (Garnero & Godone, 2013):

$$Z_j = \frac{\sum_{i=1}^{n} \frac{z_i}{\left(h_{ij}+\delta\right)^{\beta}}}{\sum_{i=1}^{n} \frac{1}{\left(h_{ij}+\delta\right)^{\beta}}} \tag{2}$$

where Z_j is the value at an unsampled location, Z_j is the known value, β is the weight, δ is the smoothing parameter, and h_{ij} is the Euclidian separation distance between the known and the unknown point. The main assumption in this technique is that all points on the surface of the world are treated interdependent based on distance. According to (Achilleos, 2011), the height of an interpolated point is connected with the elevations of the reference points around, and it is generally inversely proportional to the distance from each point of reference raised to a square or cubic power. The study also discusses the method's weaknesses in DEM generation, such as the occurrence of peaks in various places, smoothened out curved parts in low-resolution DEM, and the creation of small isolated islands in contour lines. As an example, interpolation results are shown on the Ankeny dataset. For the sake of computation, the Ankeny dataset (Becker, Rosinskaya, Häni, D'Angelo, & Strecha, 2018) is resampled into approximately 75000 points. Then from these points, a DSM is generated by using IDW interpolation. The resulting DSM and the elevation histogram are illustrated in Figure 23 and Figure 24, respectively.

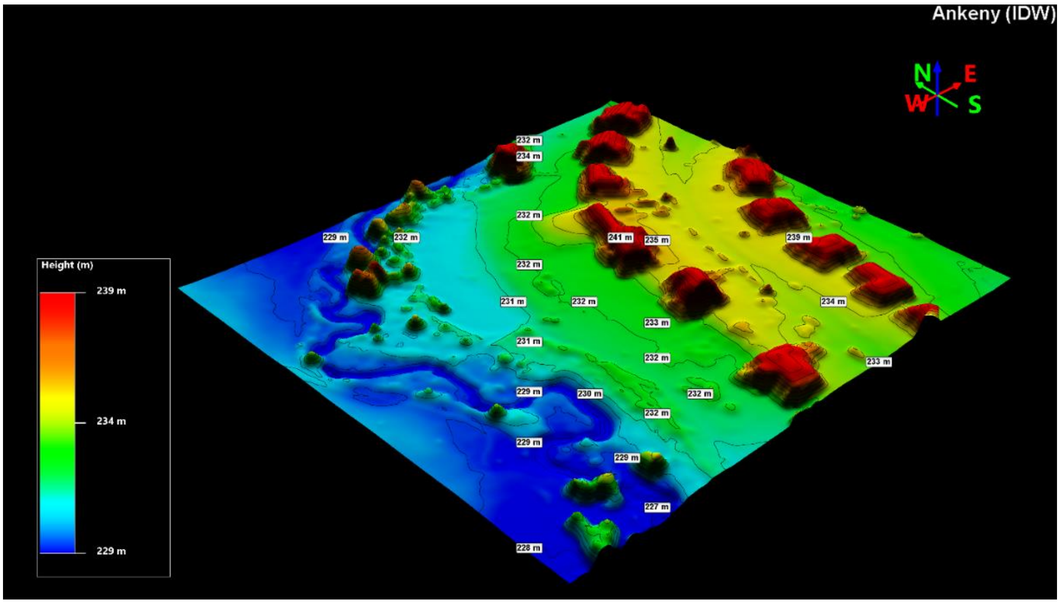


Figure 23: IDW interpolation of resampled Ankeny data

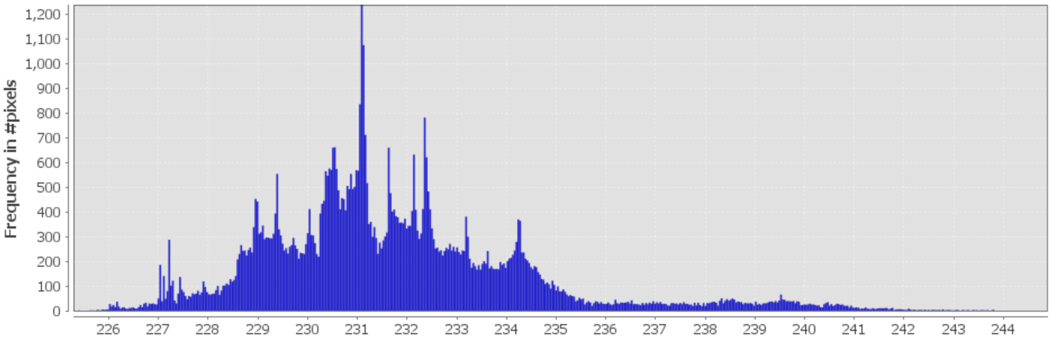


Figure 24: Elevation histogram of IDW interpolated Ankeny data

2.9.3 Spline Interpolation

Spline interpolation is a method of interpolation to construct new points using a specific type of piecewise polynomial called a spline. An interpolation of low degree polynomials is chosen to fit into sub-sets of values rather than a single high-degree polynomial fitting all values directly. A linear combination of n functions is used in the method, one for every known point (Garnero & Godone, 2013), and the generic form of the interpolation is defined as:

$$\hat{Z}(s_0) = \sum_{i=1}^{n} \omega_i \phi(\|s_i - s_0\|) + \omega_{n+1} \qquad (3)$$

where $\phi(r)$ is the interpolation function, $\|s_i - s_0\|$ is the Euclidian distance between an unknown and observed point, and ω_i is the weight.

(Garnero & Godone, 2013) listed the common spline functions, such as thin-plate spline, multi-quadratic, inverse multi-quadratic, completely regularized spline function, and spline with tension function.

Table 4: Common spline functions (Garnero & Godone, 2013)

Example Spline Functions	
Thin-plate Spline function	$\phi(r) = (\sigma \cdot r)^2 \ln(\sigma \cdot r)$
Multi-quadric function	$\phi(r) = [r^2 + \sigma^2]^{1/2}$
Inverse Multi-quadric function	$\phi(r) = [r^2 + \sigma^2]^{-1/2}$
Completely regularized Spline function	$\phi(r) = -\sum_{n=1}^{\infty} \frac{(-1)^n \cdot r^{2n}}{n!\, n} = \ln\left[\frac{\sigma \cdot r}{2}\right]^2 + E_1\left[\frac{\sigma \cdot r}{2}\right]^2 + C_E$
Spline with tension function	$\phi(r) = \ln\left(\frac{\sigma \cdot r}{2}\right) + K_0(\sigma \cdot r)^2 + C_E$

For the given functions in Table 4, r is the distance between the point and the sample, σ is the tension parameter, E_1 is the exponential integral function, C_E is the Euler – Mascheroni constant (0,577215), and K_0 is the modified Bessel function.

From the thinned Ankeny data, a DSM is generated by using spline interpolation. The resulting DSM and the elevation histogram are illustrated in Figure 25 and Figure 26, respectively. There are some spikes and surface anomalies around the vegetated area and building boundaries in this DSM.

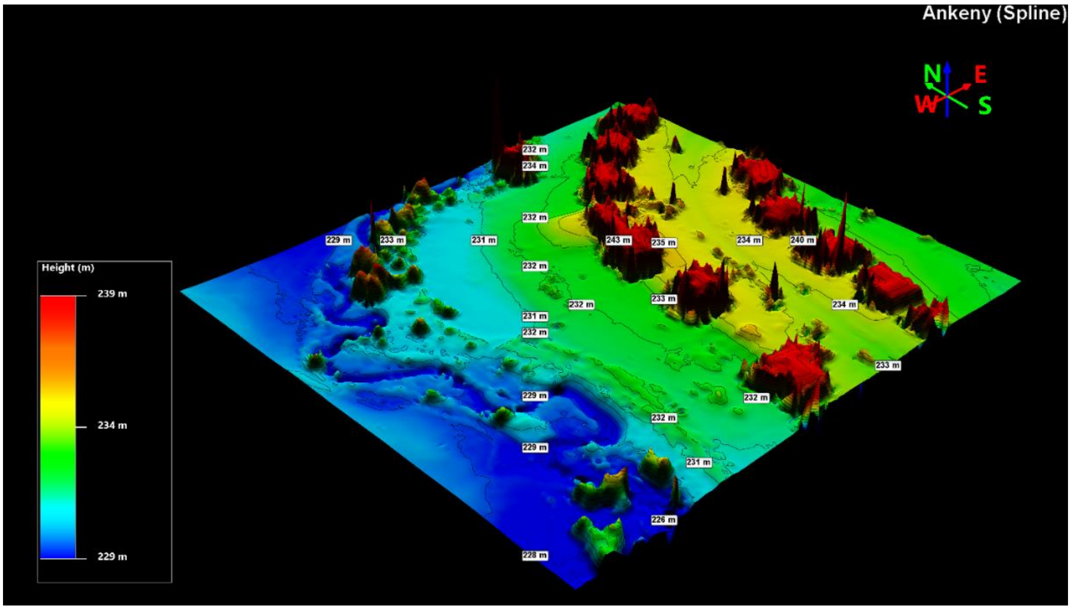


Figure 25: Spline interpolation of resampled Ankeny data

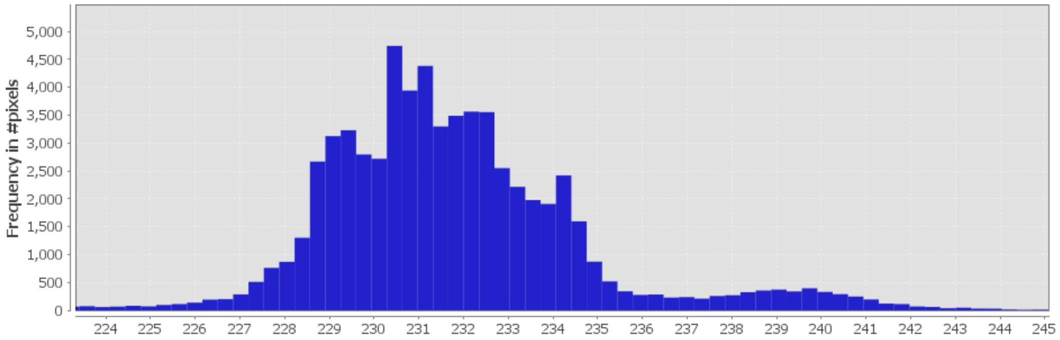


Figure 26: Elevation histogram of spline interpolated Ankeny data

2.9.4 Triangulated Irregular Networks (TIN)

Triangulation is a relatively straightforward technique to determine a complete collection of minimal clusters in a scattered data set where these minimal clusters are triangles for *(x, y, f(x, y))* data forms, and data points are located at each vertex of the triangle (Watson & Philip, 1984). Elevation data can be evaluated in the form of *(x,y,f(x,y))* where *f(x,y)* is the elevation. An ensemble of space-filling triangles forms a triangular, irregular mesh, and the interpolation system is used directly to raw data without griding (Watson & Philip, 1984). A triangulation that is based on a well-defined and cost-efficient criterion is required in systems using triangular tesselations, which is why Delaunay is utilized in those systems (Fowler & Little, 1979). According to (Fortune, 1995), let S be a set of n points in a d-dimensional space (E^d). The diagram of Voronoi divides E^d amongst regions with one region per site so that the points of site $s \in S$ in the region are closer to s than any other location in S. The triangulation of Delaunay S is the unique triangulation of S so that no S elements are contained in the circumference of any triangle. In this case, triangulation is evolved from planar to the arbitrary dimension. The convex hull of the S is decomposed into simple structures utilizing S vertices.

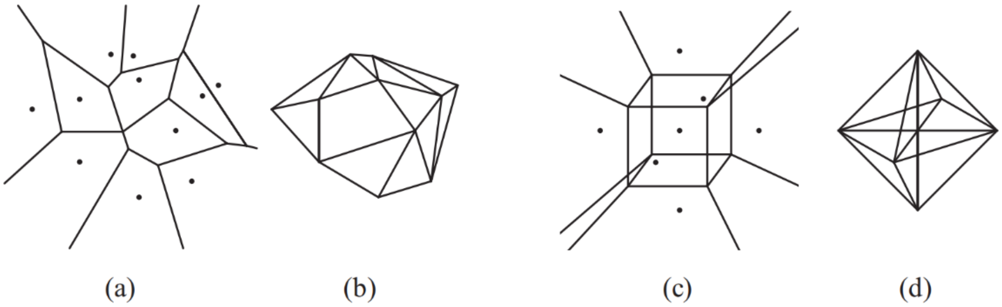


Figure 27: (a,c) represent the Voronoi diagram both in 2D and 3D, respectively. Similarly, (b,d) showing the Delaunay triangulation for the same set of points in 2D and 3D, respectively (Fortune, 2017)

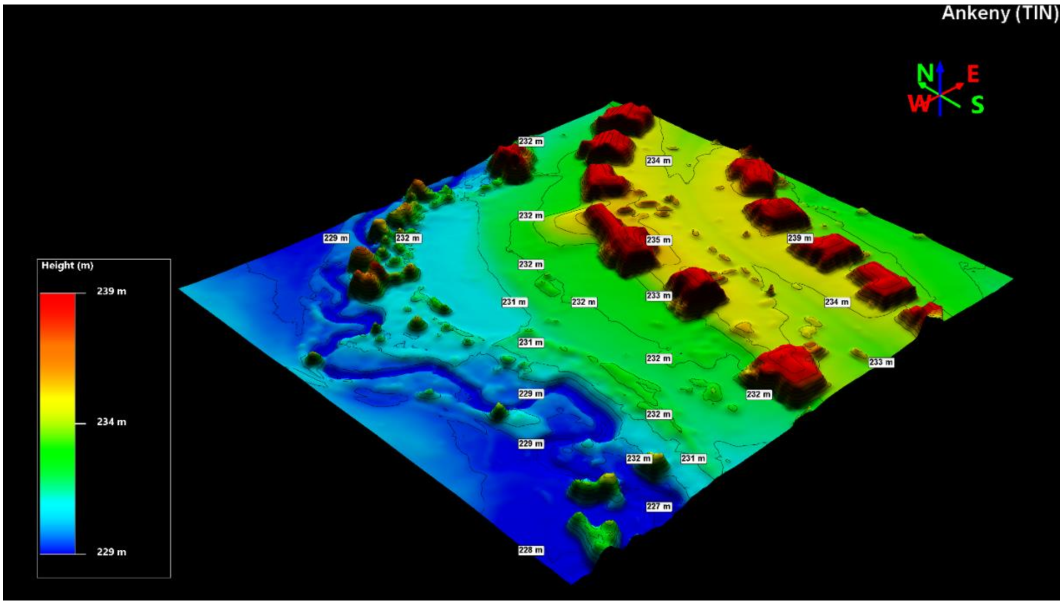


Figure 28: Rasterized TIN interpolation of resampled Ankeny data

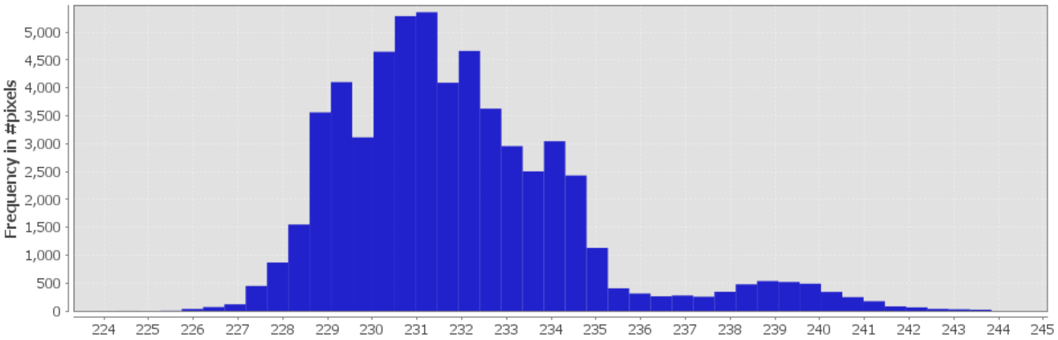


Figure 29: Elevation histogram of TIN interpolated Ankeny data

2.9.5 Kriging

Kriging is a geo-statistical local estimation method of the best linear unbiased estimator that utilizes variograms for spatial and temporal variables (Arun, 2013; Chung, Venkatramanan, Elzain, Selvam, & Prasanna, 2019). The variogram is a measure of spatial dependence (Chung, Venkatramanan, Elzain, Selvam, & Prasanna, 2019) that gives a piece of precise information about the influence zone (Matheron, 1963). In other words, the more or less fast growth of the variogram means a deterioration of the impact of a sample on more and more remote areas of the value measured (Matheron, 1963). Kriging is defined as (Chung, Venkatramanan, Elzain, Selvam, & Prasanna, 2019):

$$Z_K^* = \sum_{j=1}^{n} \lambda_i Z_i \tag{4}$$

where Z_K^* is the Kriging estimation, λ_i is the weight, and Z_i is the variable.

The weight is selected to provide an unbiased estimator and a minimum estimated variance (Journel & Huijbregts, 1976), and in unbiased kriging, the condition is satisfied when $E\{Z_V - Z_K^*\} = 0$, where Z_V is the actual value and Z_K^* is the estimated value. Also, note that the summation of the weights is equal to unity. The estimate of variance of kriging is defined as:

$$\sigma_K^2 = E\{[Z_V - Z_K^*]^2\} = \bar{C}(V,V) + \mu - \sum_{i=1}^{n} \lambda\bar{C}(v_i,V) \tag{5}$$

where $\bar{C}(V,V)$ is the covariance between sample variables, μ is the Lagrange parameter, and $\bar{C}(v_i,V)$ is the covariance between the sample variable and the estimates.

The semi-variogram is defined as (Chung, Venkatramanan, Elzain, Selvam, & Prasanna, 2019):

$$\gamma(h) = \frac{1}{2N(h)}\sum_{i=1}^{N(h)} [Z(x_i) - Z(x_i + h)]^2 \tag{6}$$

where $Z(x_i)$ and $Z(x_i + h)$ are the sampled values at x_i and $x_i + h$, and $N(h)$ is the number of pairs of samples separated by the lag h.

The following equation is used to calculate the covariance between sample data (Chung, Venkatramanan, Elzain, Selvam, & Prasanna, 2019):

$$C(h) = sill - \gamma(h) \tag{7}$$

where $C(h)$ is the covariance, and $\gamma(h)$ is the semivariogram.

Spherical, exponential, Gaussian, linear, and power models are the most widely used theoretical semivariograms that fit to the empirical semivariogram (Pouliou, Kanaroglou, Elliott, & Pengelly, 2008). Simple, Ordinary, and Universal Kriging types are the most frequently used kriging types, and the major distinction is the assumptions made concerning the spatial trend of the mean value of the examined variable (Pouliou, Kanaroglou, Elliott, & Pengelly, 2008). An ordinary kriging with a spherical semivariogram is used to create Figure 31. The generic form of the spherical variogram is defined as (Bailey & Gatrell, 1995):

$$\gamma(\mathrm{h}) = \begin{cases} \alpha + \sigma^2 \left(\dfrac{3\mathrm{h}}{2\mathrm{r}} - \dfrac{\mathrm{h}^3}{2\mathrm{r}^3} \right) & \text{for } 0 < \mathrm{h} \leq \mathrm{r} \\ 0 & \text{for } \mathrm{h} = 0 \\ \sigma^2 & \text{otherwise} \end{cases}$$

where σ^2 is the sill, r is the range, and α is the nugget effect.

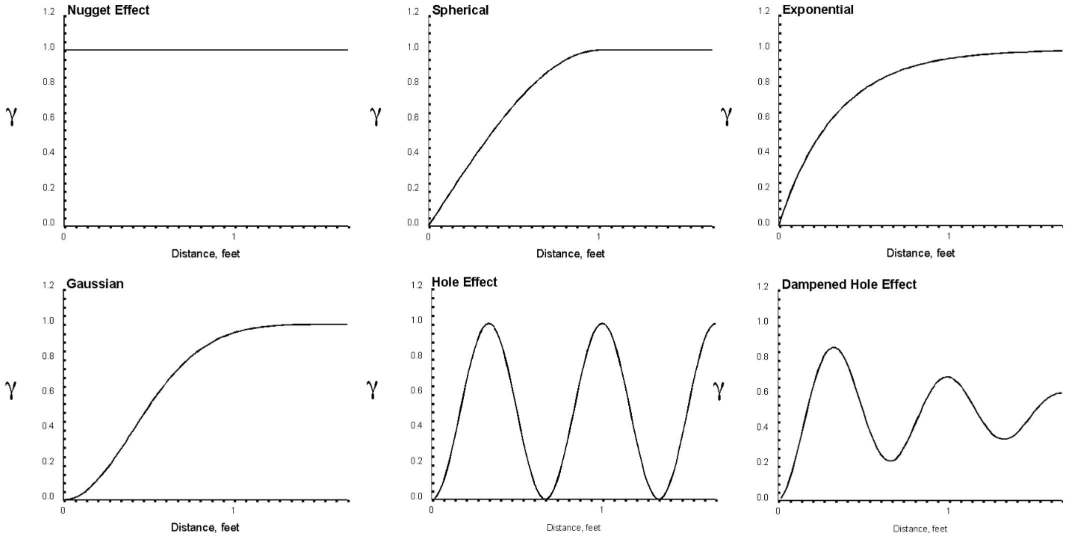


Figure 30: Common variogram types (Deutsch, 2003)

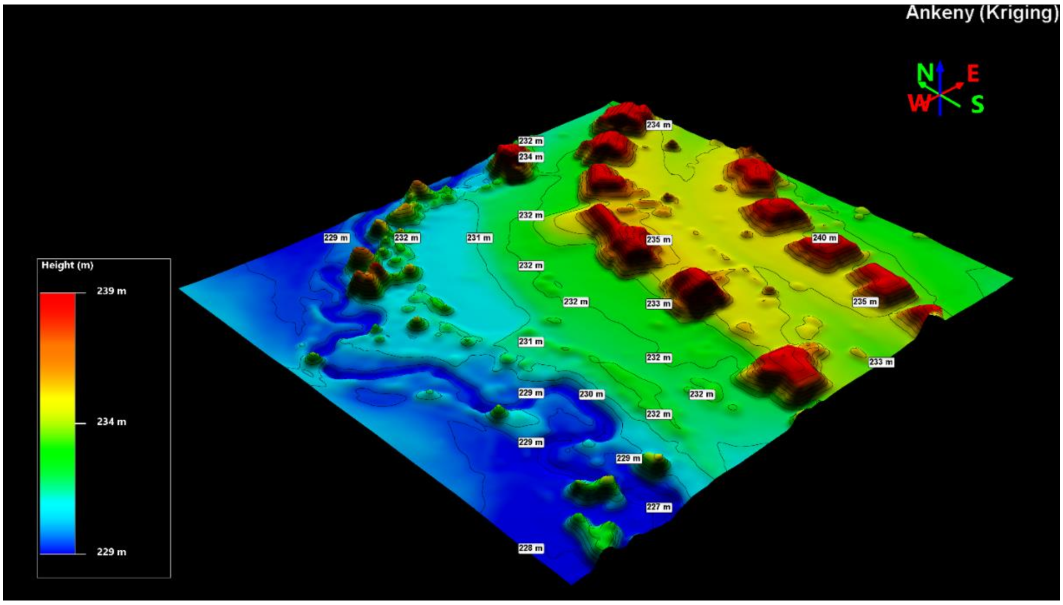


Figure 31: Kriging interpolation of resampled Ankeny data

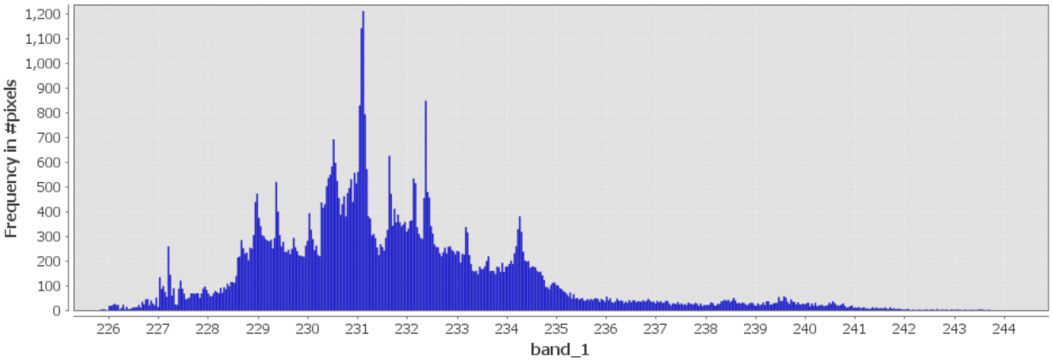


Figure 32: Elevation histogram of kriging interpolated Ankeny data

2.10 Point Cloud Classification and Digital Terrain Model Extraction Studies

2.10.1 Introduction

Digital terrain modeling is the common name for methods for building a DTM from different data inputs such as point clouds and aerial imagery. Some points are sampled from the ground to form accurate, dense, and distributive observations; then, these sample points represent the ground surface (Li, Zhu, & Gold, 2004). If information about other than sample points is needed, in that case, interpolation techniques are used with the sampled data points to construct a DTM surface (Li, Zhu, & Gold, 2004) to obtain information about the rest of the points. DTM has been a critical research subject for the International Society for Photogrammetry and Remote Sensing (ISPRS) since the 1960s, as photogrammetrists usually are manufacturers of DTM (Li, Zhu, & Gold, 2004). Since then, many different approaches have been proposed to create the DTM. Since there are multiple data acquisition methods such as satellite imaging, ALS, and photogrammetry, these studies have focused on various remote sensing data sources. Several studies focused on aerial point cloud classification and, consequently, classification of ground points. In the meantime, some focused on DTM extraction specifically. These studies are divided into two chapters and discussed separately.

2.10.2 Point Cloud Classification Studies

Point cloud classification is a crucial step in understanding the topography, the environment for further applications like object recognition, surface reconstruction, etc. According to the American Society for Photogrammetry and Remote Sensing (ASPRS), standard LIDAR point classes are shown in Table 5.

Table 5: ASPRS Standard Lidar Point Classes (Point Data Record Formats) (The American Society for Photogrammetry and Remote Sensing, 2019)

Classification Value	**Meaning**
0	Created, never classified
1	Unclassified
2	Ground
3	Low Vegetation
4	Medium Vegetation
5	High Vegetation
6	Building
7	Low Point (noise)
8	Reserved
9	Water
10	Rail
11	Road Surface
12	Reserved
13	Wire - Guard (Shield)
14	Wire - Conductor (Phase)
15	Transmission Tower
16	Wire-structure Connector (e.g., Insulator)
17	Bridge Deck
18	High Noise
19	Overhead Structure
20	Ignored Ground
21	Snow
22	Temporal Exclusion
23-63	Reserved
64-255	User definable

(Oude Elberink & Maas, 2000) used anisotropic height texture measure to detect trees and buildings from laser scanner image textures using the knowledge that the buildings have a regular, smooth pattern with height, in contrast to trees that show the opposite characteristics. This distinctness is used in this study to derive co-occurrence matrices for discriminating between trees and buildings.

The 3D covariance matrix, also known as a 3D structure tensor, describes the local 3D structure (Jutzi & Gross, 2009). Using this matrix, (Demantké, Vallet, & Paparoditis, 2012) derived verticality to detect vertical rectangles. Similarly, (Weinmann, Urban, Hinz, Jutzi, & Mallet, 2015) derived linearity, planarity, scattering omnivariance, anisotropy, eigenentropy, local surface variation and used these features to classify the points in large-scale scene analysis in urban areas. (Hackel, Wegner, & Schindler, 2016) extended the set of covariance features; additionally, they used 1st order 1st axis, 1st order 2nd axis, 2nd order 1st axis, and 2nd order 2nd axis moments for feature extraction. Then they used a random forest classifier to calculate the probabilities of the points for each class. (Becker, Rosinskaya, Häni, D'Angelo, & Strecha, 2018) have combined pointwise color information and multi-scale pyramid approach with the features utilized by (Hackel, Wegner, & Schindler, 2016). They indicated that if color information is provided, it can be used to boost performance and classification accuracy. (Xu & Yang, 2018) used eigenentropy based scale selection for the covariance and color features. They pointed out that scale has a vital role in point classification.

2.10.3 DTM Extraction Studies

(Kraus & Pfeifer, 1998) proposed the first interpolation-based filtering method that approximates the terrain iteratively. While doing that, they used weighted linear least squares interpolation. In binary pictures, two critical operations are usually used to expand (dilate) or decrease (erode) feature size (Zhang, et al., 2003). (Nar, Yilmaz, & Camps-Valls, 2018), proposed an algorithm based on sparsity-driven stepwise nonlinear smoothing operation by minimizing the cost function.

(Elmqvist, Jungert, Lantz, Persson, & Söderman, 2001) proposed an algorithm based on active contours to extract DTM on grid data. Metaphorically, they draped a rubber cloth on the terrain and tried to extract it by using the cloth's rigidity and elasticity. (Zhang, et al., 2016) has a similar approach based on a physical process called Cloth Simulation Filtering. First, they process the point cloud upside down as a cloth draped on these inverted points. Then, they analyze the interaction between the cloth and the points to give the shape of the final cloth.

(Vosselman, 2000) used a slope-based approach which is similar to erosion operation in mathematical morphology. In this research, the height difference between two points plays an essential role. (Zhang, et al., 2003) proposed a progressive morphological filter that combines erosion and dilation operation to get rid of non-ground objects. For the robustness of the approach, they used different window sizes. Also, note that their filter depends on the elevation difference.

(Axelsson, 2000) used adaptive TIN models that start with a coarse TIN for the given point cloud, then iteratively densifying this TIN model with the adjusted parameters in each iteration. It goes on until all the points are classified as ground or non-ground. (Sohn & Dowman, 2002), used two-step progressive TIN approach to obtain DTM. They used downward and upward TIN densification and, Minimum Description Length to find the flattest tetrahedrals.

(Brovelli, Cannata, & Longoni, 2002) have an edge-detection-based approach to create splines. These splines are later used to detect above-ground object. In their assumption, the points above the splines are considered as potential objects, and the points below the splines are the potential ground points. To connect the edges, they used a region growing step that works on raster data.

(Wack & Wimmer, 2002) first, convert the point cloud into a low-resolution elevation model to overcome the large objects and dense vegetation. Then, using a Laplacian of Gaussian operation, all the non-ground objects are detected and removed. They do the same for the finer resolution raster. The values of the non-ground object elements in finer grids are replaced with the coarser resolution grid. They do it iteratively to obtain the final results.

(Mongus & Žalik, 2012) conducted a parameter-free ground filtering of LIDAR study based on multi-resolution thin-plate spline interpolation to detect non-ground objects. In each resolution level, they use top-hat transformation to enhance the low-contrast high-frequency details. In the end, they use a statistical filter to distinguish non-ground objects.

(Bartels, Wei, & Mason, 2006) came up with a statistical approach that finds the points that disturb the normal distribution. Their assumption is based on naturally measured samples that would lead to a normal distribution, and the ones that disrupt the normal distribution are non-ground points. Although it performs well in building and vegetation classification, it might fail in complex scenes that contain bridges and motorways.

Advances in Deep Learning and Graphic Processor Units (GPU) led remote sensing researchers to use them for DTM extraction. (Hu & Yuan, 2016) presents a novel way of extracting ground points using Convolutional Neural Networks based (CNN). The adjacent points within a window are retrieved and converted into a picture for each point with a spatial context. The classification of a point may thus be considered as image classification; the transition from point to an image is carefully carried out in the surrounding region, taking account of high-level information. Similarly, (Xu & Yang, 2018) used CNN with intensity, eigenvalue-based covariance features, normal vector-based features, and height above ground values. They benefit from eigenentropy for their scale selection to make the classification more efficient and straightforward.

(Çınar & Koçan, 2019) trained a generative adversarial network that creates DTM. In their approach, instead of modifying the existing DSM, they create DTM from scratch. Although some artifacts exist, the study shows that creating DTM for a region with a generative network is possible.

CHAPTER 3

DATA AND METHODOLOGY

3.1 Data Collection

3.1.1 The Study Area and Point Cloud Characteristics

Two different photogrammetric point clouds representing various terrain characteristics are selected to test the robustness of the algorithm. Both photogrammetric point clouds contain seven fields: X, Y, Z, R, G, B, and Classification. The first dataset called "Ankeny" has around 9 million points representing relatively flat ground, buildings, vehicles, poles, trees, and vegetation. The second dataset, called "Cadastre" has approximately 5.8 million points showing fairly steep ground, buildings, vehicles, poles, trees, powerlines, and vegetation (Becker, Rosinskaya, Häni, D'Angelo, & Strecha, 2018). The points are classified by their proposed algorithm, and the overall errors for ground points are 16.5% and 18.9% in Ankeny and Cadastre datasets, respectively. Both data have some missing points on the facades, making these point clouds challenging for DTM extraction. Ground sampling distances (GSD) for these data are $2.3\, cm/pixel$ and $5.1\ cm/pixel$, respectively. Figure 33 and Figure 34 illustrate the Ankeny data from front and back isometric views. Similarly, Figure 35 and Figure 36 show Cadastre data from front and back isometric views. Table 6 shows the general information about the datasets. Meanwhile, Table 7 shows the point cloud dataset content.

Table 6: Point cloud dataset used for evaluation (Becker, Rosinskaya, Häni, D'Angelo, & Strecha, 2018)

Dataset	**Acquisition**	**Color**	**Number of Points**	**GSD** $(cm/pixel)$
Ankeny	Aerial images	RGB	≈ 9 million	2.3
Cadastre	Aerial images	RGB	≈ 5.8 million	5.1

Table 7: Point cloud dataset content (Becker, Rosinskaya, Häni, D'Angelo, & Strecha, 2018)

Feature	**Ankeny**	**Cadastre**
Roads	+	+
Ground/Grass on flatland	+	+
Ground/Grass on slopes	-	+
Cropland	+	-
Powerlines	-	+
Vehicles	+	+

Figure 33: Front isometric view of Ankeny data

Figure 34: Back isometric view of Ankeny data

Figure 35: Front isometric view of Cadastre data

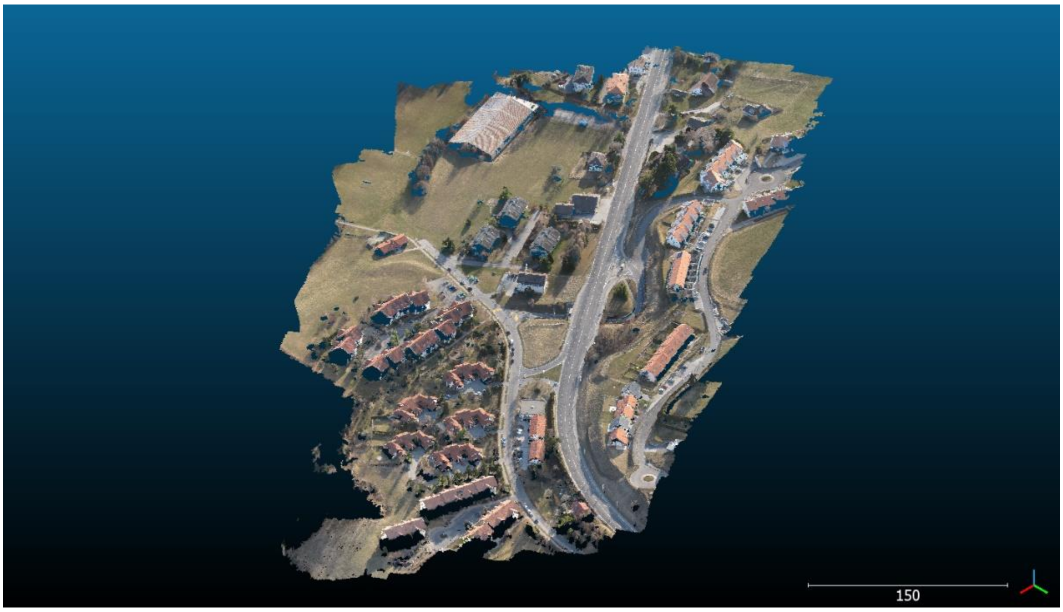


Figure 36: Back isometric view of Cadastre data

3.1.2 Ground Truth for Validation

The point cloud datasets have a "Classification" field for different classes such as ground, building, vegetation, low vegetation, etc. However, 16.5% and 18.9% of the ground points in Ankeny and Cadastre datasets are misclassified by their algorithm (Becker, Rosinskaya, Häni, D'Angelo, & Strecha, 2018). Thus, these points are corrected manually so that they can be used as the ground truth data. For the ground points, misclassified points are corrected manually within the framework of this thesis. Later on, from these ground points, terrain surfaces are created by using Delaunay triangulation to be used for the accuracy assessment of the proposed algorithm.

3.2 The Proposed Method

3.2.1 Overview

In this section, the proposed algorithm steps, whose flowchart is given in Figure 37, are discussed briefly. These steps are explained in detail in the corresponding sections.

1. Section 3.2.2 explains the voxelization process and why it is needed.
2. Section 3.2.3 describes what Approximate Coplanarity check is and how it is exploited to get rid of the non-ground object.
3. Section 3.2.4 explains why a grid-based planarity check is needed and how it is used to get rid of vertical planar parts.
4. Section 3.2.5 explains region growing segmentation and its use to segment the corresponding parts from the previous steps.
5. Section 3.2.6 explains how Delaunay triangulation is utilized to calculate a rough terrain surface to calculate the approximate height above ground values.

6. Section 3.2.7 describes the method used to eliminate the left-over roof planes from the previous steps.
7. Section 3.2.8 explains the rasterization process from the ground points.
8. Section 3.2.9 explains how the parameters and thresholds are selected.

These steps are realized using Point Data Abstraction Library (PDAL), Point Cloud Library (PCL) Strawlab Python Implementation, and some custom scripts are written from the scratch. Also note that PDAL library uses Fast Library for Approximate Nearest Neighbors (FLANN) for fast nearest neighbor search.

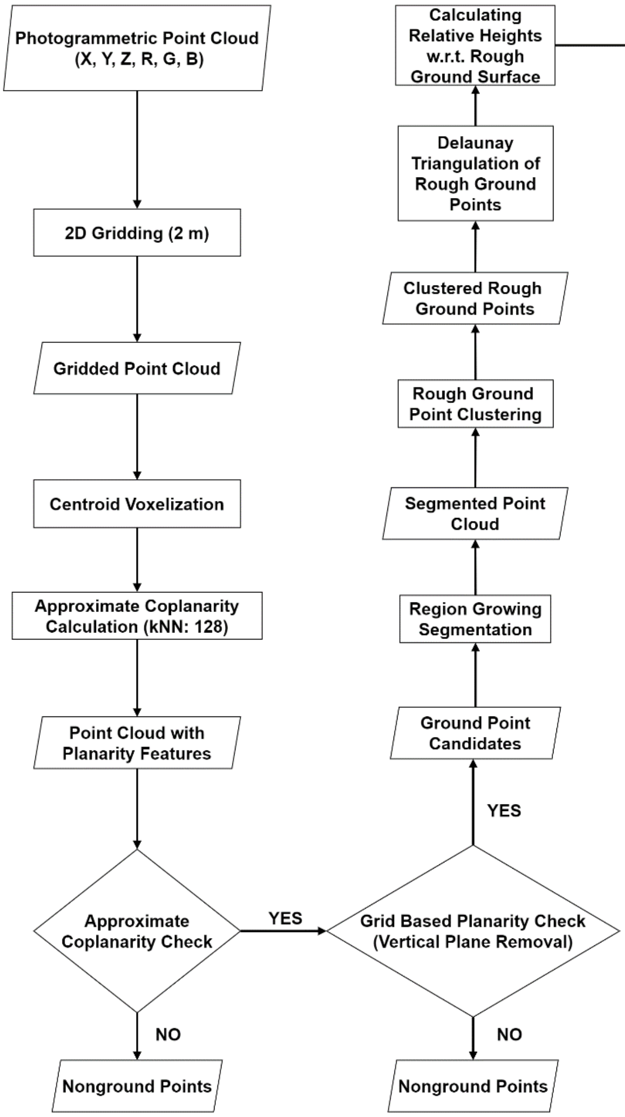


Figure 37: The flowchart of the proposed algorithm

3.2.2 Voxelization

Since raw point clouds are unstructured, they have limited use beyond visualization and simple distance measurements; thereby, a considerable effort has been made to create automated techniques for point cloud interpretation (Hackel, Wegner, & Schindler, 2016). In other words, the distance between two random adjacent points might not be equal, and as a result, spatial algorithms may perform poorly in some datasets. A voxel is a three-dimensional equivalent of a pixel (volumetric pixel or volumetric image element) (Pyysalo, Oksanen, & Sarjakoski, 2009). A voxelization process is applied to overcome this unstructured data problem. In this study, the covariance features are calculated primarily based on nearest neighborhood queries. Voxelization not only helps these queries be faster since it makes the data organized compared to raw data, but also reducing number points that allows neighborhood queries and algorithms to work faster. In Figure 38, the raw point cloud, and in Figure 39, voxelized point cloud are shown. For the voxelization process, at the chosen cell size (0.25 m), the input point cloud is split into 3D voxels. Each voxel is represented with a point that satisfies the given conditions below. For the voxelization, the following ruleset is applied by using PDAL "voxelcentroidnearestneighbor" filter:

1. The input point cloud is split into 3D voxels for the chosen cell size (0.25 m).
2. If the given cell size contains only one point, this point is used directly.
3. The point closest to the voxel center is selected if the given cell size has two points.
4. If there are more than two points, first, the centroid is calculated for the points in the given cell size. Then, the closest point to the centroid is selected (PDAL Contributors, 2018).

In Table 8, the comparison between the raw point and the voxelized point cloud is shown. Figure 42, Figure 43, Figure 44, Figure 45 are the spherical volume density (r = 1 m), which is a measure of the point count in the given spherical volume, histograms of Ankeny raw data, Ankeny voxelized data, Cadastre raw data, and Cadastre voxelized data, respectively. Spherical volume density is the value of the number of neighbors divided by the neighborhood volume in a given unit sphere (r = 1 m).

Table 8: Comparison between raw and voxelized data

Dataset	**Input Number of Points**	**Voxelized Number of Points**	**Voxel Cell Size (m)**	**Raw Point Spherical Volume Density Mean (1 m^3)**	**Voxelized Point Spherical Volume Density Mean (1 m^3)**
Ankeny	8924117	776465	0.25	126.703	8.590
Cadastre	5771358	2641573	0.25	20.101	6.972

Figure 38: Ankeny raw point cloud data

Figure 39: Ankeny voxelized point cloud data

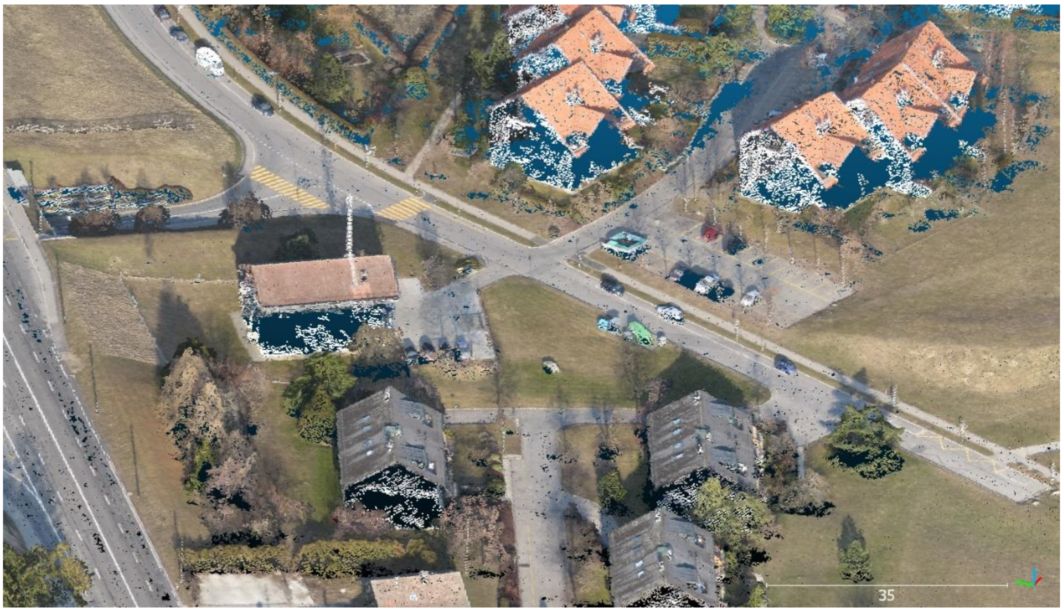


Figure 40: Cadastre raw point cloud data

Figure 41: Cadastre voxelized point cloud data

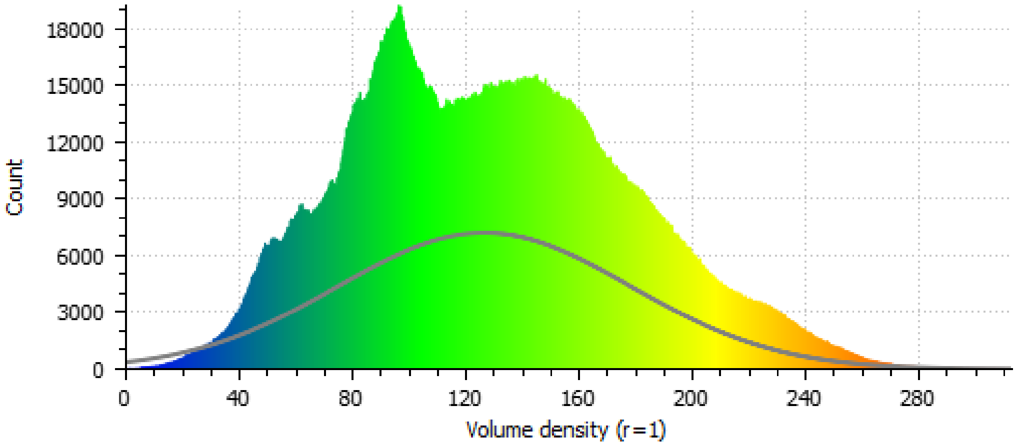


Figure 42: Histogram of spherical volume density (r =1 m) in Ankeny raw data (Gaussian Mean = 126.703, Standard Deviation = 51.732)

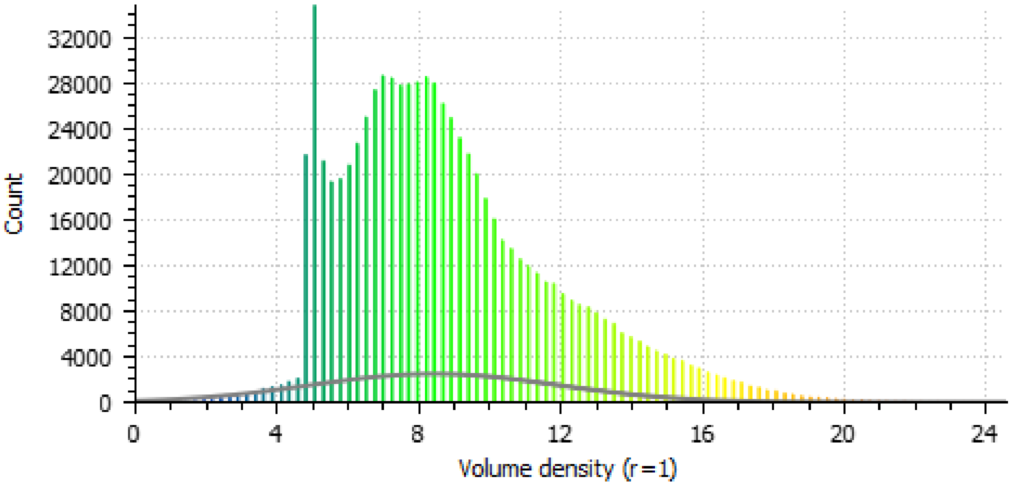


Figure 43: Histogram of spherical volume density (*r* =1 m) in Ankeny voxelized data (Gaussian Mean = 8.590, Standard Deviation = 3.432)

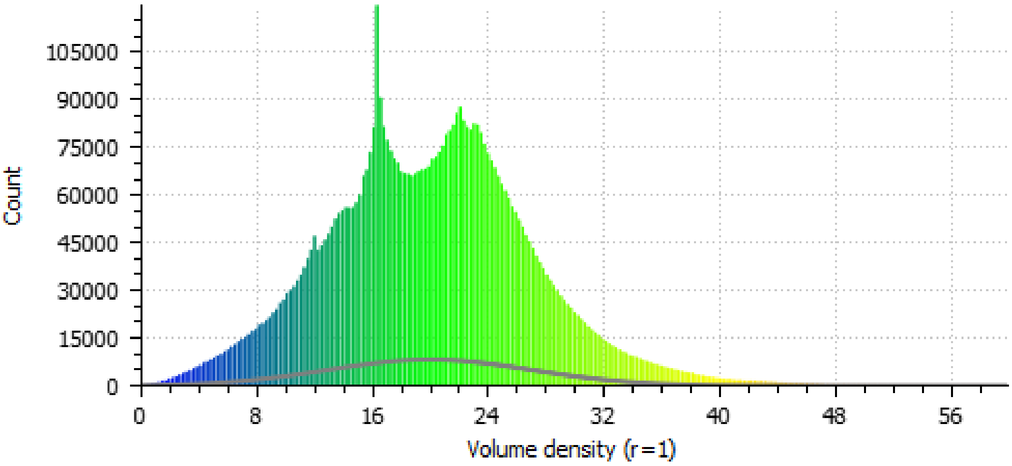


Figure 44: Histogram of spherical volume density (r =1 m) in Cadastre raw data (Gaussian Mean = 20.101, Standard Deviation = 7002)

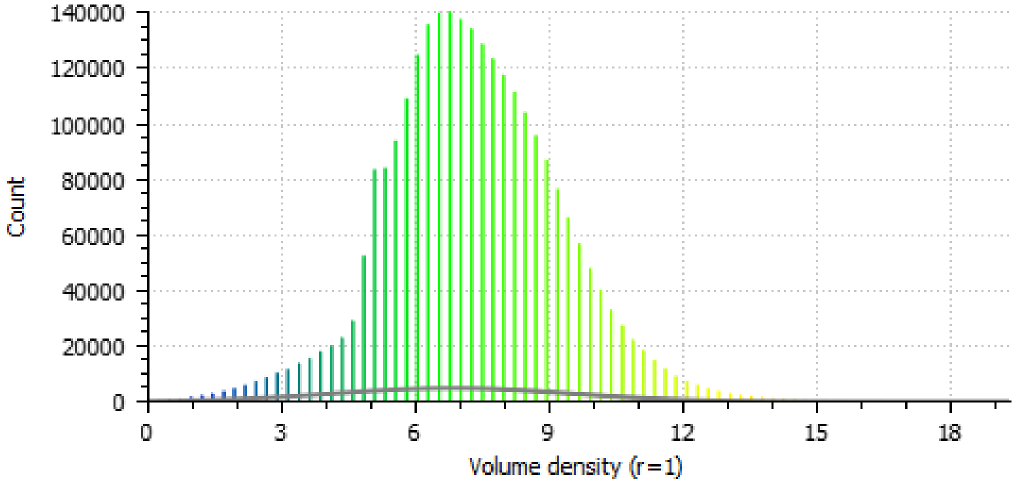


Figure 45: Histogram of spherical volume density (r =1 m) in Cadastre voxelized data (Gaussian Mean = 6.972, Standard Deviation = 2.593)

3.2.3 Approximate Coplanarity Check

In this thesis, a divide-and-conquer approach is used to split this problem into simpler steps. First, by using an approximate planarity check, the data is divided into two main groups namely, ground and non-ground candidates. In this thesis, a planarity-based divide-and-conquer algorithm is utilized. It is assumed that ground candidates often show a planar characteristic, whereas above-ground objects distort this assumption. After voxelization, as a first step of the divide and conquer approach, all the points are labeled as planar or nonplanar using an approximate coplanarity check. This allows the algorithm to eliminate most of the above-ground objects such as vehicles, posts, trees, etc., since they show a nonplanar behavior within the neighborhood. However, since building facades and roofs show a planar behavior, it is not possible to filter these points in this step. This check also helps us eliminate outliers that do not fit to plane in a given size. Although an approximate planarity check can help us to filter above-ground objects, these points are not removed in this step. These points help the algorithm to calculate a grid-based planarity to get rid of the vertical planes in the following algorithm step. The approximate planarity check is applied to the voxelized data by checking 128 nearest neighborhoods using the approximate planarity algorithm explained in Section 2.7, which is efficient and fast since it is designed initially for real-time plane detection in unorganized point clouds. Different neighborhood sizes (8, 16, 32, 64, 128, 256) have been tried, and it is decided to check 128 neighbors because if the size decreases, there may be some leftovers from the approximate planarity check. On the other hand, when the size increase, the object boundaries are extended, and ground points near the objects are labeled nonplanar. The approximate planarity check results with neighborhood sizes of 8, 12, 16, 32, 64, 128, 256 are illustrated in Figure 46, Figure 47, Figure 48, Figure 49, Figure 50, and Figure 51, respectively. Note that the red points show the approximate planar samples, whereas the blue shows the non-planar examples. If the neighborhood size decreases too much, some walls are considered nonplanar due to noisy data. There is a trade-off between the computation time and the robustness; however, since the approximate coplanarity algorithm is designed to work in real-time, for the sake of robustness, it is chosen to check 128 neighbors that do not cost a significant computation time. These coplanar points primarily represent the facades, the ground, and the roofs. Figure 52 shows a section of the RGB-colored voxelized Ankeny dataset that contains trees, buildings, vehicles, posts; Figure 53 indicates the coplanarity result, and Figure 54 shows the result after filtering on the Ankeny dataset. Figure 55, an area of the RGB-colored voxelized Cadastre dataset with the same features, in Figure 56, the coplanarity filter result, and in Figure 57, the filtered points are shown, respectively.

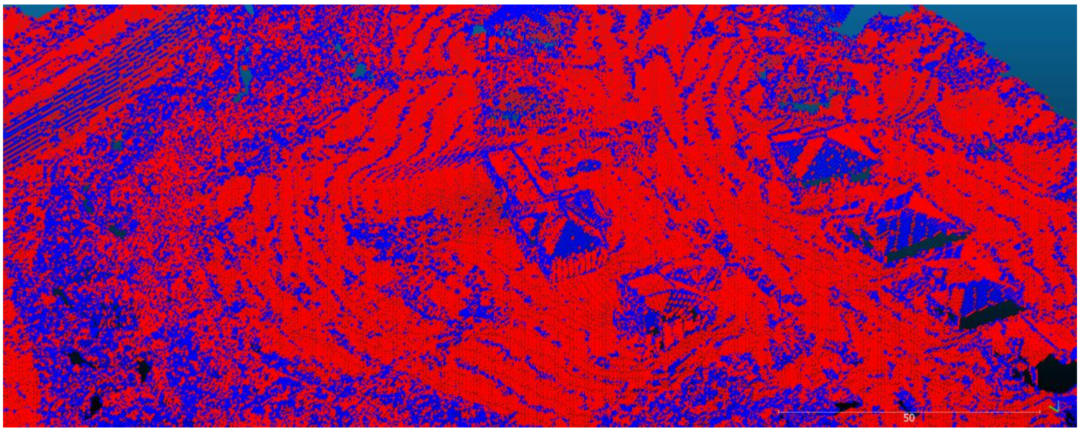


Figure 46: Approximate coplanarity check with 8 neighbors

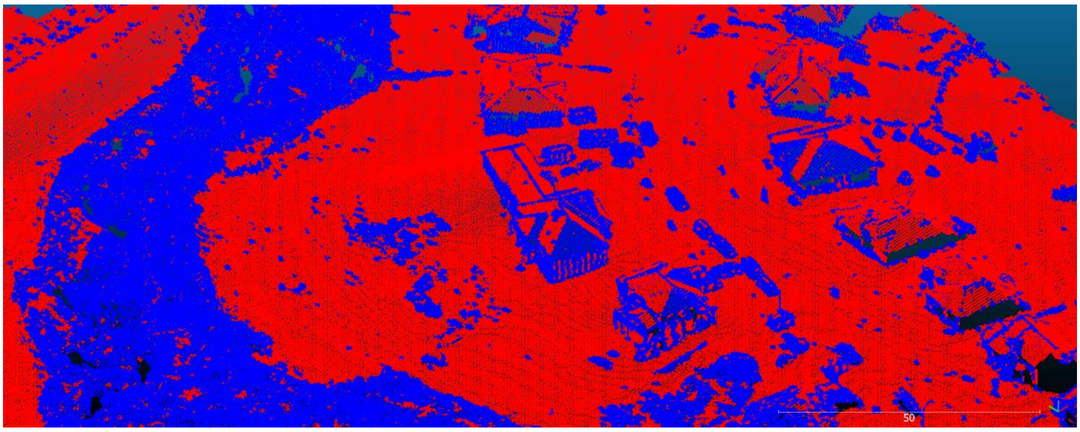


Figure 47: Approximate coplanarity check with 16 neighbors

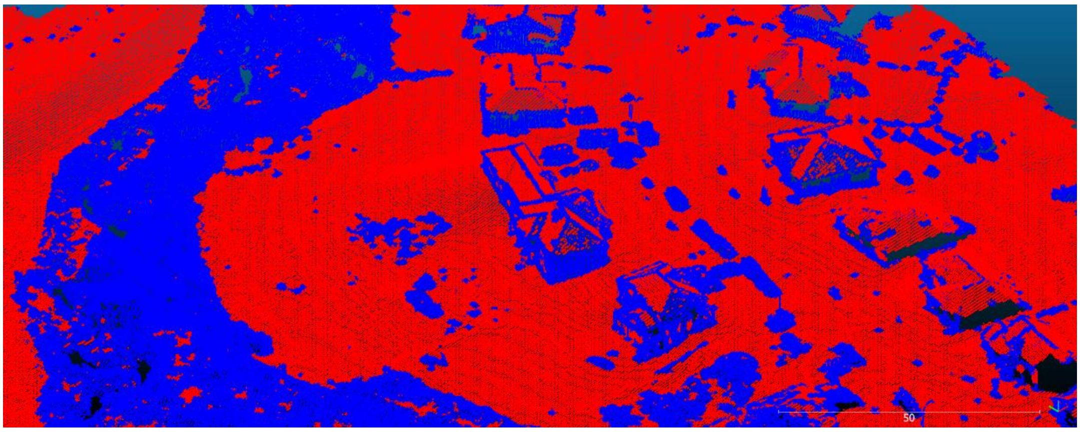


Figure 48: Approximate coplanarity check with 32 neighbors

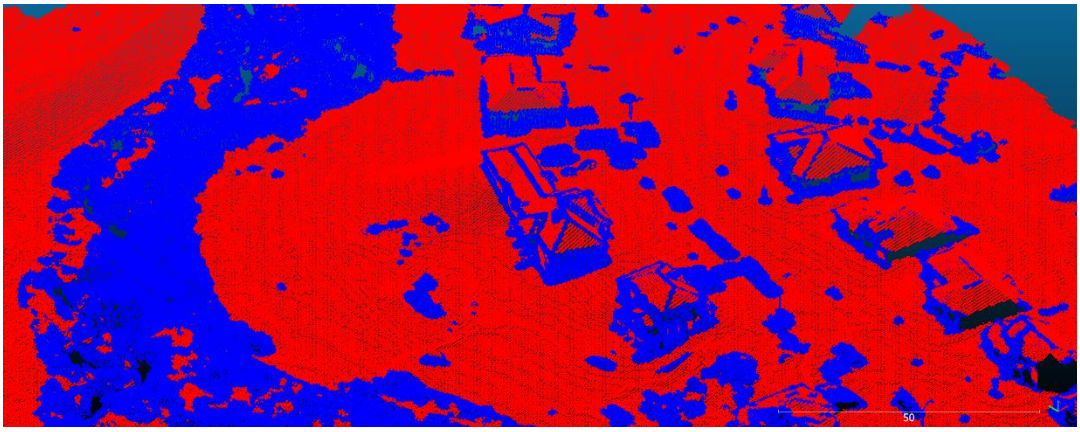


Figure 49: Approximate coplanarity check with 64 neighbors

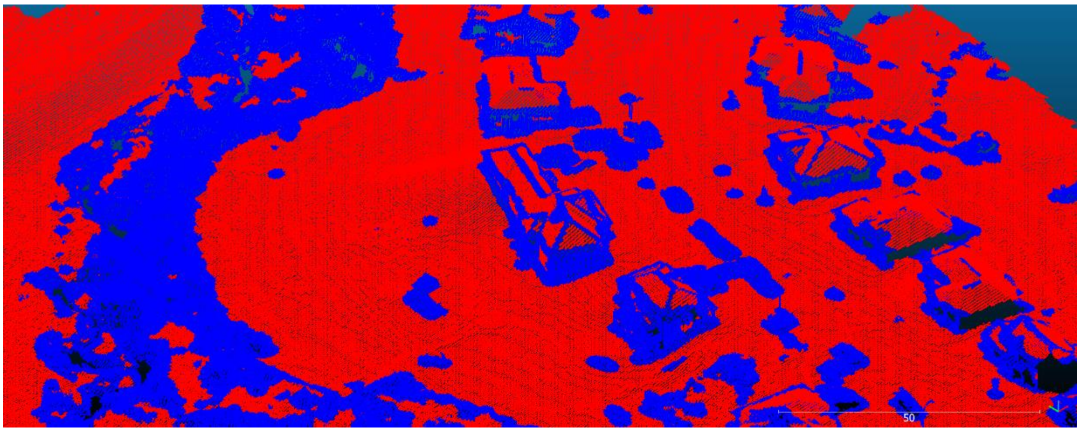


Figure 50: Approximate coplanarity check with 128 neighbors

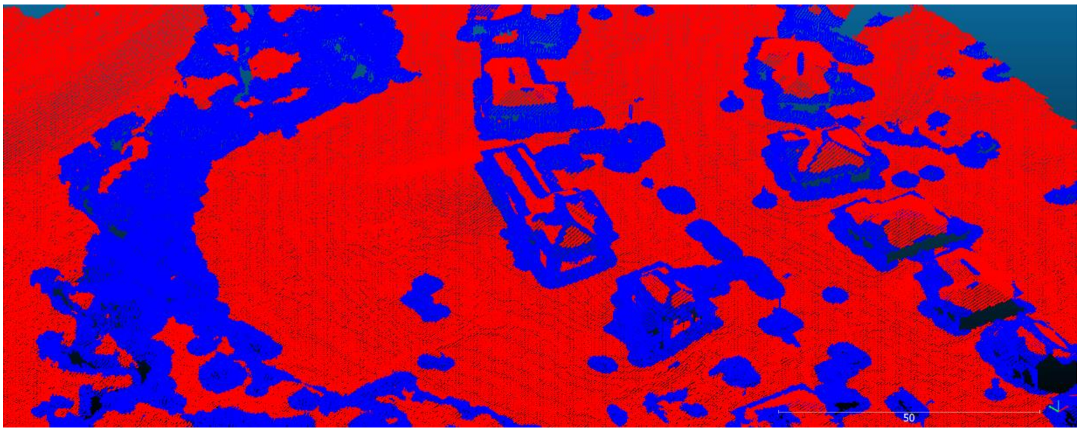


Figure 51: Approximate coplanarity check with 256 neighbors

Figure 52: Voxelized Ankeny dataset

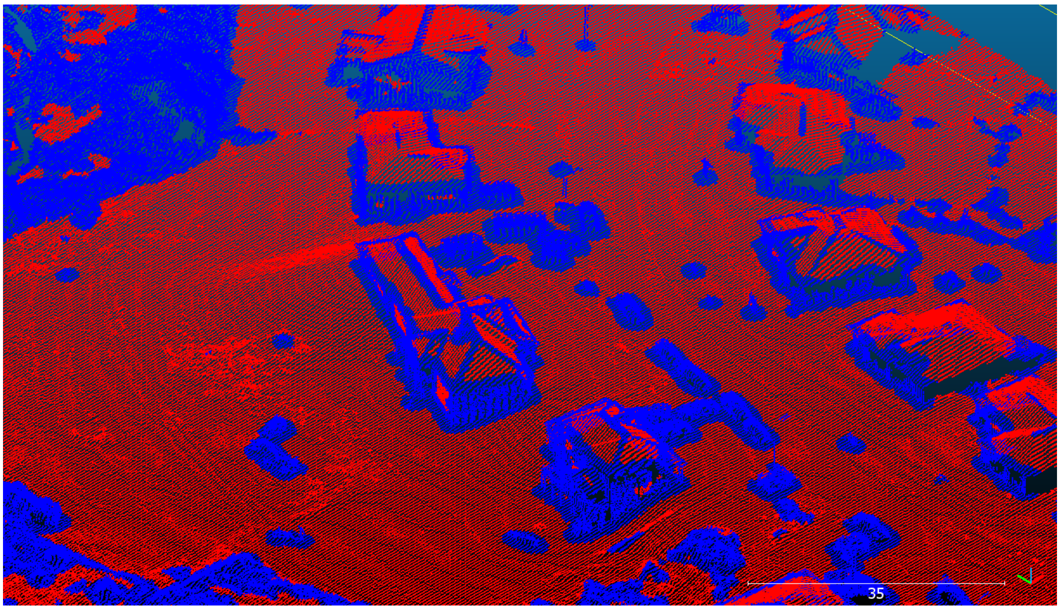


Figure 53: Approximate coplanarity check result on Ankeny dataset. Red points represent the approximate planar points, and blue points are the non-planar ones.

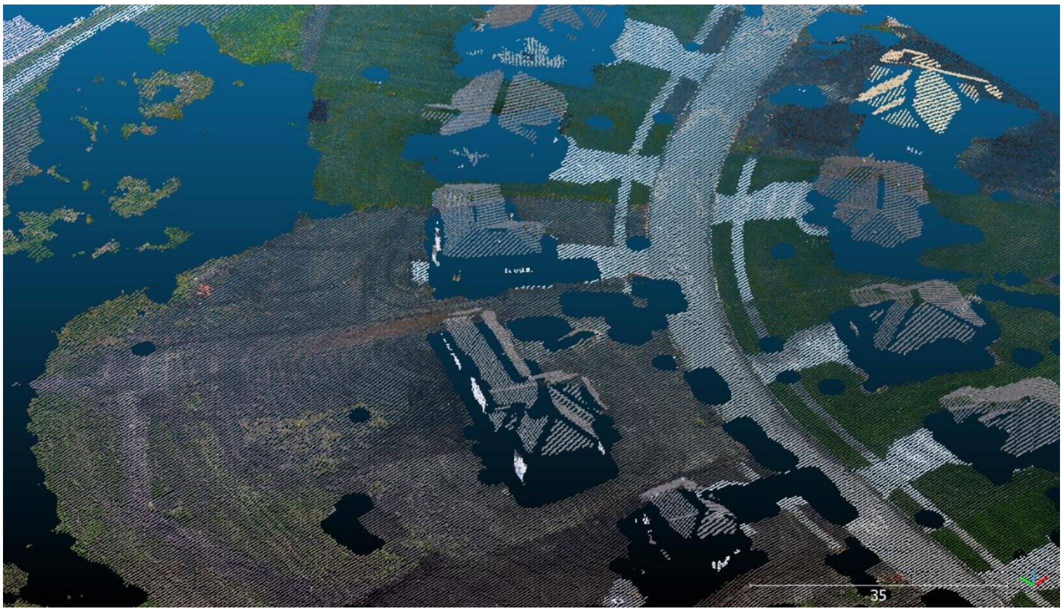


Figure 54: Ankeny dataset after approximate coplanarity filter is applied

Figure 55: Voxelized Cadastre dataset

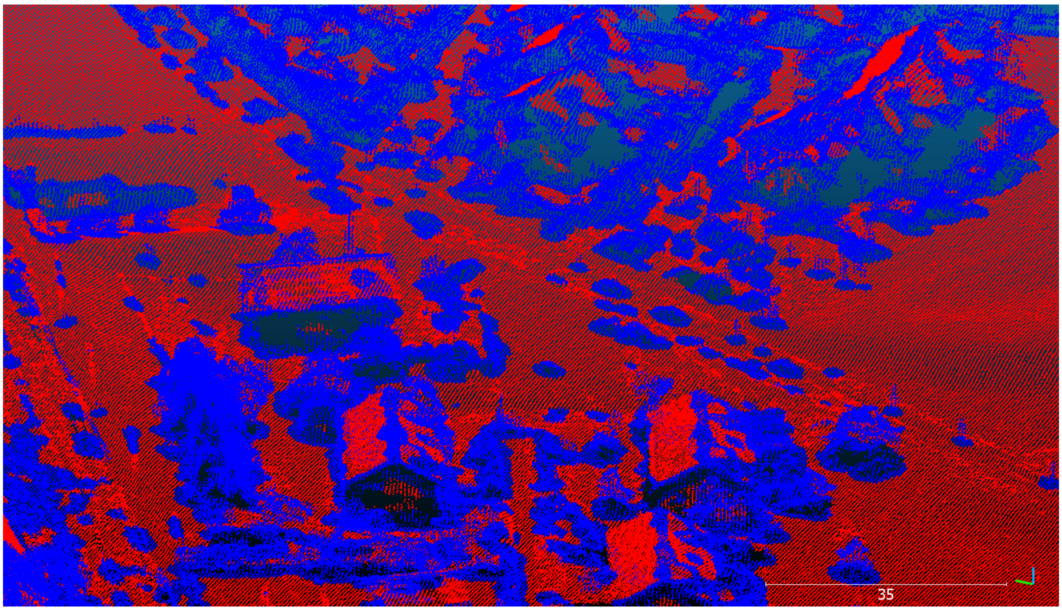


Figure 56: Approximate coplanarity check result on Cadastre dataset. Red points represent the approximate planar points, and blue points are the non-planar ones.

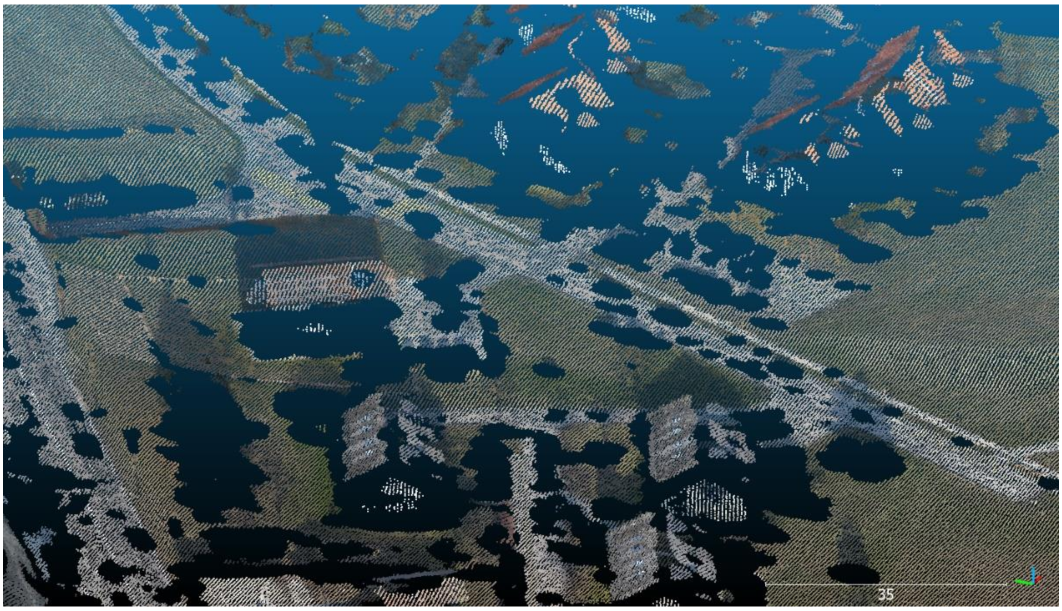


Figure 57: Cadastre dataset after approximate coplanarity filter is applied

3.2.4 Vertical Plane Removal

After an approximate planarity check, most of the non-planar regions are detected. Building facades, vertical planes, roof planes, and other non-ground areas should be removed so that the ground points are not segmented together with the ground points in the following region growing segmentation. A grid-based planarity method is designed to get rid of the vertical planes. It is valuable for detecting vertical planes and beneficial to preserve some contextual information since some features might be misclassified in the nearest neighborhood approach.

In the grid-based planarity check, the data is divided into 2D grids. Then for these grids, the eigenvalues of the covariance matrix are calculated together with the approximate coplanarity filter. The planarity value of the grids lower than 0.3 are not considered as the ground planes because they contain both ground and vertical plane samples. The grid size is selected as 2 m, so these grids have multi-class samples (both ground and building facade points). In Figure 59, the grids colored as blue have a grid planarity value less than the given threshold. It can be seen that near the building footprints, these grids contain some ground points as well. Since removing these points do not have a significant impact on the resulting DTM, they are eliminated as well. In Figure 58, a single building is shown in RGB colors, and in Figure 59, the grid planarity values are calculated for 2 m grids. It can be seen in Figure 59 that the grids having points from different classes (generally the building boundaries) have high grid planarity values. Then from these eigenvalues, the planarity values for these grids can be obtained. If $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$, the planarity value is defined as:

$$(\lambda_2 - \lambda_3)/\lambda_1 \tag{8}$$

Because, in a planar distribution, the first two eigenvectors are expected to define the plane, and corresponding eigenvalues are expected to be large, while the third component is perpendicular to the plane and the variance on that direction (hence eigenvalue) is expected to be just the noise. The grids with a planarity value lower than 0.3 and the approximate planar points in the previous step are filtered out together in this step. In Figure 60, the approximate planarity filtered Ankeny data, and in Figure 61, the grid-based filtered Ankeny data are illustrated. The same results are shown for Cadastre data in Figure 62 and Figure 63, respectively.

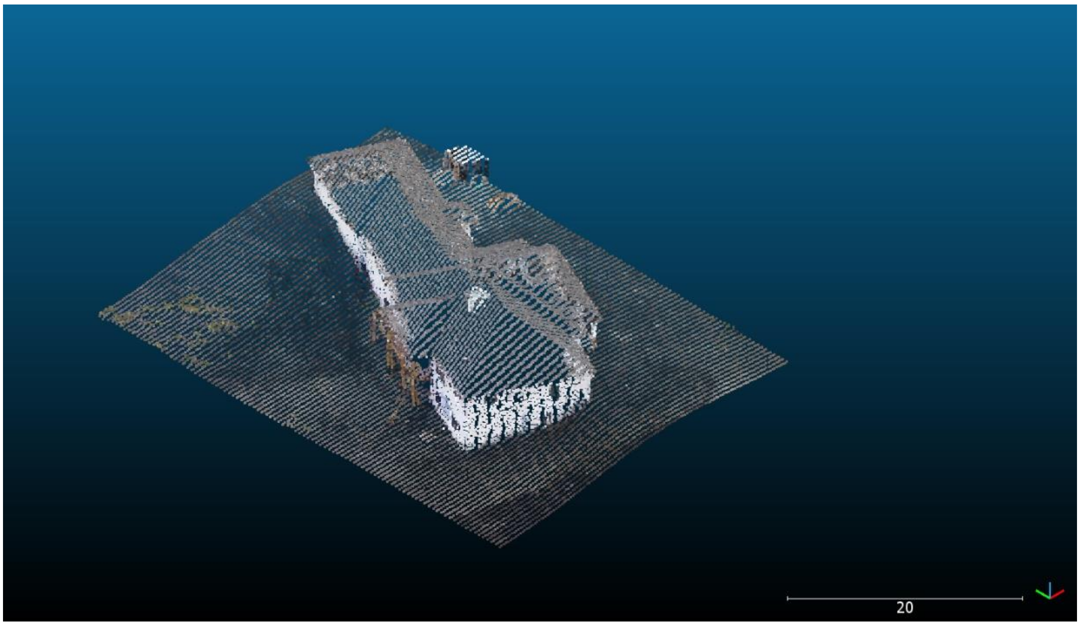


Figure 58: A single building colored in RGB

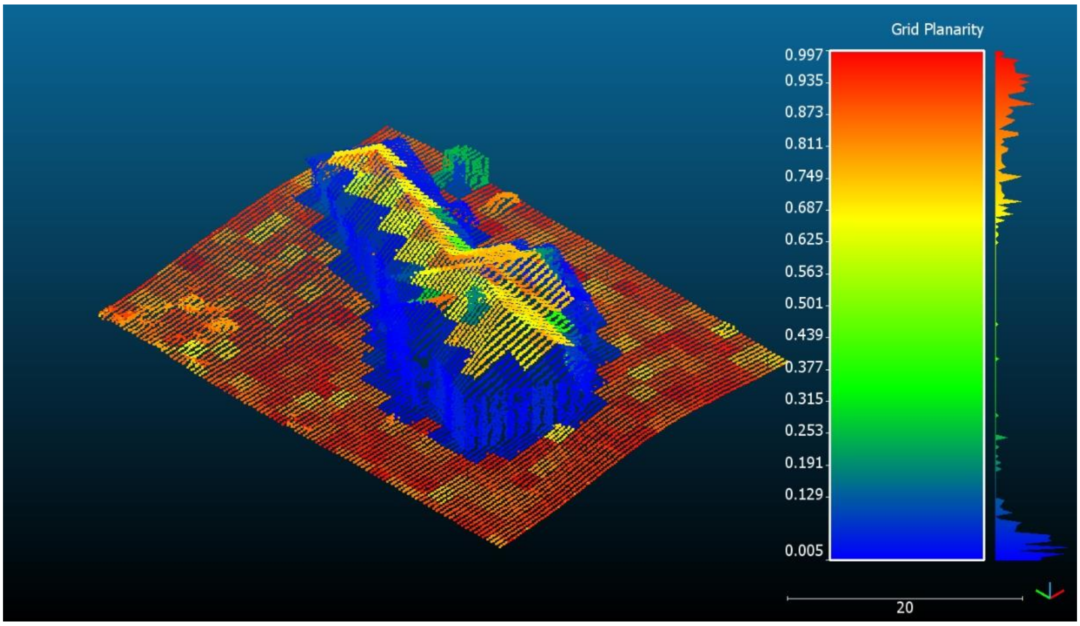


Figure 59: Grid planarity values on a single building

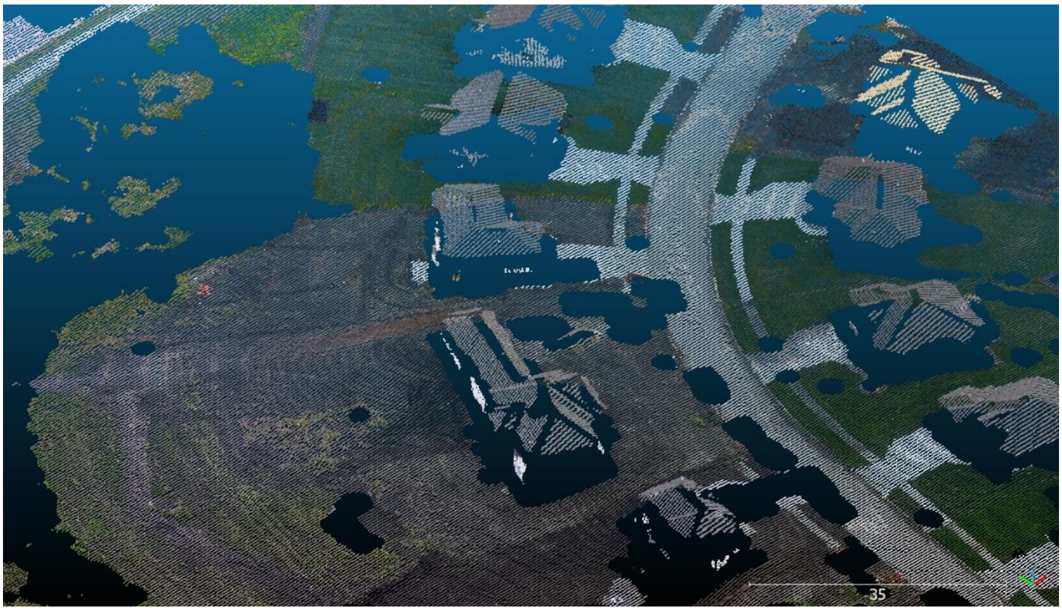


Figure 60: Ankeny dataset after approximate coplanarity filter is applied

Figure 61: Eliminating vertical planes in Ankeny dataset using grid-based planarity values

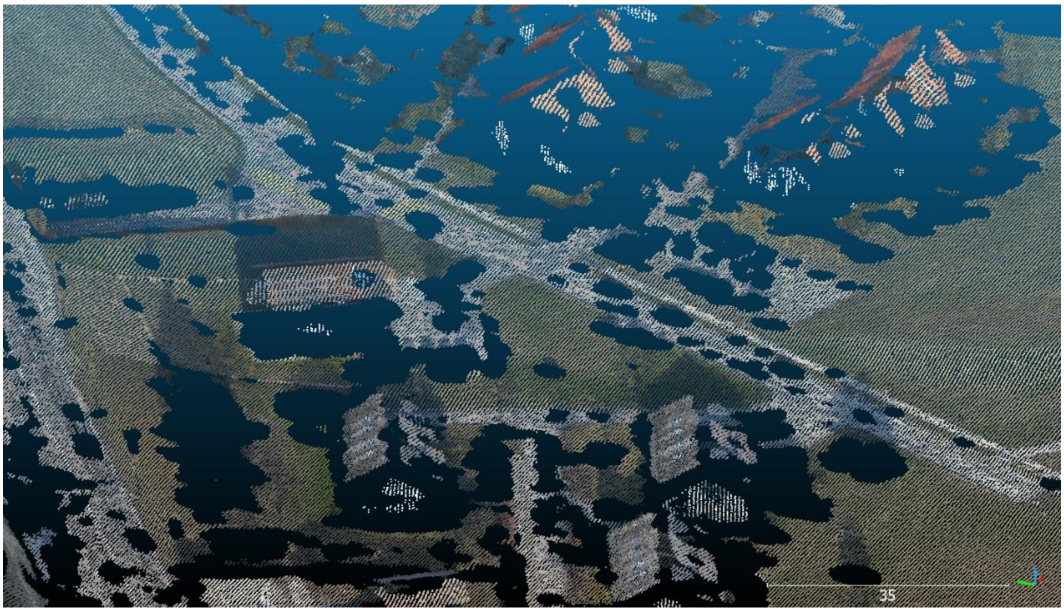


Figure 62: Cadastre dataset after approximate coplanarity filter is applied

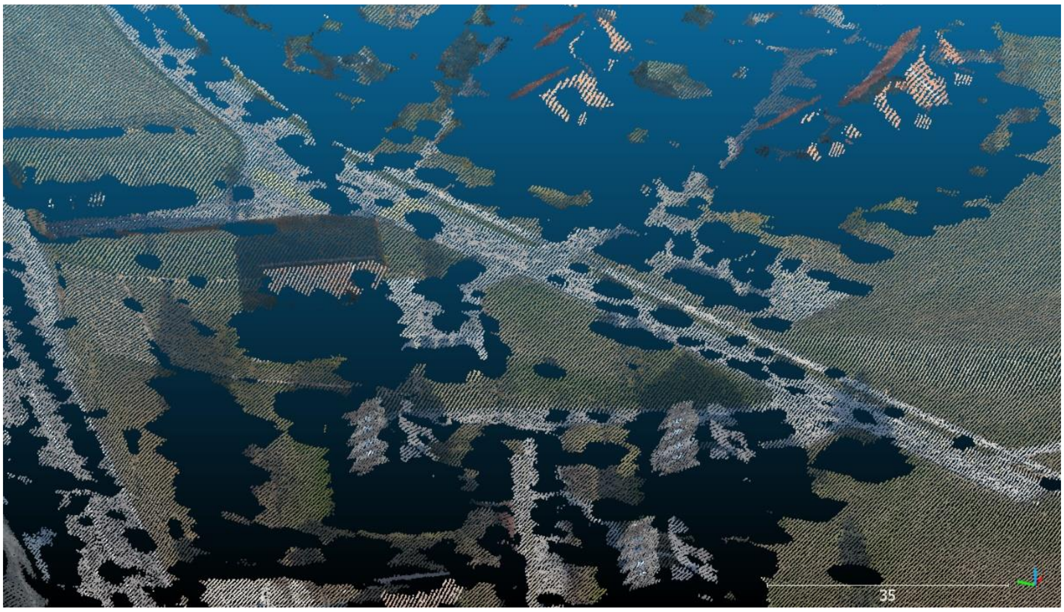


Figure 63: Eliminating vertical planes in Cadastre dataset using grid-based planarity values

3.2.5 Region Growing Segmentation

The principles of the region growing segmentation are discussed in section 2.8. For region growing segmentation, Point Cloud Library (PCL) Strawlab Python Implementation is used.

In Figure 64 and Figure 65, random RGB-colored segmentation results of two datasets are shown for Ankeny and Cadastre datasets, respectively. It can be seen that, although there are small segments representing terrain, most of the ground points are segmented into a big piece that contains enough points to estimate a rough ground surface.

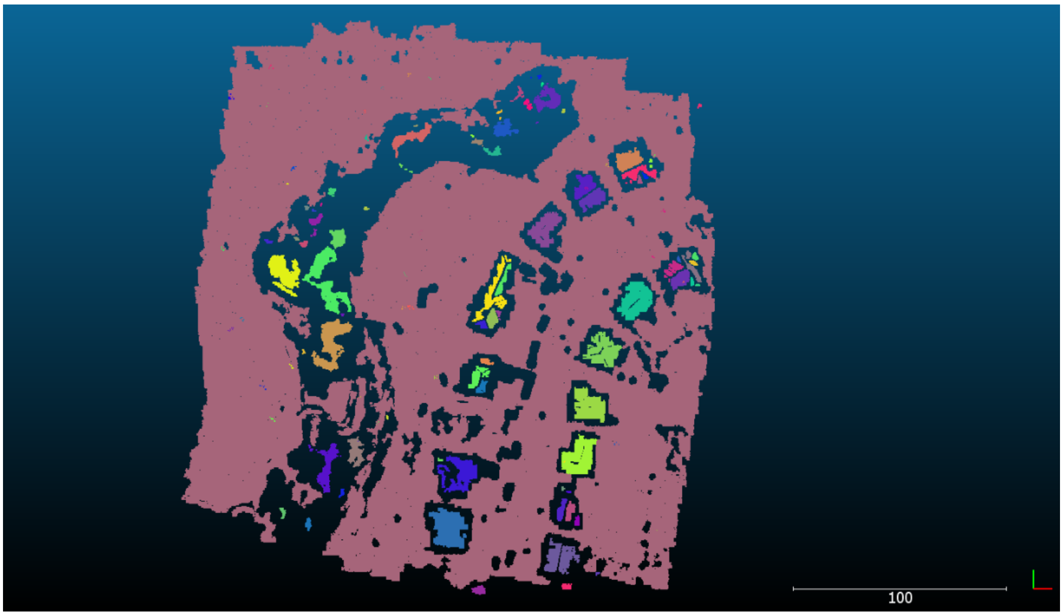


Figure 64: Top view of random RGB-colored segmentation result of Ankeny dataset

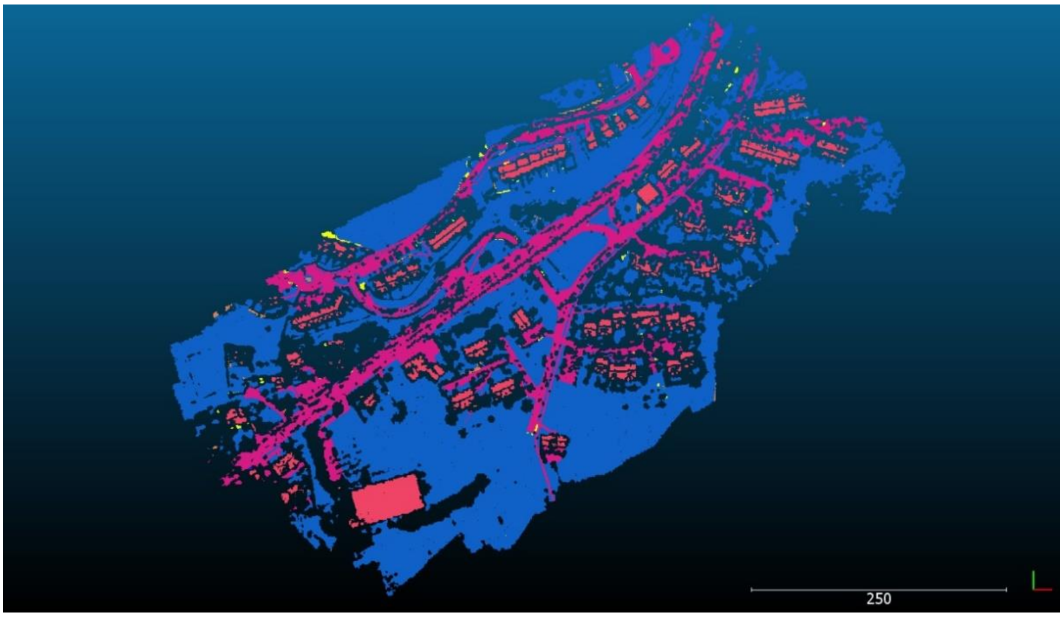


Figure 65: Top view of random RGB-colored segmentation result of Cadastre dataset

3.2.6 Delaunay Triangulation and Rough Relative Height Calculation

The input point clouds store elevation as absolute values. Since there is no prior DTM, there is no information about "Height Above Ground (HAG)" in the input dataset. However, after segmentation, the most significant segments for each dataset are considered to estimate a rough ground surface to make the height-based elimination process available. If the area is not dense urban area, the biggest segment is expected to belong to the ground samples. PDAL offers “hag_delaunay” filter that takes ground points into account and creates a surface. In this step, the segments with the highest number of points are considered as the ground points to estimate the rough surface. Later, the distances are calculated from each point to obtain HAG values. By doing so, it is possible to eliminate the small clusters (roof parts, etc.) in the next step based on HAG values. Absolute height values might misguide since the elevation change may occur due to terrain. To overcome this problem, the relative height values are taken into account in the further steps.

3.2.7 Eliminating Roof Planes

After calculating HAG values, it can now be estimated whether a point is close to the ground or not. HAG-based thresholding is applied to eliminate the points that are 0.5 m distant from the rough ground surface. In Figure 66 and Figure 67, the points are colored by HAG values for Ankeny and Cadastre datasets, respectively. In Figure 66 and Figure 67 the negative values represent the points beneath the rough surface, and within 0.5 m threshold, the points are considered ground points. In other words, if the HAG value is greater than -0.5 m and less than 0.5 m, these points are labeled as ground points.

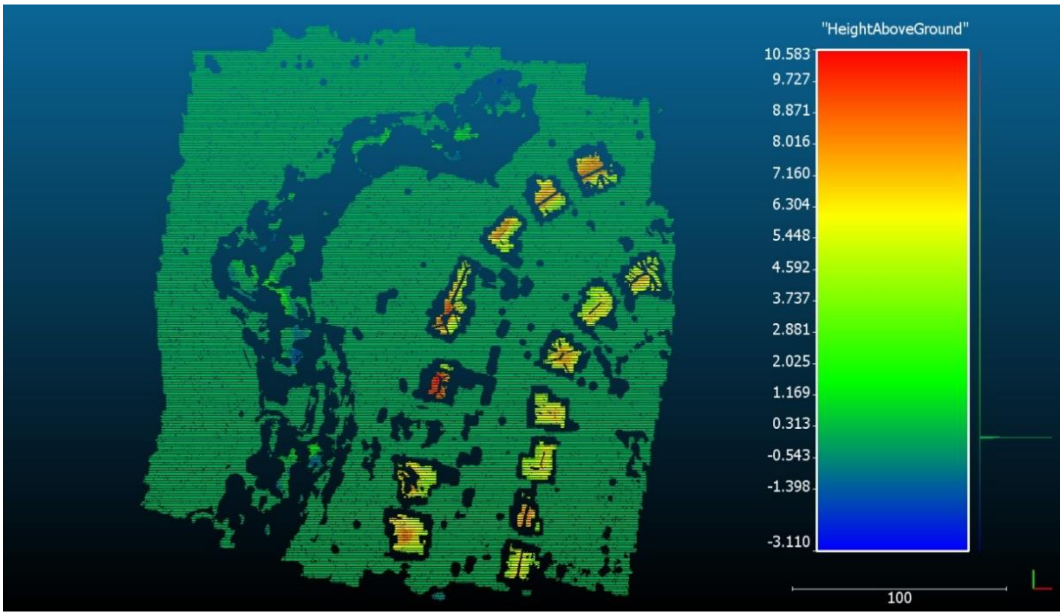


Figure 66: Ankeny dataset is colored based on HAG values

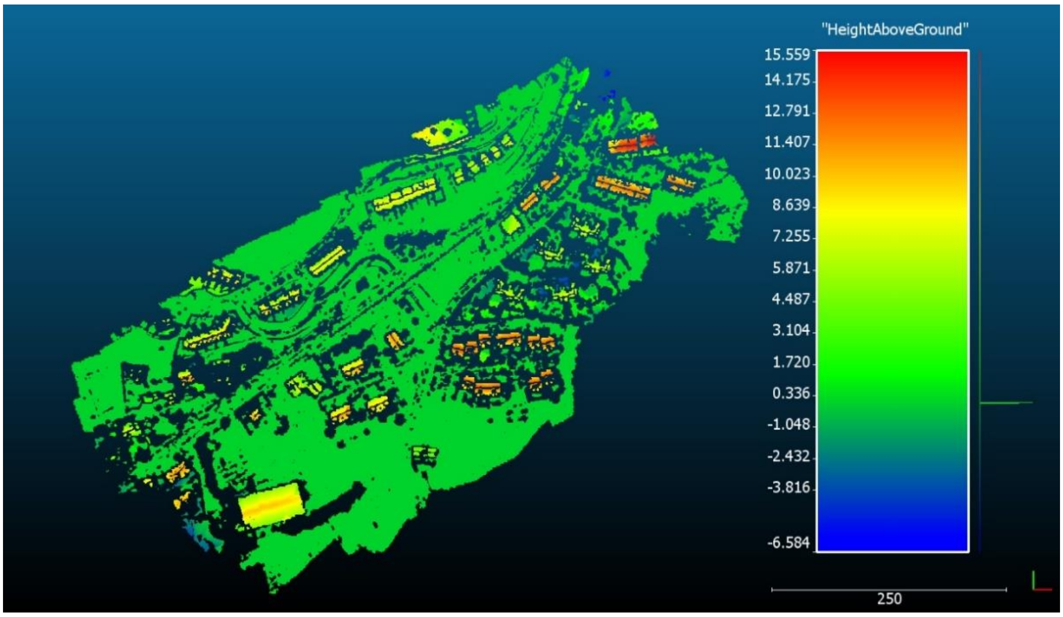


Figure 67: Cadastre dataset is colored based on HAG values

3.2.8 Rasterization

After HAG-based filtering is applied, the only points left are the ground points. By the IDW interpolation method, a surface is interpolated considering using 12 neighboring points and a power of 2. The IDW is selected because of its simplicity and speed. The increase in the number of neighboring points causes a huge increase in computation time, so due to limited computational power, and our aim of having a fast algorithm, 12 neighbors are considered. The interpolation is done in ArcGIS desktop. The input datasets do not have regular boundaries, and some data gaps around the borders might cause errors in accuracy assessment. The distorted areas are clipped from the resulting raster to have better accuracy assessment. For better visualization, the RGB-colored clipped rasters are shown in Figure 68, and Figure 69, for Ankeny and Cadastre datasets, respectively.

Figure 68: RGB-colored clipped Ankeny dataset

Figure 69: RGB-colored clipped Cadastre dataset

3.2.9 Parameter Selection

The proposed algorithm contains various parameters in processing steps and this section describes how these values are selected. For varying parameter and threshold values, the remaining parameters kept constant. The parameters are tested, and selected with respect to Ankeny dataset. Later, these parameters are tested for Cadastre dataset to check the algorithm’s robustness in different point density.

Table 9: CSF parameters

	Terrain Type	**Cloth Resolution (m)**	**Maximum Iteration**	**Classification Threshold**
Ankeny	Flat	1	500	0.5
Cadastre	Steep Slope	1	500	0.5

3.2.9.1 Grid Resolution

In the proposed algorithm, the grids filter out the vertical planes, especially the building facades. Because of this reason, it is better to select a grid resolution bigger than the building wall thickness to cover these facade points as much as possible in a single grid. The change in this value does not significantly affect the RMSE of the DTM; so, this value is selected as 2 m to filter out the building walls and vertical planes. Any other step except grid planarity check uses kNN search. For these queries, the neighborhood size is fixed.

On the other hand, the number of points inside the grids is critical for calculating covariance features. The more points inside the grids help us to understand the geometric properties of the grid better (planar or nonplanar). For this step, both datasets, the relation between grid size, grid point count, are evaluated. Table 11 shows the relation between the grid size, and the point count.

Table 10: Grid size vs. RMSE for Ankeny dataset

Grid Size vs. RMSE				
Grid Size	1 m	**2 m**	3 m	4 m
RMSE (m)	0.258199778	**0.262687771**	0.264310471	0.263813
Max. Height Difference (m)	2.67928	**2.68216**	2.80679	2.67928

Table 11: Grid size vs. point count

Grid Size vs. Point Count		
	Ankeny	Cadastre
Grid Size (m)	Grid Point Count	Grid Point Count
1	160.9921525	29.1104879
2	**632.6020415**	**114.1690174**
4	2464.544877	446.2505219
8	9513.984009	1724.852959

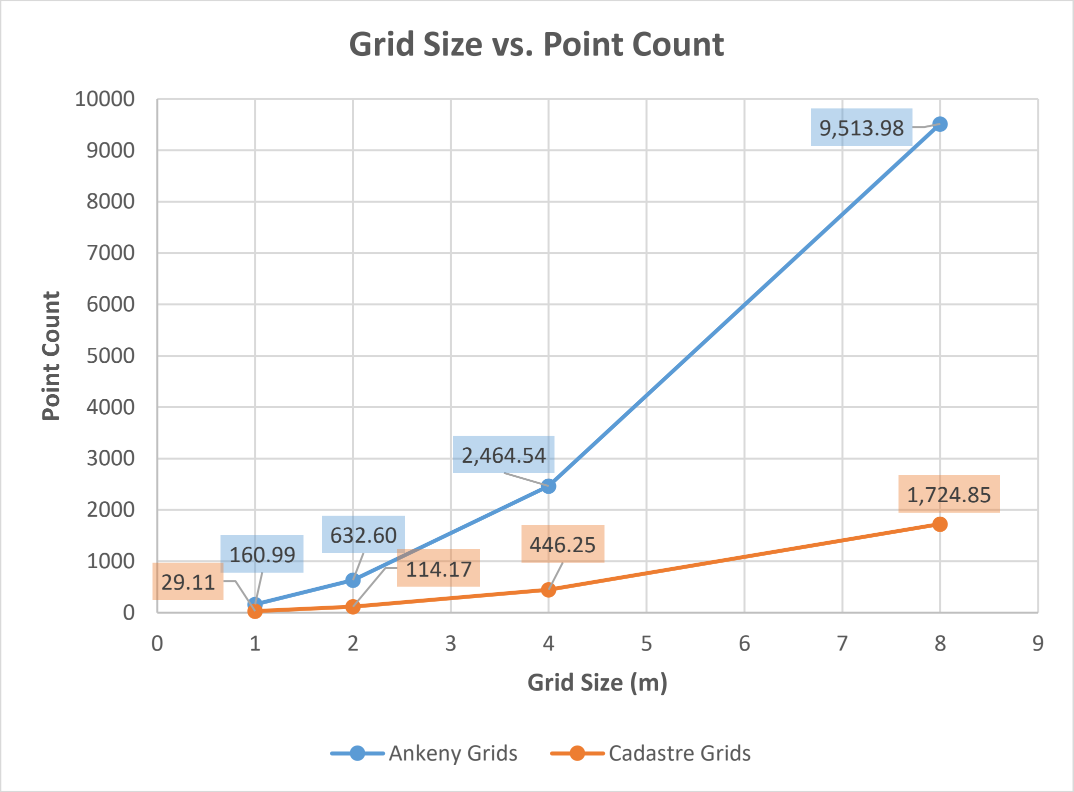


Figure 70: Grid Size vs. Point Count for both datasets

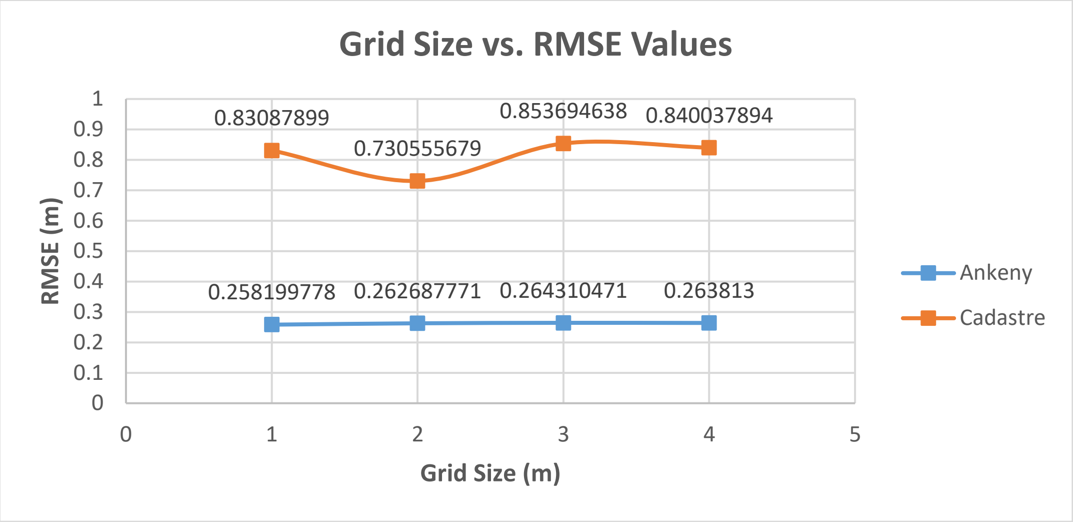


Figure 71: Grid Size vs. RMSE values for both datasets

Table 12: Grid Size vs. RMSE for both datasets

Ankeny Grid Size vs. RMSE				
Grid Size	1 m	**2 m**	3 m	4 m
RMSE (m)	0.2582	**0.262688**	0.264310471	0.263813
Max. Height Difference (m)	2.67928	**2.68216**	2.80679	2.67928
Cadastre Grid Size vs. RMSE				
Grid Size	1 m	**2 m**	3 m	4 m
RMSE (m)	0.830879	**0.730556**	0.853694638	0.840037894
Max. Height Difference (m)	6.01813	**4.71918**	7.40826	7.63757

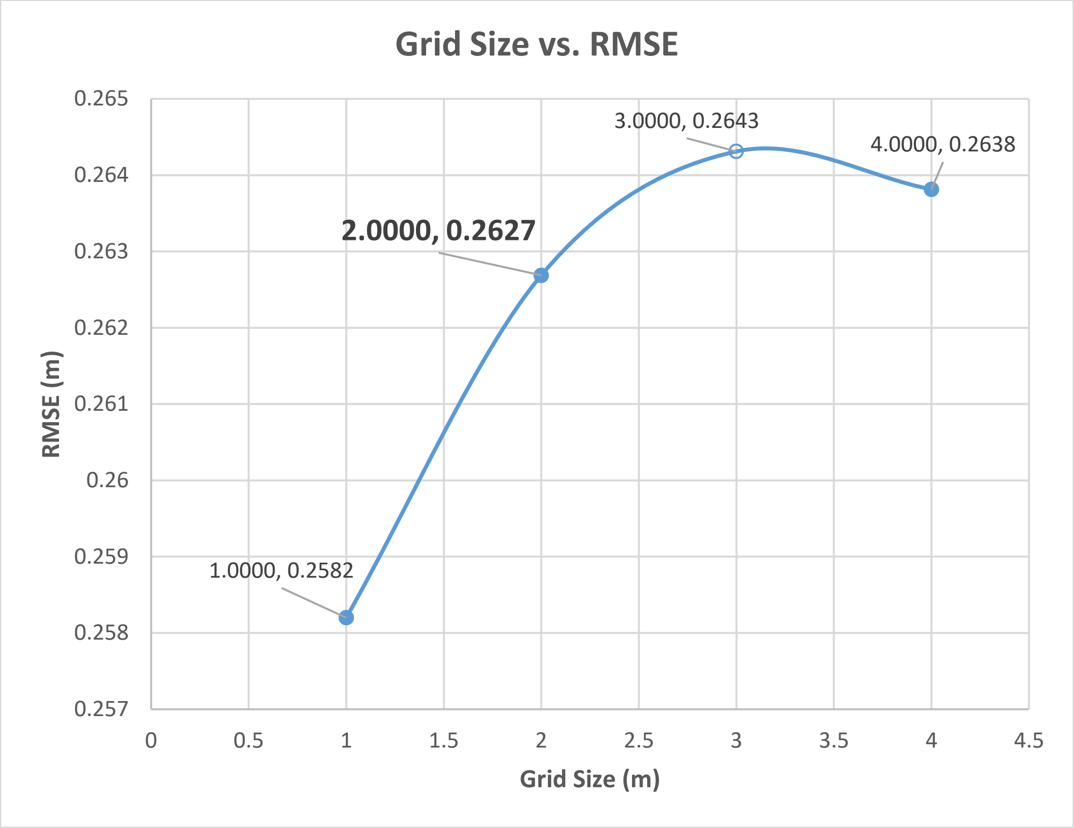


Figure 72: Grid Size vs. RMSE for Ankeny dataset

Table 12, Figure 70, Figure 71, and Figure 72 together indicate that a change from 1 m to 2 m grid does not significantly impact (around 4 mm) the RMSE value for Ankeny dataset. However, for Cadastre dataset, from 1 m to 2 m grid size, RMSE value, and maximum error decreased significantly (around 13 cm RMSE, and 2 m in maximum error). Although 1 m grid shows slightly better results in Ankeny dataset, 2 m grid is a much better solution for Cadastre dataset with less point density. 2 m grid also help to cover the walls, and ground points in the same grid, and helps to detect buildings facades more accurately.

3.2.9.2 Approximate Coplanarity Neighborhood

For the approximate coplanarity check, the information from neighboring points is needed. The approximate coplanarity algorithm is tested with varying kNN values. Up to 64 kNN, there are some vertical leftovers, so it is decided to check more neighbors. If kNN is 128, the proposed algorithm can filter out the nonplanar features. Between 128 and 256 neighbors, although RMSE values decrease, the change is not significant. Since kNN search is a costly operation, for the sake of computation time, the kNN size is selected as 128 for coplanarity check.

Table 13: Approximate Coplanarity kNN vs. RMSE for Ankeny dataset

Approximate Coplanarity vs. RMSE						
kNN	8	16	32	64	**128**	256
RMSE (m)	0.493466706	0.336471	0.280662261	0.277444	**0.262688**	0.261824
Max. Height Difference (m)	5.68613	5.81039	2.83746	3.10413	**2.68216**	2.63638

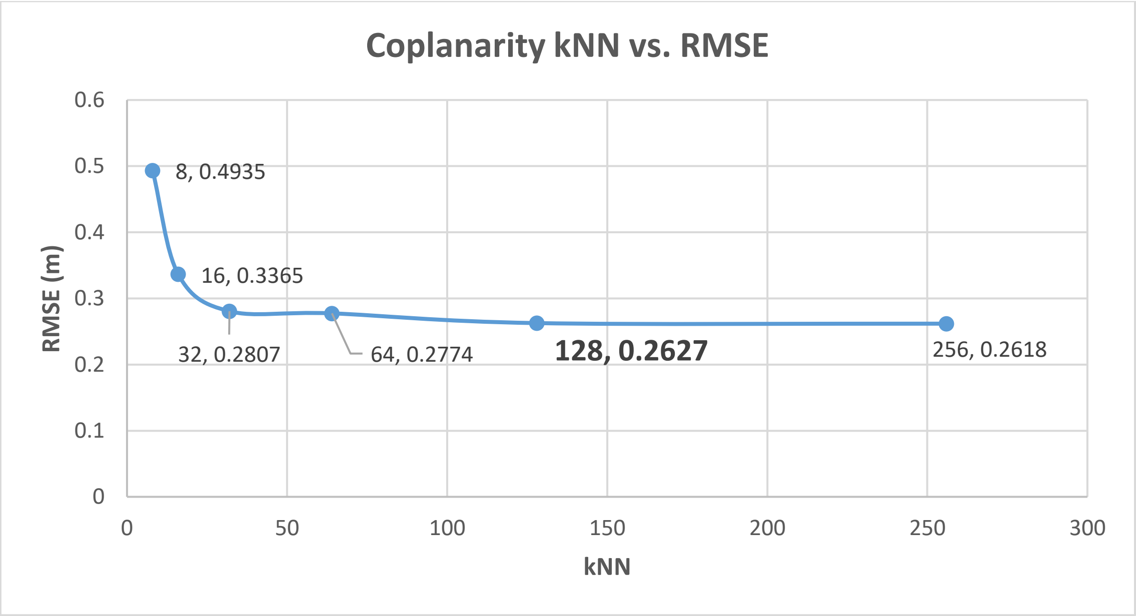


Figure 73: Approximate Coplanarity kNN vs. RMSE for Ankeny dataset

3.2.9.3 Grid Planarity Threshold

The grids used in this part is based on the grid size decided in Section 3.2.9.1. By using a grid-based threshold, it is possible to filter out the vertical planes. Although RMSE values are close in different threshold values, the maximum height difference varies. From 0.2 to 0.3, there is approximately a 10 cm change in the maximum height difference between ground truth and the extracted DTM. As a result, the grid planarity value is chosen as 0.3.

Table 14: Grid Planarity vs. RMSE for Ankeny dataset

Grid Planarity vs. RMSE				
Grid Resolution (m)	0.1	0.2	**0.3**	0.4
RMSE (m)	0.261701	0.262116	**0.262688**	0.262781
Max. Height Difference (m)	2.78195	2.78195	**2.68216**	2.68216

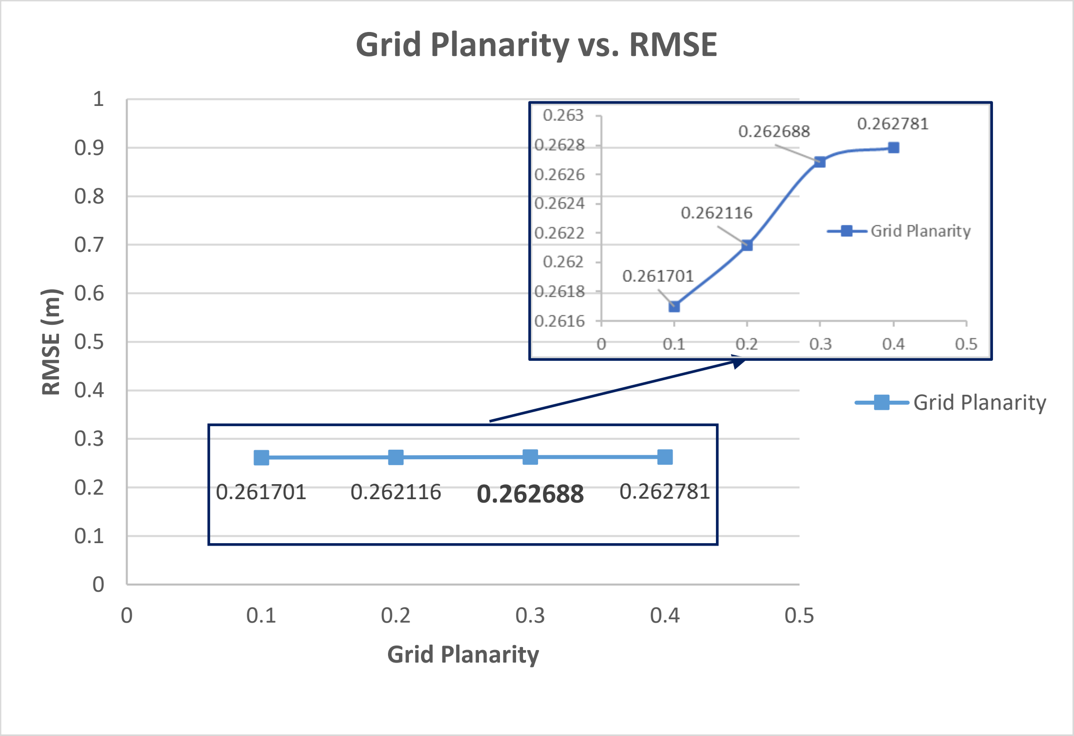


Figure 74: Grid Planarity vs. RMSE for Ankeny dataset

3.2.9.4 Region Growing Segmentation Parameters

Three different parameters for region growing segmentation are tested, namely, smoothness, curvature, and kNN.

- Smoothness: If the difference between the normals of the points is less than the smoothness threshold, they are thought to be in the same cluster.
- Curvature: The disparity between the curvatures of two points is checked if they have a slight normals deviation. If this number is lower than the curvature threshold, the algorithm will use the newly added point to continue cluster expansion.
- kNN: Number of neighbors to be checked

The smoothness value is selected as low as possible to segment the ground points with minor normal deviations. So, for this segmentation, it is aimed to keep the smoothness threshold as low as possible to segment ground points accurately for rough ground point estimation. There is a slight improvement in RMSE, and maximum height difference value between the ground truth and DTM in 0.2 radians; however, this improvement can be neglected.

Table 15: Region Growing Smoothness vs. RMSE for Ankeny dataset

Region Growing Smoothness vs. RMSE				
Smoothness (radian)	**0.05**	0.1	0.2	0.4
RMSE (m)	**0.262687771**	0.263641636	0.261275482	0.262107
Max. Height Difference (m)	**2.68216**	2.68216	2.67928	2.67928

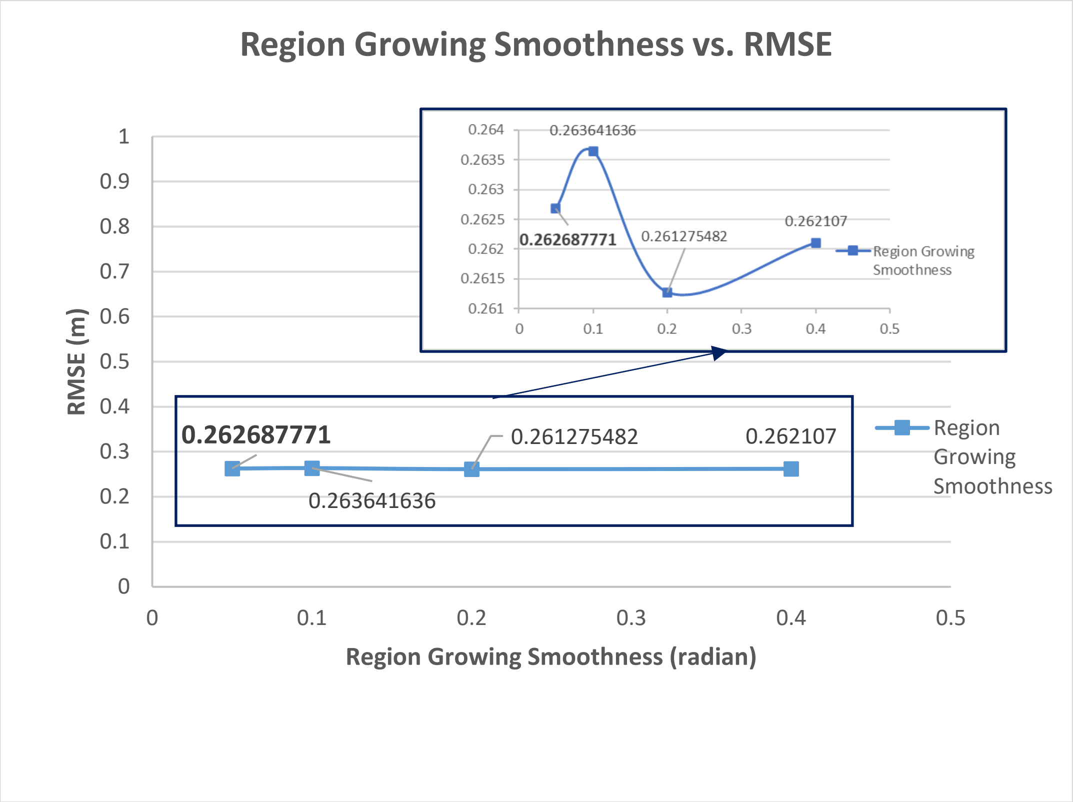


Figure 75: Region Growing Smoothness vs. RMSE for Ankeny dataset

Similarly, for the curvature value, there is a slight change in RMSE and maximum height difference values with varying curvature values, so the lowest value within those values is selected to segment ground points that have low curvature values.

Table 16: Region Growing Curvature vs. RMSE for Ankeny dataset

Region Growing Curvature vs. RMSE				
Curvature	**1**	2	3	4
RMSE (m)	**0.326329**	0.326349	0.326235	0.326241631
Max. Height Difference (m)	**3.35944**	3.36423	3.35944	3.35944

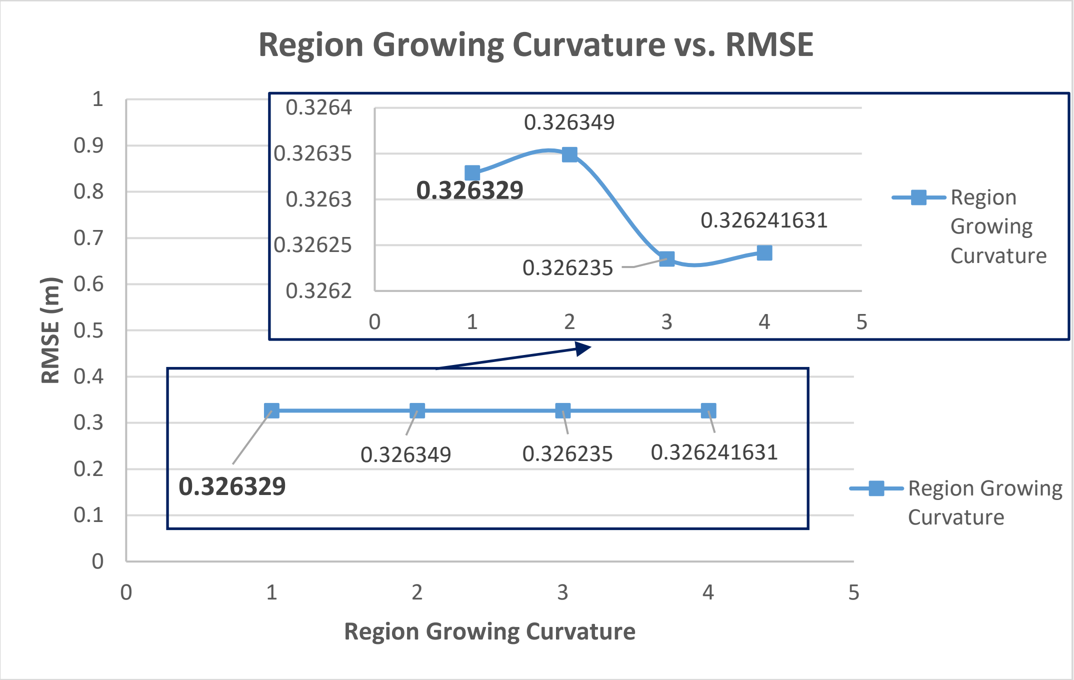


Figure 76: Region Growing Curvature vs. RMSE for Ankeny dataset

Lastly, different neighborhood sizes are tested for region-growing segmentation. As previously mentioned, the higher the kNN value, the more computation time. Here, the fastest solution is to check 8 neighbors in the tested conditions. Nevertheless, in densely populated point distributions, 8 kNN might represent a small neighborhood that may be segmented inaccurately. Although checking 16 kNN is more costly, for the sake of robustness, 16 is chosen.

Table 17: Region Growing kNN vs. RMSE for Ankeny dataset

Region Growing kNN vs. RMSE				
kNN	8	**16**	32	64
RMSE (m)	0.311714	**0.317983**	0.325347	0.325336
Max. Height Difference (m)	3.40657	**3.39337**	3.36423	3.36423

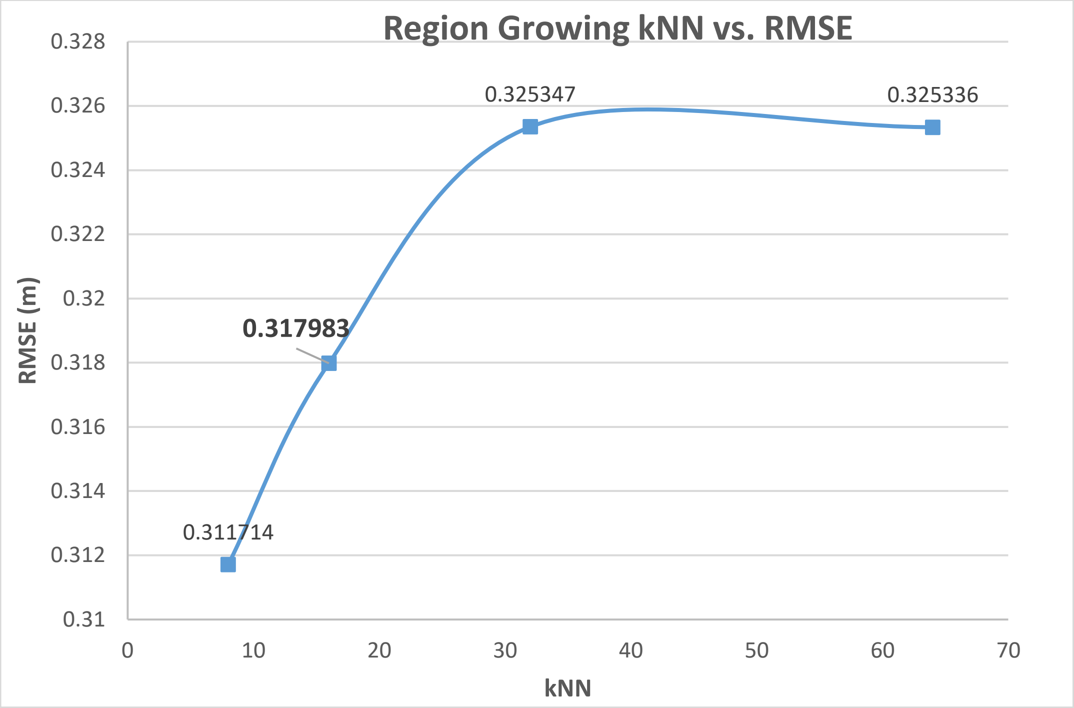


Figure 77: Region Growing kNN vs RMSE for Ankeny dataset

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In this study, the aim is to create a DTM rather than point cloud classification. The input point cloud is thinned in a way that the omitted points have a minor effect on the resulting DTM. Because of that, the accuracy assessment is based on raster data rather than point cloud. (Becker, Rosinskaya, Häni, D'Angelo, & Strecha, 2018) used an algorithm to classify the point clouds used in this thesis; however, this algorithm misclassified some of the ground points. Due to this reason, the labeled data cannot be used directly as the ground truth. Since this thesis focuses on DTM rather than point cloud classification, a DTM is created directly from their point cloud. Afterwards, the DTM model is corrected to use as the ground truth. By doing so, their results could also be evaluated, and their findings are compared with our algorithm. In addition, the results are also compared with an available DTM extraction algorithm. In CloudCompare software, there is a free DTM extraction called CSF (Cloth Simulation Filter) Plugin developed by (Zhang, et al., 2016).

There are two different datasets with different point densities and characteristics. According to (The American Society for Photogrammetry and Remote Sensing, 2004), vertical accuracy is the main criterion to evaluate the quality of a DEM. Root Mean Square Error (RMSE) is calculated to assess DEM accuracies. RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i-0}^{n} (x_i - y_i)^2}{n}} \tag{9}$$

where x_i is the original data, y_i is the reference data and n is the total number of samples.

In Section 4.2, the overall RMSE values for Ankeny and Cadastre are evaluated. In Section 4.3, the classification performance on different objects (buildings, vegetation vehicles, etc.) is compared qualitatively with the ground truth data.

Table 18: Comparison of RMSE values three different algorithms

	Ankeny		Cadastre	
	RMSE (m)	Maximum Height Difference (m)	RMSE (m)	Maximum Height Difference (m)
Proposed Algorithm	0.25	2.58	0.70	4.00
CSF	0.26	2.67	0.72	10.14
Pix4D Research	0.25	4.57	0.44	3.95

4.2 Quantitative Results

The RMSE values for both datasets are calculated. In Table 19, the results can be seen for Ankeny and Cadastre datasets. The height difference results from CSF for each dataset are illustrated in Figure 78 and Figure 84. The height differences between the extracted DTM and ground truth for both datasets are shown in Figure 80 and Figure 86 for Ankeny and Cadastre, respectively. Similarly, from (Becker, Rosinskaya, Häni, D'Angelo, & Strecha, 2018) classification result, the points labeled as ground and road are taken into account to create a DTM. The height difference between the ground truth and the results are presented in Figure 82, and Figure 88, for Ankeny and Cadastre dataset, respectively. For each difference map, the histogram plots are shown in Figure 79, Figure 81, Figure 83, Figure 85, Figure

87, Figure 89 for three different algorithms in two different datasets. CSF algorithm removes some steep regions that may represent the fallacious. All of the algorithms perform well in building removal tasks; however, in the Cadastre dataset, all of the DTMs have some errors where the adjacent buildings occur due to data gaps after the removal that cause a poor interpolation over the regions.

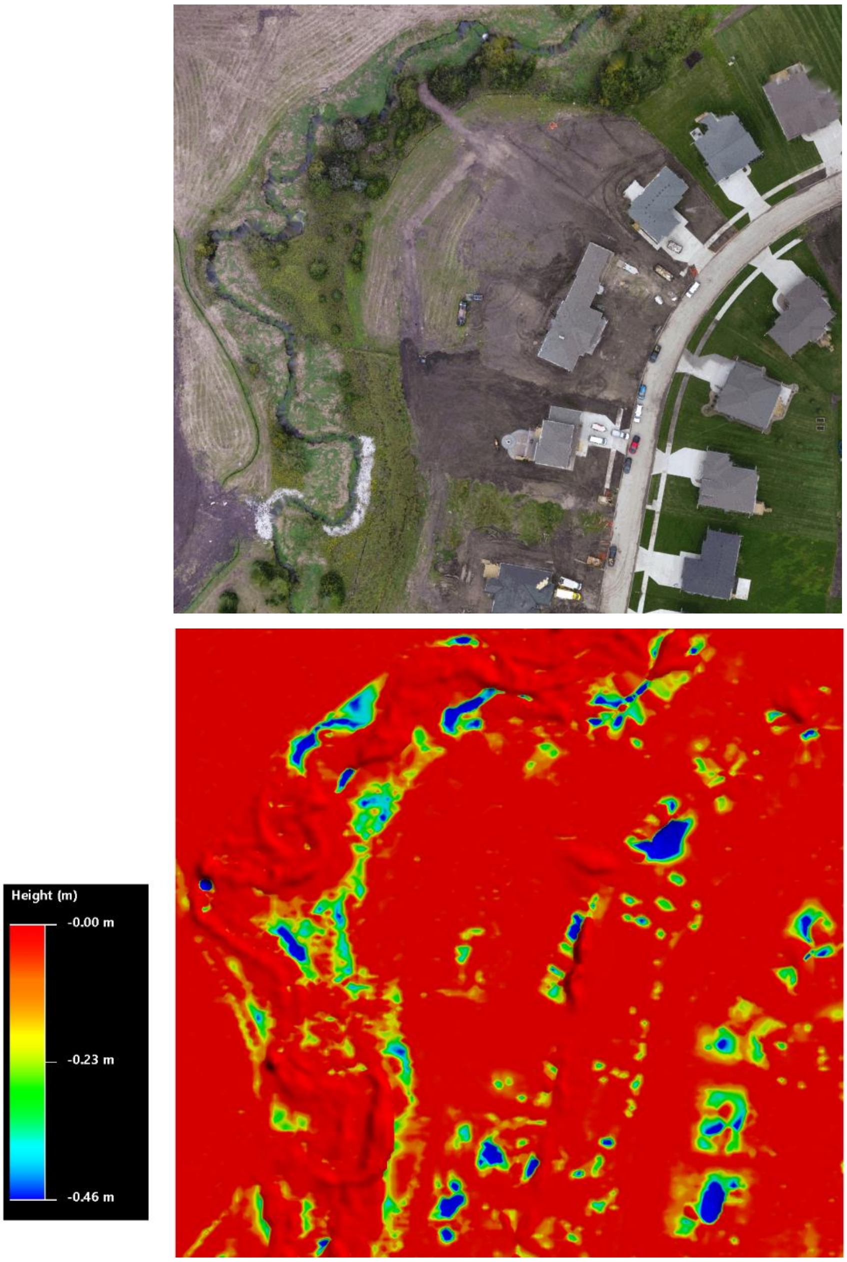


Figure 78: The RGB colored Ankeny dataset (top) and difference map between ground truth and CSF output

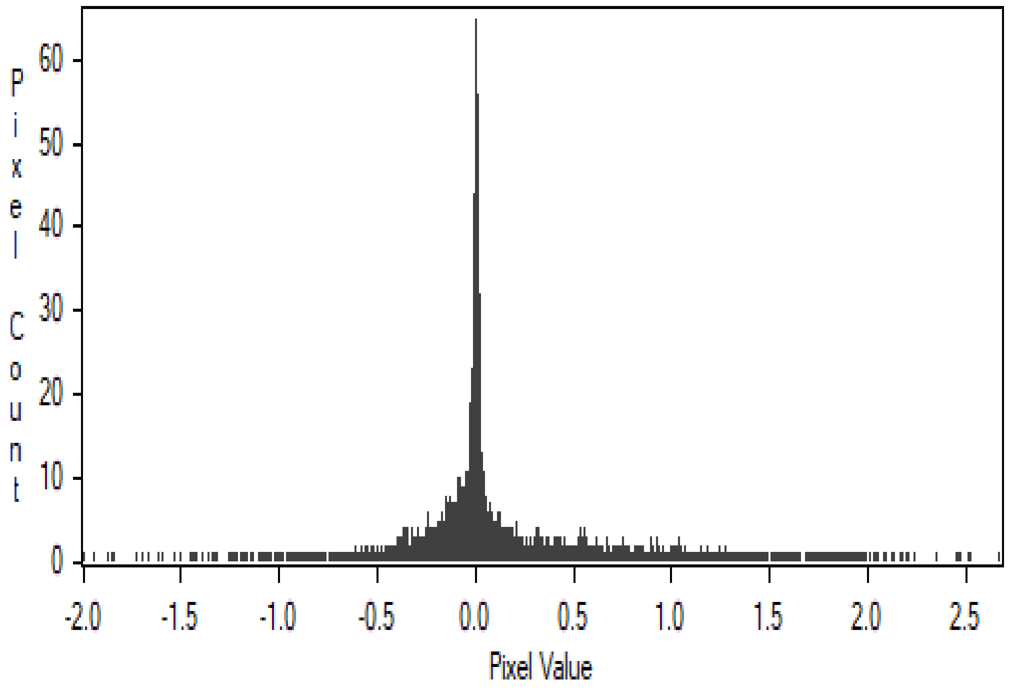


Figure 79: The histogram of the CSF output height difference in Ankeny dataset

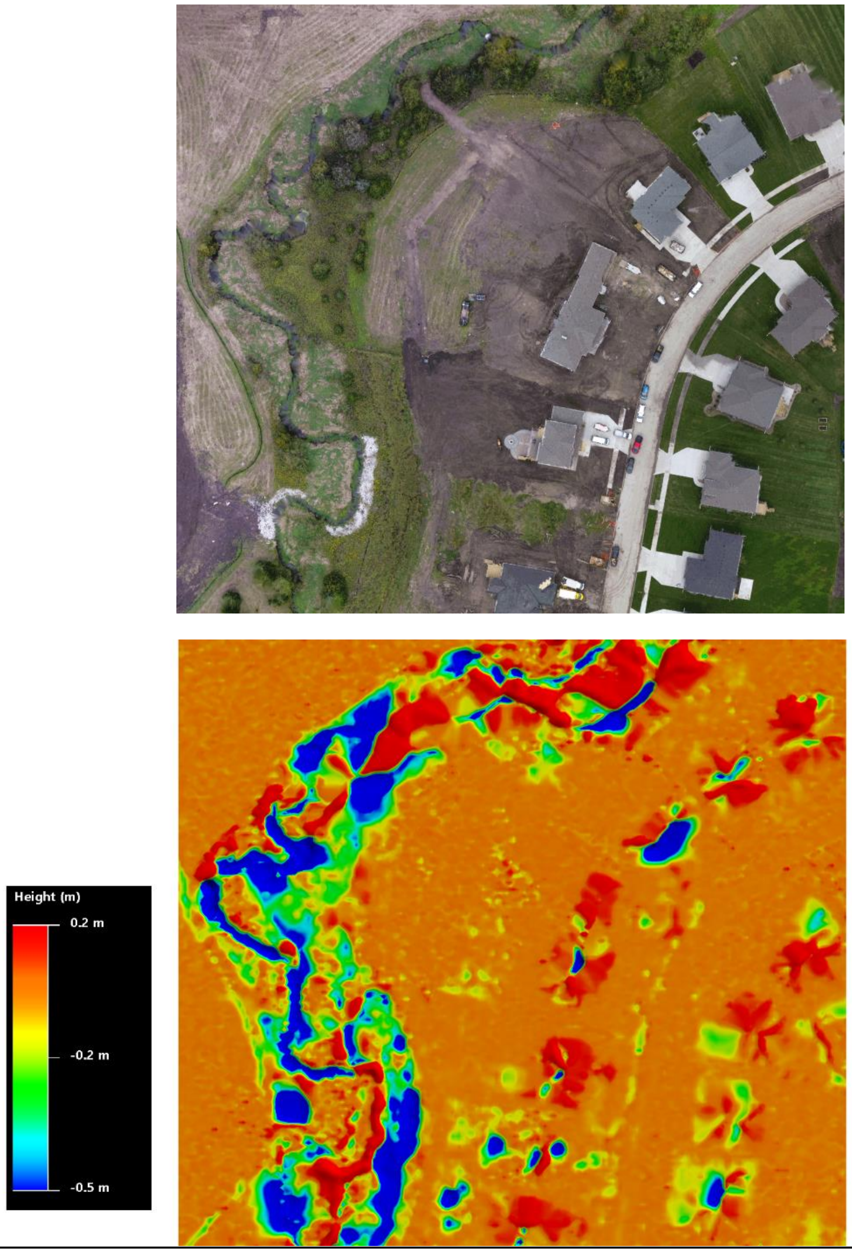


Figure 80: The RGB colored Ankeny dataset (top) and difference map between ground truth and the proposed algorithm output

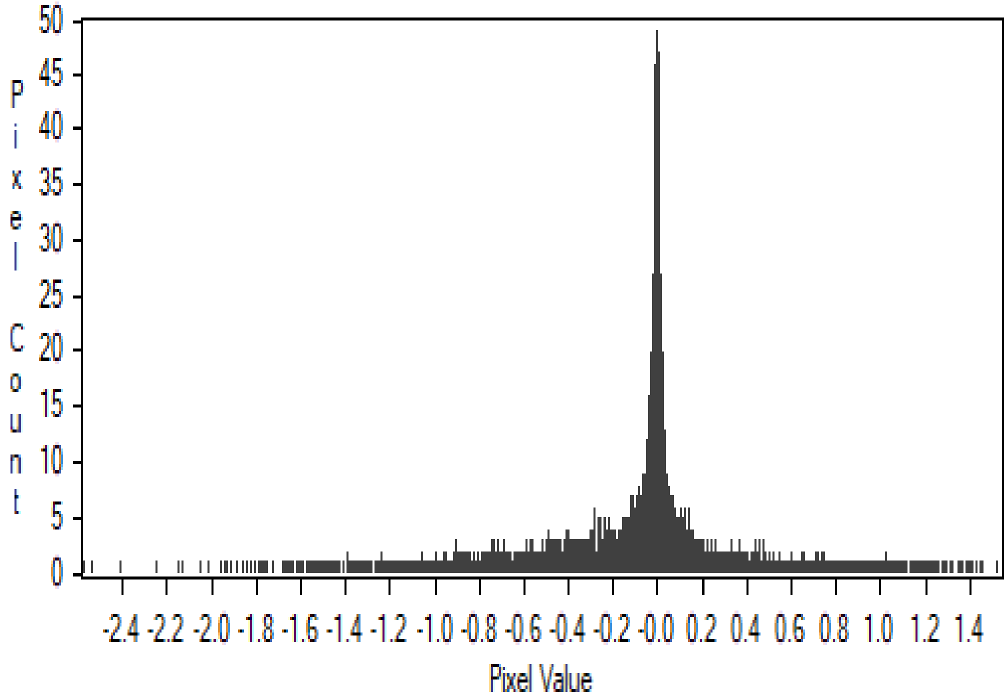


Figure 81: The histogram of the proposed algorithm output height difference in Ankeny dataset

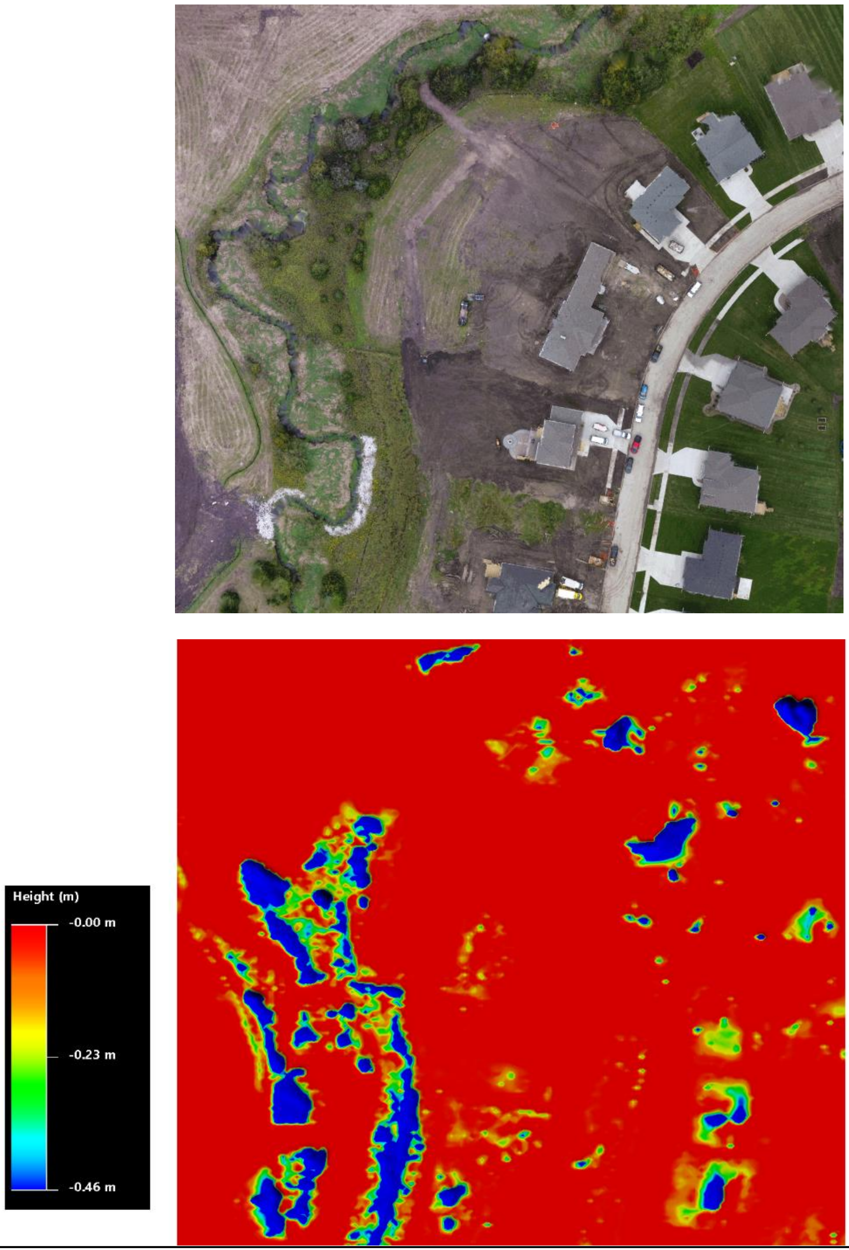


Figure 82: The RGB colored Ankeny dataset (top) and difference map between ground truth and Pix4D research output

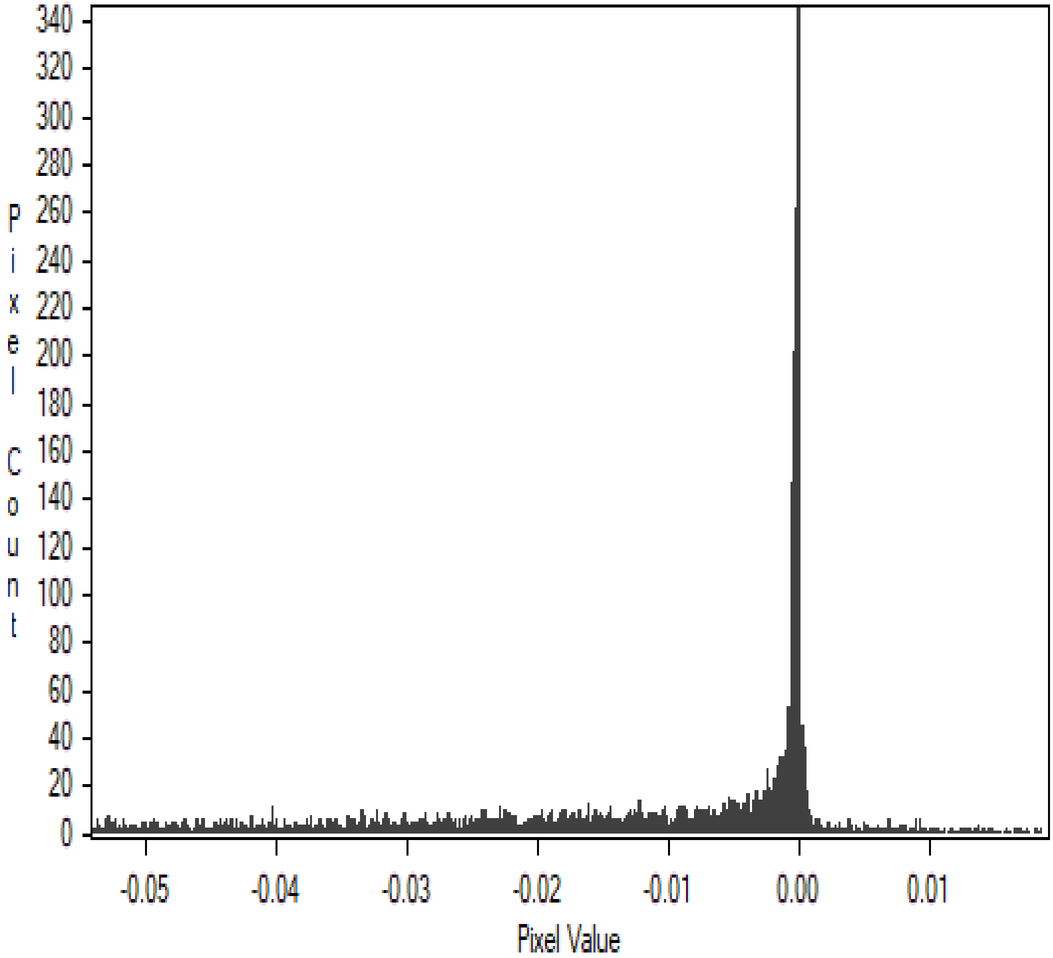


Figure 83: The histogram of the Pix4D research output height difference in Ankeny dataset

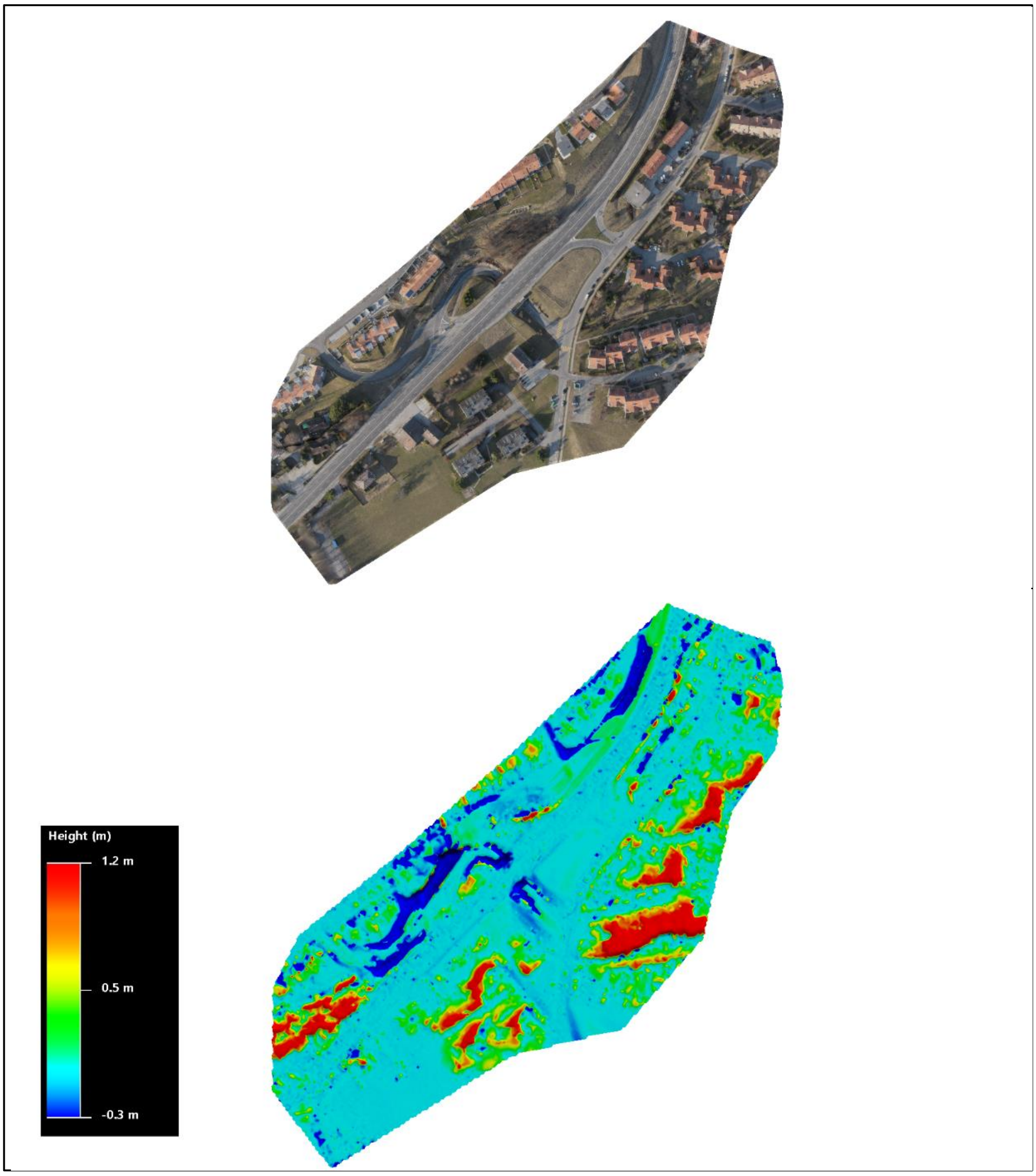


Figure 84: The RGB colored Cadastre dataset (top) and difference map between ground truth and CSF output

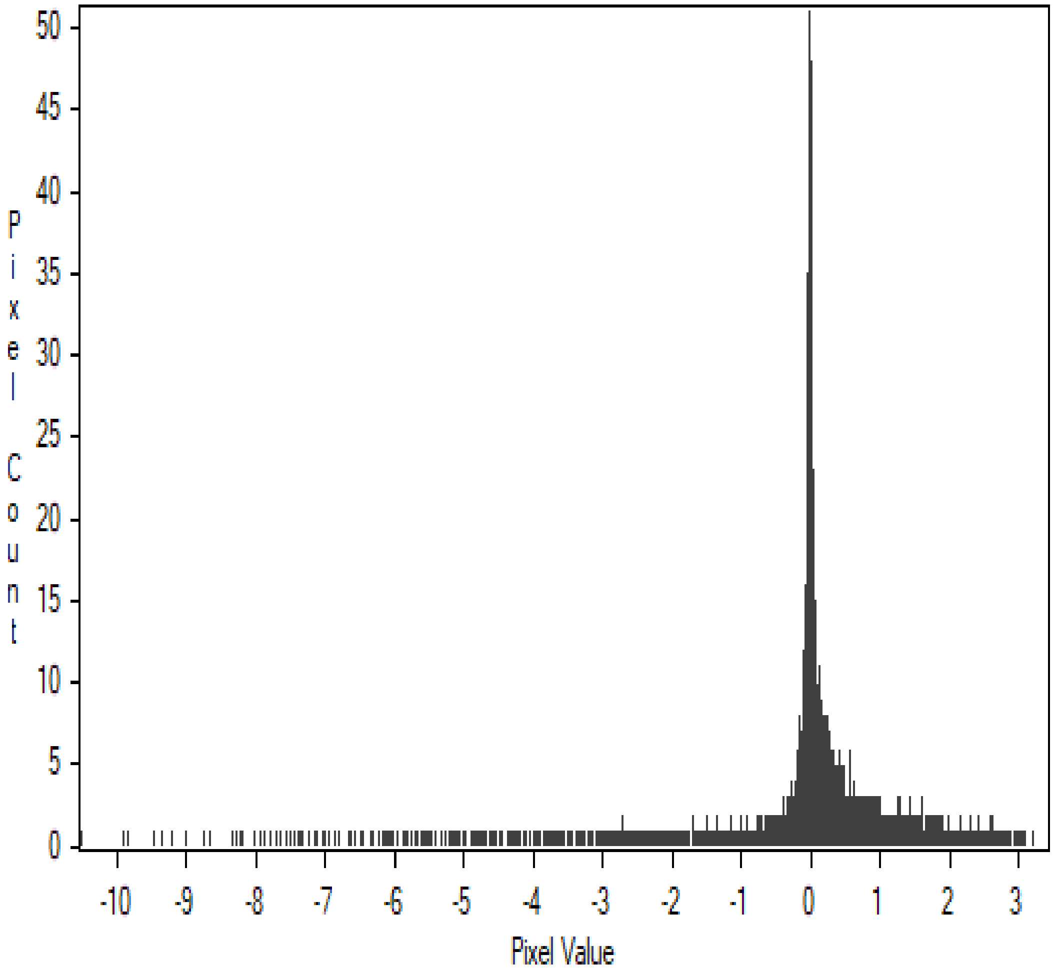


Figure 85: The histogram of the CSF output height difference in Cadastre dataset

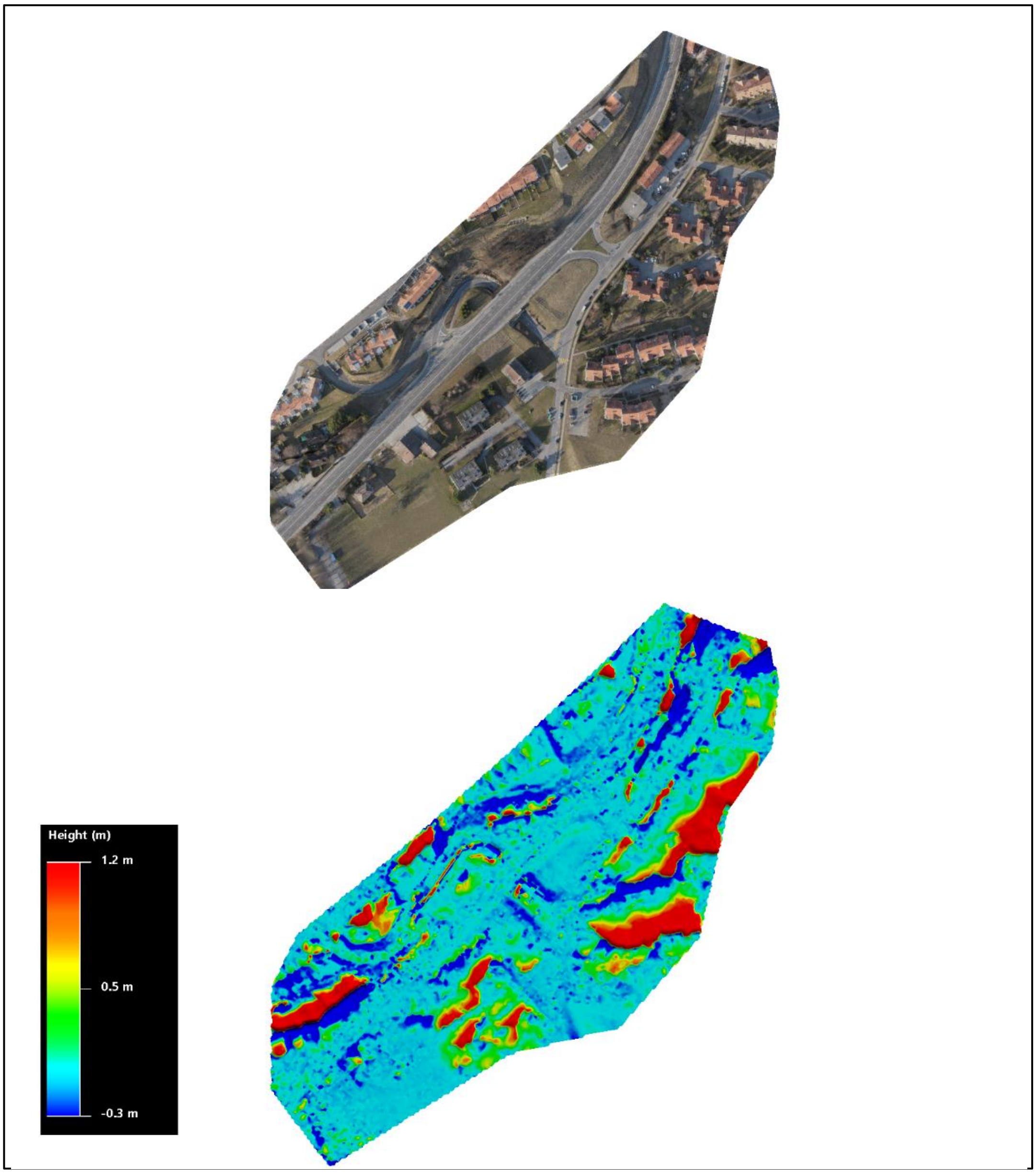


Figure 86: The RGB colored Cadastre dataset (top) and difference map between ground truth and the proposed algorithm output

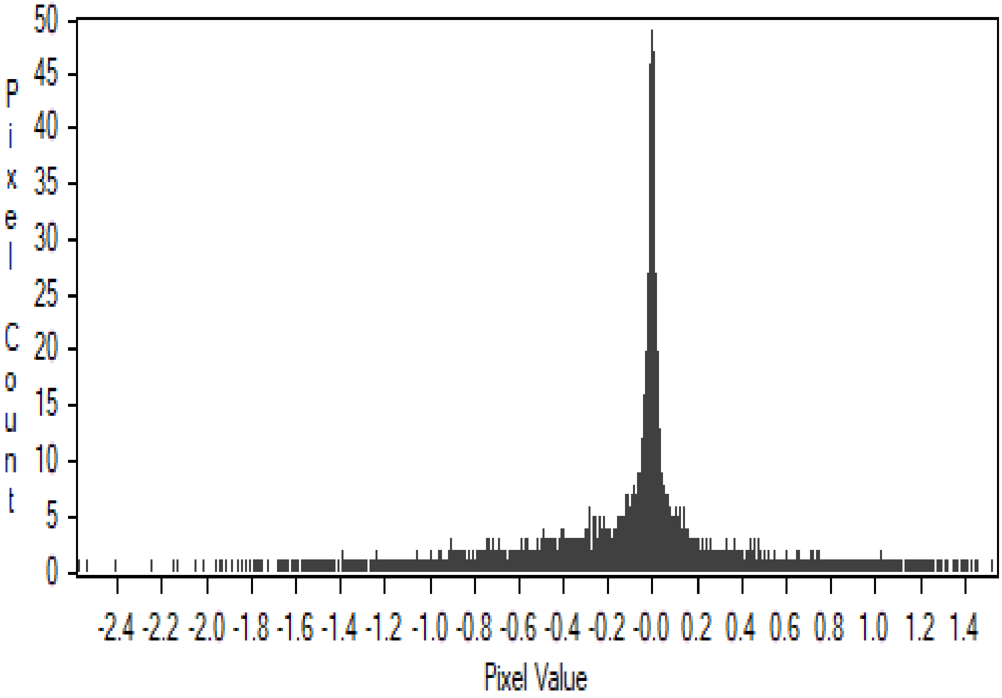


Figure 87: The histogram of the proposed algorithm output height difference in Cadastre dataset

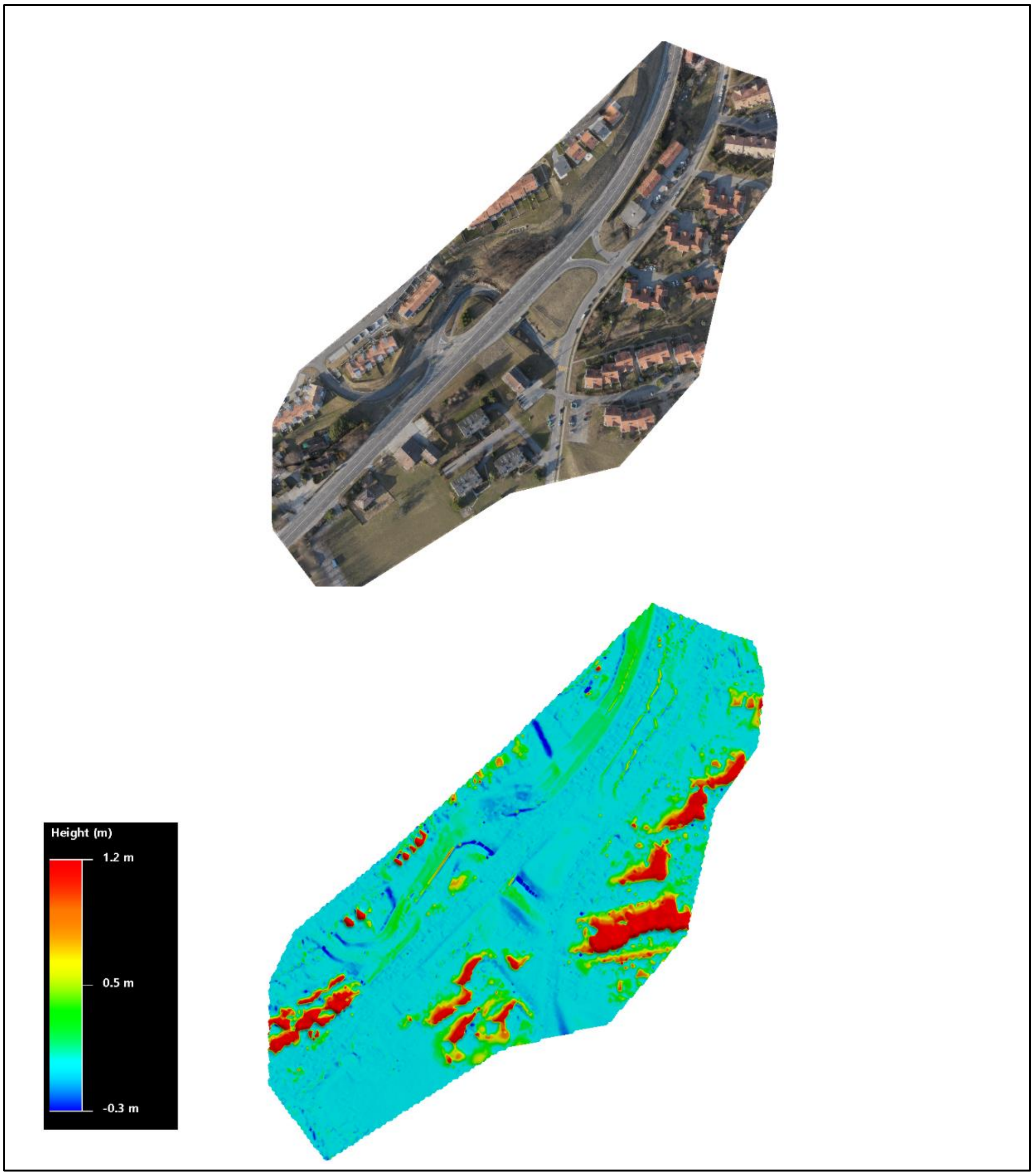


Figure 88: The RGB colored Cadastre dataset (top) and difference map between ground truth and the Pix4D research output

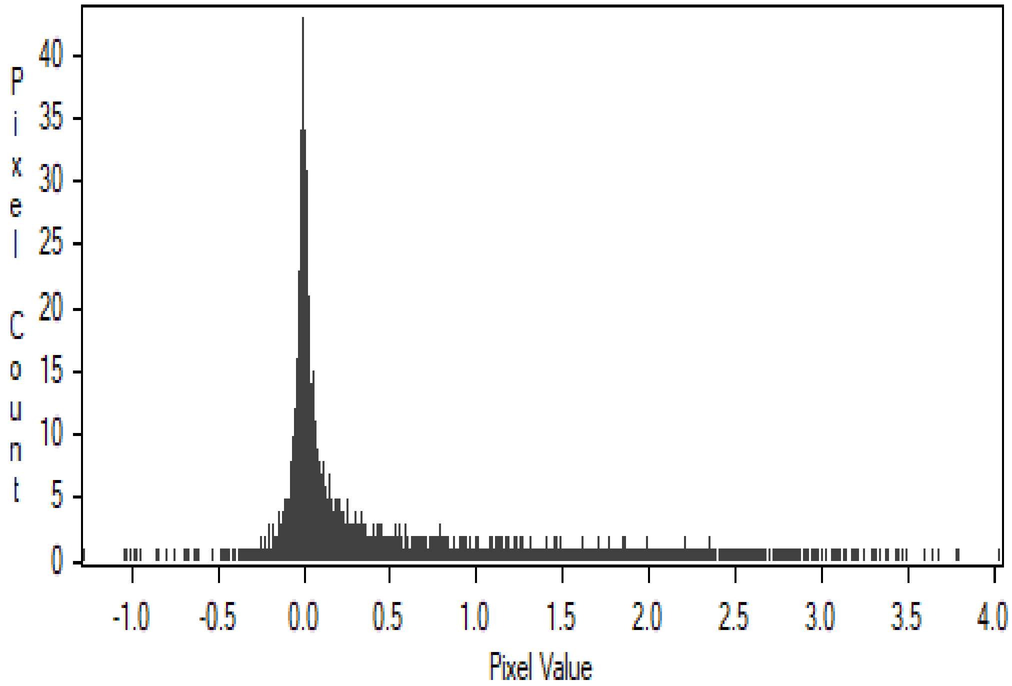


Figure 89: The histogram of the Pix4D research output height difference in Cadastre dataset

Table 19: RMSE Errors of the proposed algorithm

	Raster Spatial Resolution (m)	RMSE Error (m)
Ankeny	1	0.25
Cadastre	2.2	0.70

4.3 Qualitative Results

In this chapter, the performance of the proposed algorithm on object removal is discussed in different examples. The Ankeny dataset is good for evaluating the algorithm performance in a flat region with buildings, vehicles, and posts whereas Cadastre dataset is a relatively dense urban area with complex road structures, steep slopes, and adjacent buildings. The algorithm works as expected for removing buildings, vehicles, powerlines, and posts. However, in some densely vegetated areas and steep slopes, it may suffer. The Ankeny dataset has an example of vegetation on a riverbed, and the example is shown in 4.3.2. Cadastre dataset contains complex road structures and steep slope areas, which is discussed in 4.3.5.

4.3.1 Building Removal Performance

In Figure 90, the voxelized point cloud is shown before the building removal and in Figure 91, after the removal. The algorithm works well for individual building removal processes. These datasets do not contain dense urban areas. Consequently, the performance could not be tested for dense urban areas where the buildings are close to each other. However, since the algorithm utilizes a multi-resolution approach, the contextual information can be preserved, and it is not expected to have a problem removing such structures.

Figure 90: Building example before removal

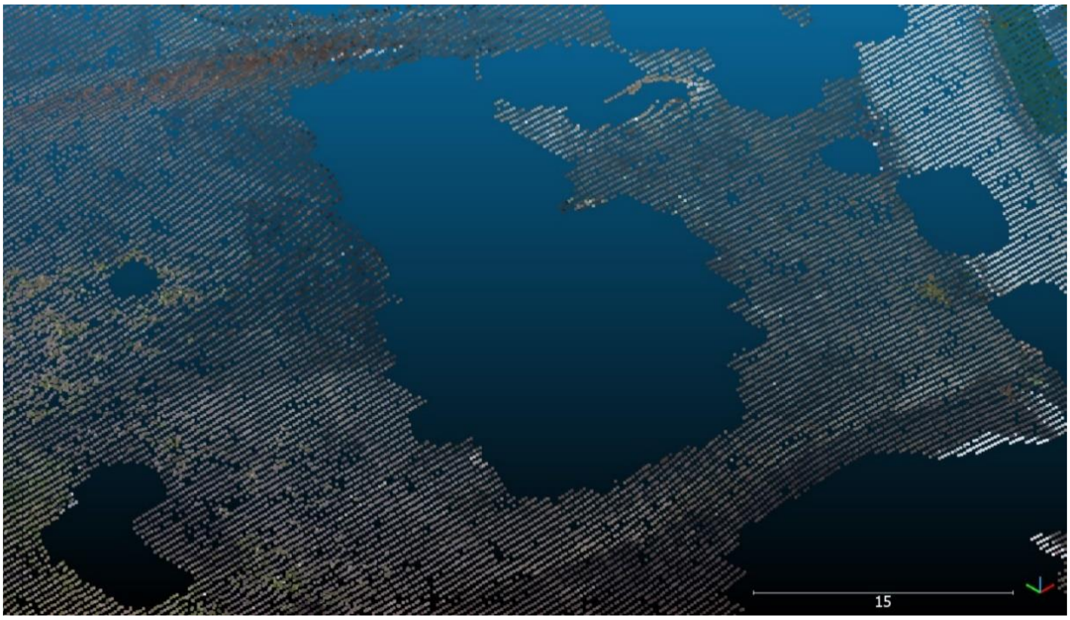


Figure 91: Building example after removal

4.3.2 Vegetation Removal Performance

In Figure 92, the voxelized point cloud is shown before the vegetation removal and in Figure 93, after the removal. The algorithm works well for individual tree removal processes. Densely vegetated areas are challenging for DTM extraction process from photogrammetric point cloud because it is hard to take reference terrain points in most cases. This study is based on exploiting the planarity values, and trees show the opposite characteristic. The multi-resolution planarity approach works well for vegetation removal in most cases. However, there are some minor terrain deviations and bushes next to the river bed in the following example. Some terrain points are filtered out with the bushes and grassland. In Figure 93, it can be seen that while removing the low vegetation and bushes, the algorithm removed some points that belong to the riverbed. The planarity-based filtering is conducted by checking 128 nearest neighbors, which is suitable for filtering objects, and some points beyond that exceed the object boundary. If these objects are next to missing data points (inside the riverbed), they may cause data gaps.

Figure 92: Vegetation before removal

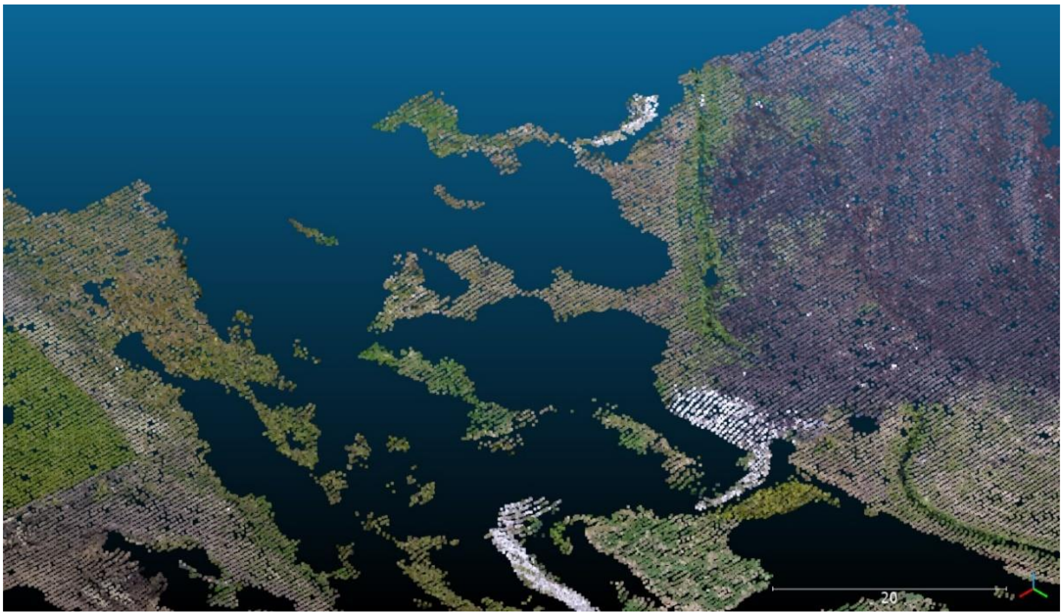


Figure 93: Vegetation after removal

4.3.3 Vehicle Removal Performance

In Figure 94, the voxelized point cloud is shown before the vehicle removal and in Figure 95, after the removal. The algorithm handles vehicle removal processes effectively.

Figure 94: Vehicles and other objects before removal

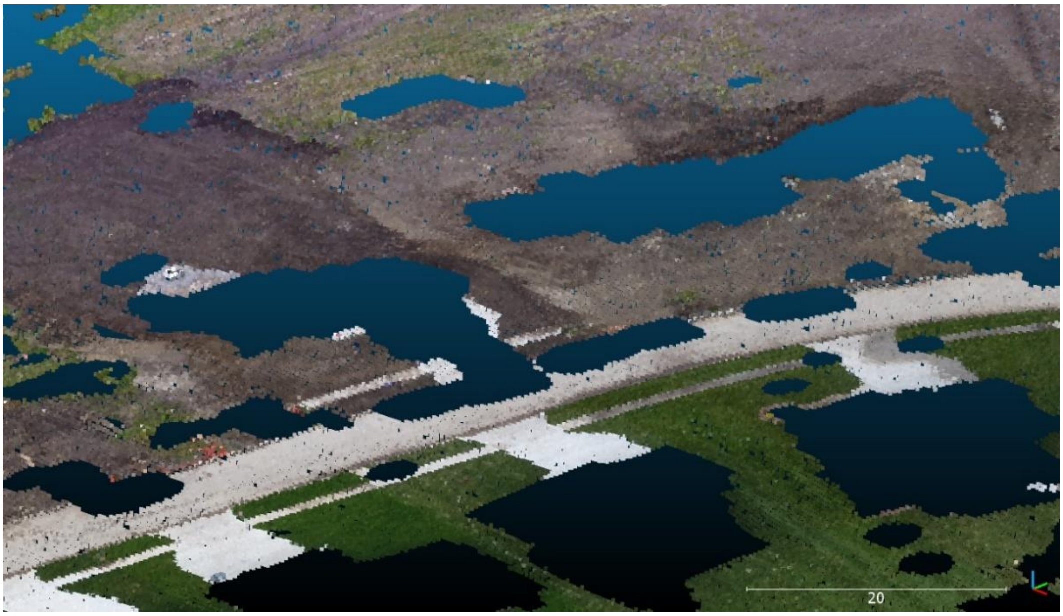


Figure 95: Vehicles and other objects after removal

4.3.4 Missing Facades Performance

Due to the flight path or missing photos, there might be some missing points where some objects are not sensed properly. For example, in Figure 96, it can be observed that there are missing points in the facades. In Figure 97, the resulting ground points from these examples are illustrated. The algorithm can handle the missing data problem by using a combination of grid-based and neighborhood-based planarity approaches.

Figure 96: Missing points on facades

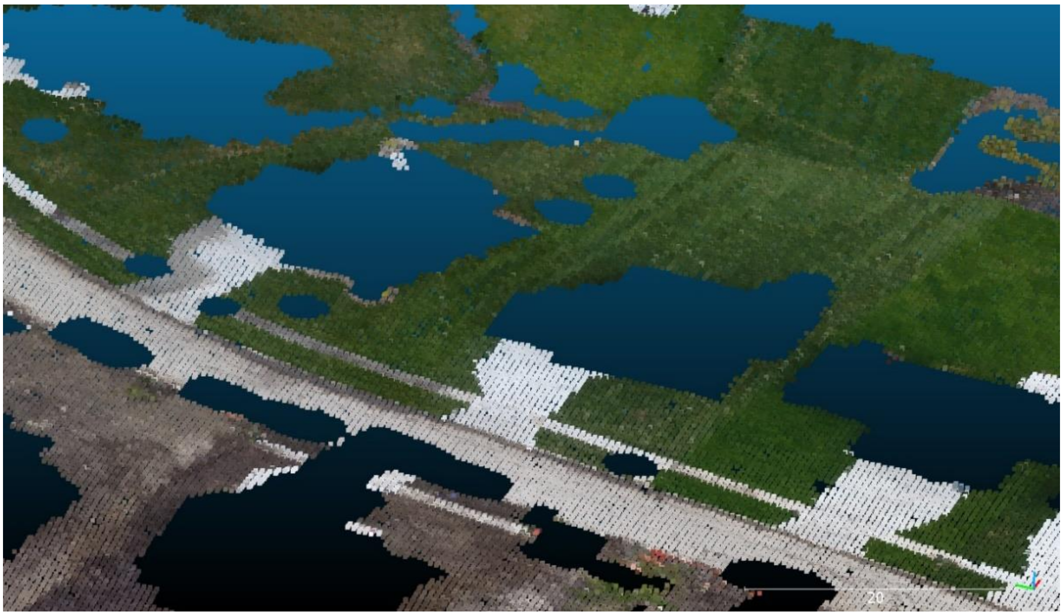


Figure 97: Resulting terrain on missing data example

4.3.5 Steep Slopes

The algorithm has some defects in some challenging cases. In Cadastre dataset, there is an example with some vegetation and buildings on a steep area shown in the middle of Figure 98 and in Figure 100. The algorithm removes some terrain points while removing the trees and the buildings, which results in a data gap in this example. However, there are other steep slope examples in the same dataset without objects on them. In these examples, the algorithm manages to remove the objects by preserving the terrain characteristics. In Figure 99 and Figure 101, the steep slope example without objects is shown as before and after the algorithm is applied, respectively.

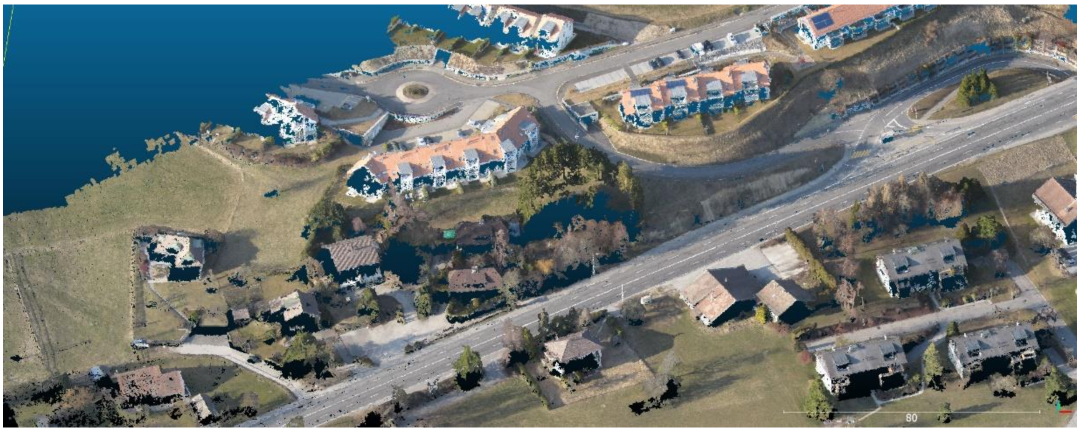


Figure 98: Steep slope with vegetation and buildings in Cadastre dataset

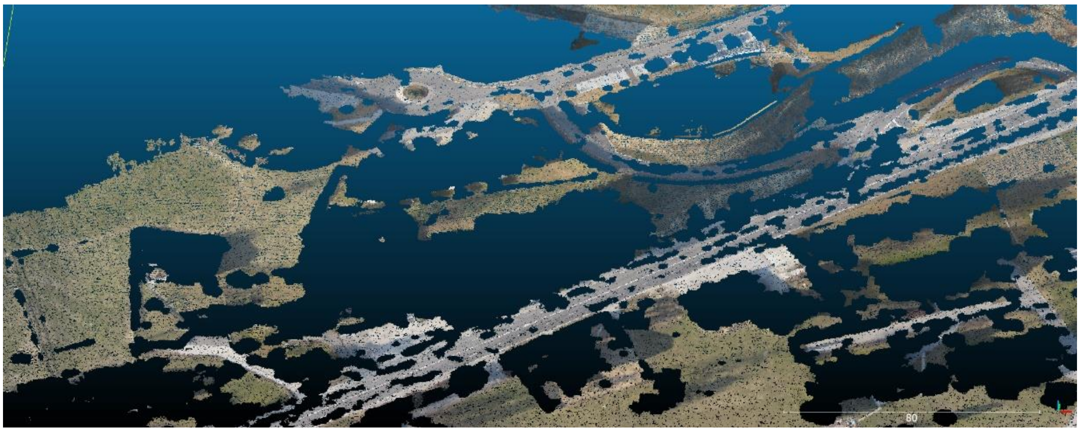


Figure 99: Steep slope with vegetation and buildings in Cadastre dataset after object removal

Figure 100: Steep slope example with complex road structure

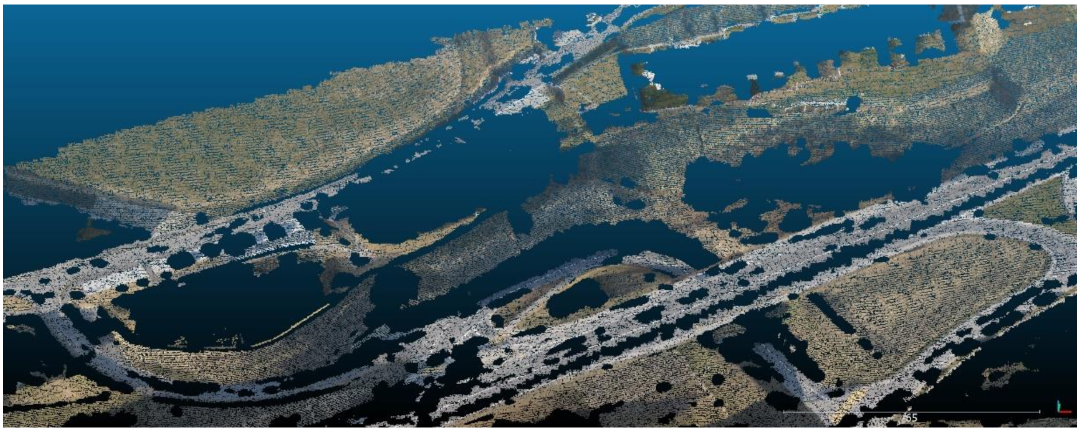


Figure 101: Steep slope example without objects after object removal

4.3.6 Qualitative Comparison

Although three algorithms show similar RMSE values, they have advantages and disadvantages depending on the terrain characteristics. There are some vegetation leftovers in Pix4D Research DTM. CSF and the proposed algorithm perform better and show similar results. It can be seen in Figure 102 that the vegetated river case is challenging for all the algorithms. In the second case, the performances of the algorithms are compared in a road example. The proposed algorithm performs better than CSF and Pix4D Research algorithms in the given case. As illustrated in Figure 103, the proposed algorithm flattens the bumps on the road where CSF and Pix4D Research algorithms keep these bumps.

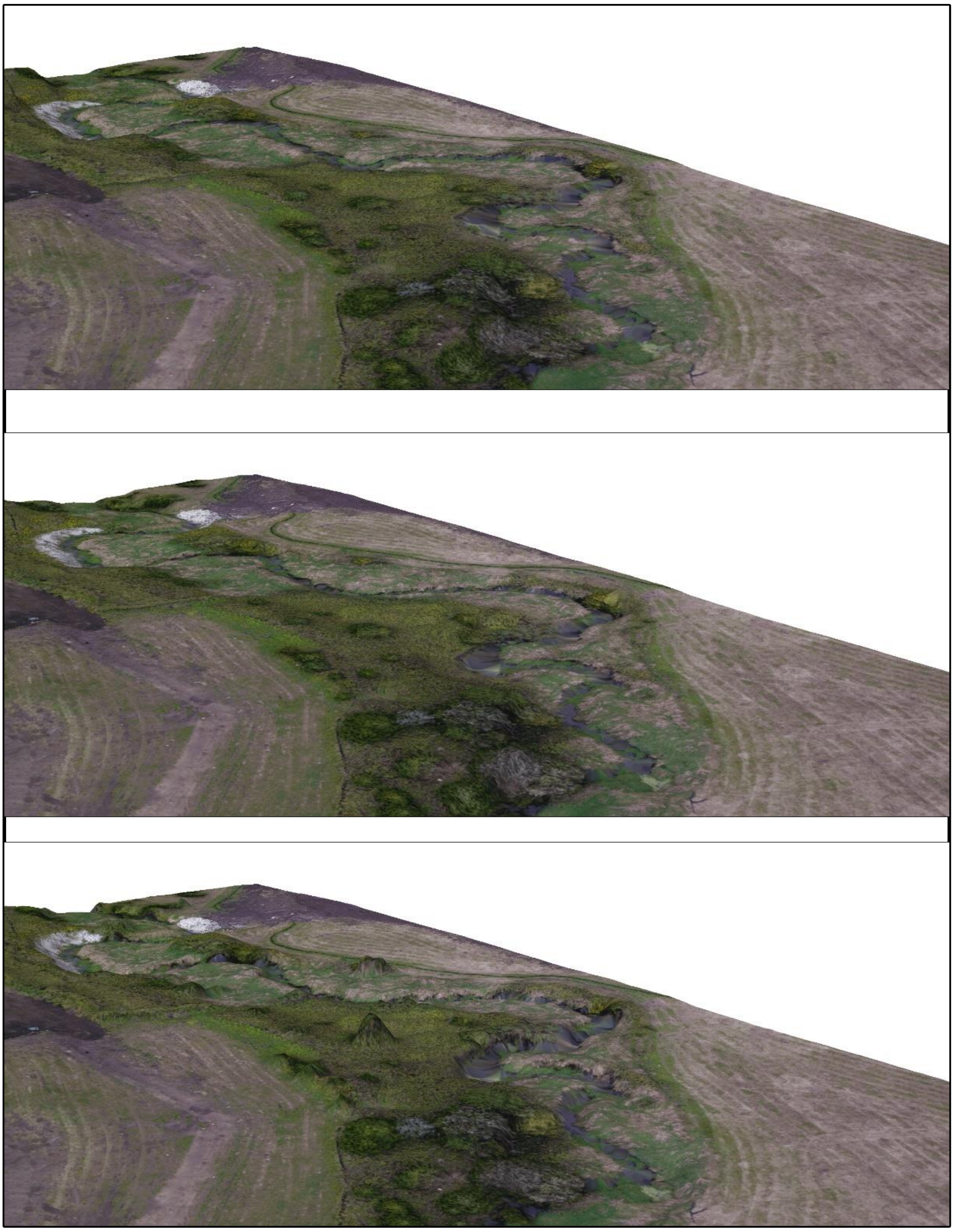


Figure 102: The proposed algorithm (top), CSF (middle), Pix4D Research results (bottom) in river example

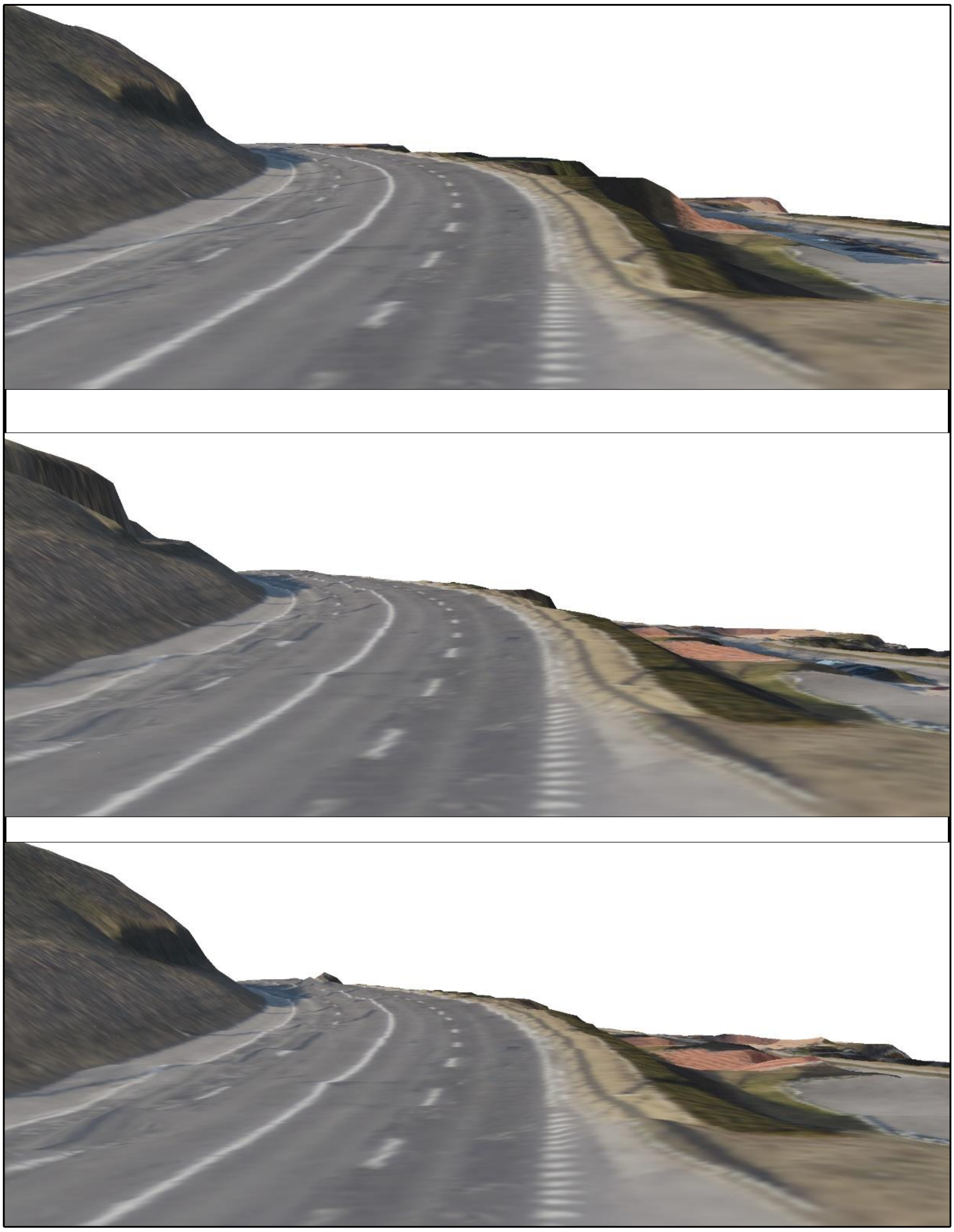


Figure 103: The proposed algorithm (top), CSF (middle), Pix4D Research results (bottom) in road example

4.4 Test Areas and Failure Cases

The proposed algorithm results are compared with CSF and Pix4D research results in different test areas to check algorithms accuracy on different features. There are six different test areas, namely; road samples, buildings samples from flat region, buildings samples from steep region, complex road samples and hilly vegetation samples, and vegetation samples from riverbed area. The samples from Ankeny and Cadastre datasets are shown in Figure 104, and Figure 105, respectively. For each sample area, the maximum difference from the ground truth, and the RMSE values are calculated. The algorithm performance results on different samples in Ankeny dataset are presented in Table 20, Table 21, and Table 22. Similarly, for Cadastre datasets, the results are shown in Table 23, Table 24, and Table 25. Based on these results, the algorithm performance is compared with the existing algorithms.

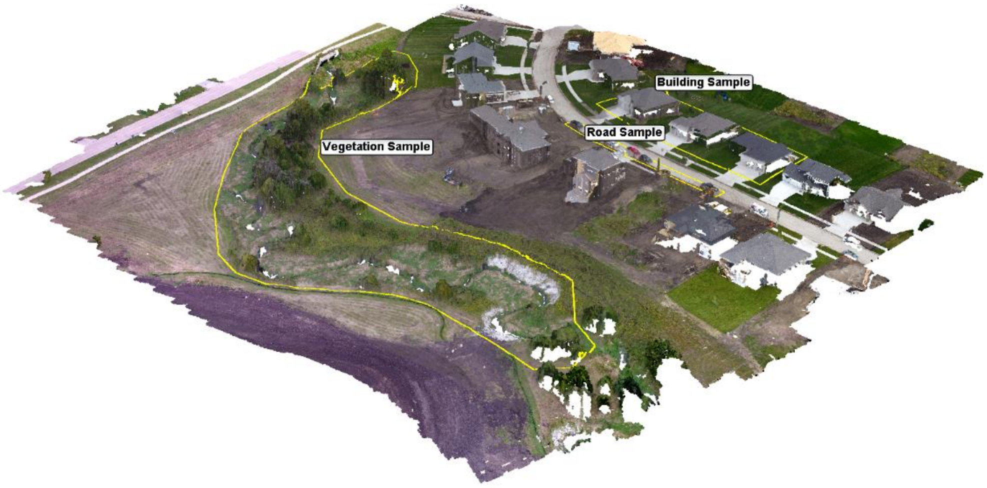


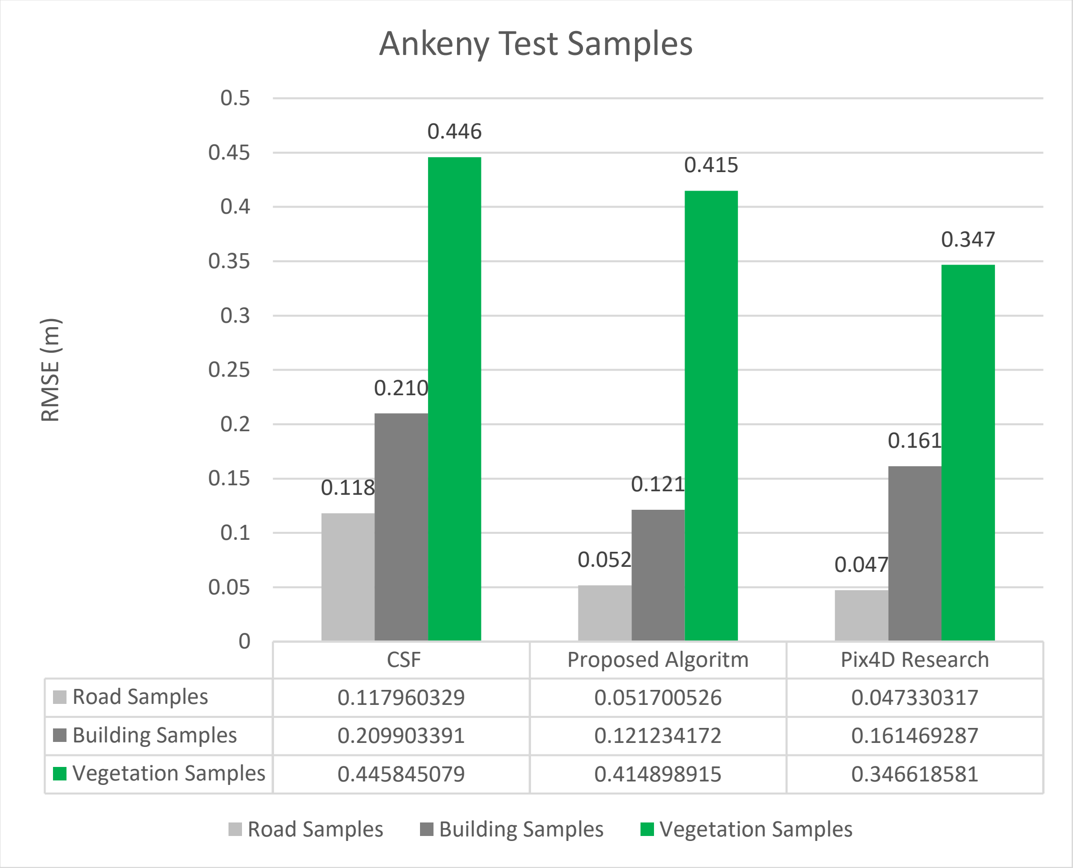
Figure 104: Ankeny test samples

Figure 105: Cadastre test samples

Table 20: Ankeny test areas

CSF			Proposed Algorithm		Pix4D Research	
	Max. Height Difference (m)	**RMSE (m)**	**Max. Height Difference (m)**	**RMSE (m)**	**Max. Height Difference (m)**	**RMSE (m)**
Road	0.68485	0.11796	0.27707	0.05170	0.33908	0.04733
Building	1.37337	0.20990	0.92186	0.12123	1.11423	0.16147
Vegetation	2.63643	0.44585	2.57681	0.41490	3.70518	0.34662

Table 21: Ankeny RMSE values for different samples



	CSF	Proposed Algoritm	Pix4D Research
Road Samples	0.117960329	0.051700526	0.047330317
Building Samples	0.209903391	0.121234172	0.161469287
Vegetation Samples	0.445845079	0.414898915	0.346618581

Results indicate that in Ankeny dataset (flat) three algorithms perform similarly. Their performance in different areas is summarized in Table 22. Although Pix4D shows the best performance in vegetation and road samples, the proposed algorithm shows better performance in building samples.

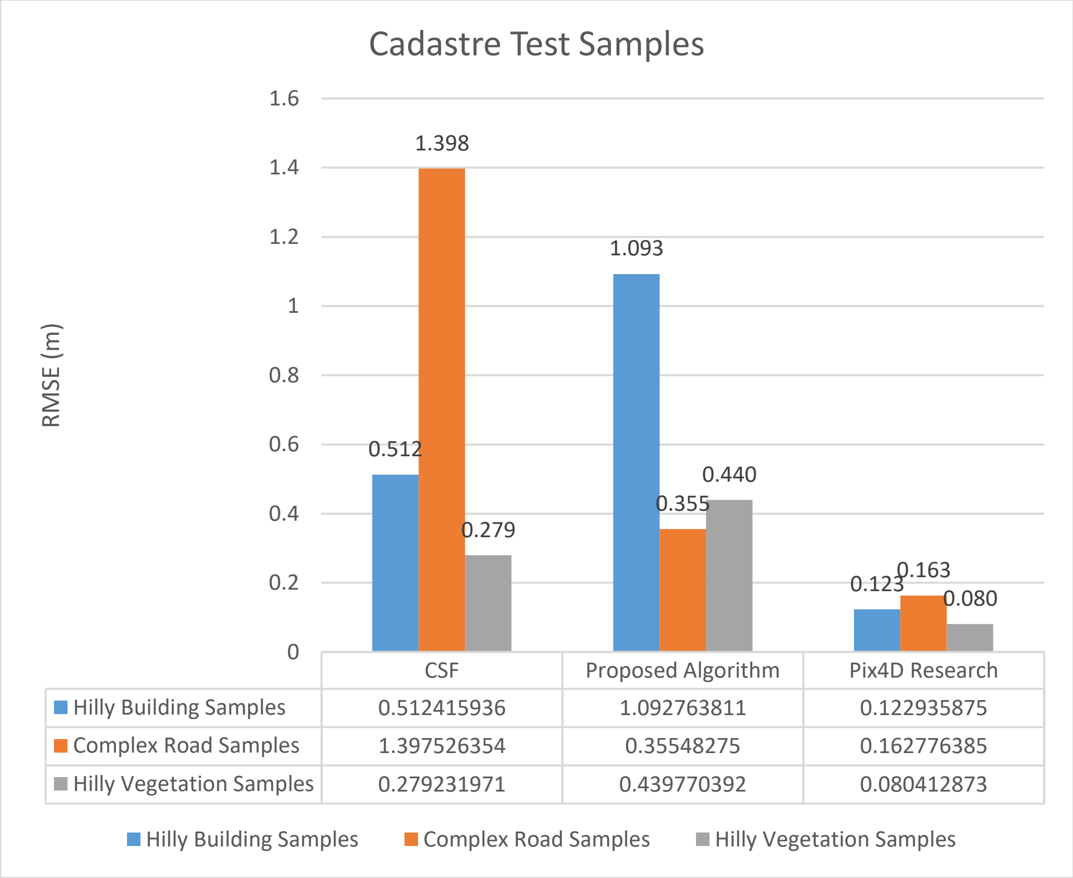
Table 22: Comparison of the algorithms on Ankeny dataset samples

	CSF	Proposed Algorithm	Pix4D
Road Samples	+	++	+++
Building Samples	+	+++	++
Vegetation Samples	+	++	+++
+ sign means better performance			

Table 23: Cadastre test samples

CSF			Proposed Algorithm		Pix4D Research	
	Max. Height Difference (m)	RMSE (m)	Max. Height Difference (m)	RMSE (m)	Max. Height Difference (m)	RMSE (m)
Hilly Building	2.99341	0.512416	3.74316	1.0927638	0.442358	0.122936
Complex Road	8.68323	1.397526	4.34857	0.3554827	1.10291	0.162776
Hilly Vegetation	2.99799	0.279232	4.20294	0.4397704	0.649841	0.080413

Table 24: Cadastre RMSE values for different samples



	CSF	Proposed Algorithm	Pix4D Research
🟦 Hilly Building Samples	0.512415936	1.092763811	0.122935875
🟧 Complex Road Samples	1.397526354	0.35548275	0.162776385
⬜ Hilly Vegetation Samples	0.279231971	0.439770392	0.080412873

Results show that in Cadastre dataset (flat) three algorithms performs differently. Their performance in different areas is summarized in Table 25. Pix4D research algorithm performs the best in each sample. In complex road sample, CSF algorithm performed poorly having around 1.4 m RMSE. Similarly, the proposed algorithm suffers in hilly building sample area having around 1.1 m RMSE.

Table 25: Comparison of the algorithms in Cadastre dataset

	CSF	Proposed Algorithm	Pix4D Research
Hilly Building Samples	++	+	+++
Complex Road Samples	+	++	+++
Hilly Vegetation Samples	++	+	+++
+ sign means better performance			

4.5 Computation Time

The computation time for both datasets are given in Table 26. The computation times are measured for different parts of the algorithms: reading the points, grid planarity calculation, approximate coplanarity calculation, planarity-based filtering, region growing segmentation, rough ground calculation, and final ground point extraction. Processing the Ankeny dataset takes around 199 seconds, and about 282 seconds on Cadastre dataset on Intel i7700HQ CPU with 16 GB memory and GTX 1060 Max-Q GPU computer. With the same computer specifications, CSF takes 28 seconds Cadastre and 5 seconds in Ankeny dataset. (Becker, Rosinskaya, Häni, D'Angelo, & Strecha, 2018) states that their classification algorithm takes 3 minutes for 17.5 million points.

Table 26: Computation time for the proposed algorithm

Computation Time (seconds)						
	Reading and Gridding	Coplanarity Filtering (kNN + Grid Based)	Grid Planarity Calculation	Region Growing Segmentation	Rough Ground Calculation and Roof Filtering	Total
Ankeny	94.4	79.7	8.4	2.5	13.8	198.8
Cadastre	57.1	158.6	12. 4	7.8	45.7	281.6

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS FOR FUTURE WORK

5.1 Conclusions

The purpose of this chapter is to abridge the study and give recommendations for future work. In Section 5.1, all the findings from the study are summarized and concluded. The applications that will benefit from the study are also evaluated in this section. In Section 5.2, the recommendation and possible enhancements for future works are discussed. Conclusions

This study aims to extract DTM from aerial photogrammetric point clouds using a robust multi-resolution planarity-based divide-and-conquer algorithm. In the proposed algorithm, the DTM extraction process is divided into more straightforward steps to solve this complex problem in multiple stages. A combination of planarity values is used in these steps for the above-ground object removal. The algorithm is tested for two different terrain types: flat and relatively steep, in which sudden elevation changes and complex road structures occur. From this study, the following conclusions can be inferred:

Only using a multi-resolution planarity-based object removal approach, different types of objects such as buildings, vehicles, powerlines, trees, bushes can be removed efficiently to estimate the DTM from given input point clouds. However, since some of the steep regions that belong to terrain filtered out in the planarity-based approach, there might be substantial data gaps around these regions. As a result, the DTM might have a high error on these regions. In a flat dataset, the RMSE error is calculated as 0.28 m in 1 m spatial resolution. It is calculated as 0.7 m in 2.2 m spatial resolution for relatively steep terrain with complex structures.

This proposed algorithm can be used as an alternative to existing DTM extraction methods. Although this study aims to create DTM from photogrammetric point cloud, it is applicable to LIDAR datasets as well.

5.2 Future Work

Based on the results from this study, if planarity values are utilized in different resolutions, they can preserve the contextual ground information. Hole filling algorithms can increase the performance in DTM extraction studies based on point clouds. However, these data gaps should be identified first to fill the missing point clouds. It is logical to assume that, in dense urban areas, color features can be used to detect vegetation together with the planarity values to increase accuracy. To increase the algorithm accuracy in vegetated areas and steep areas, another covariance features such as sphericity can be combined with planarity values. Also, the interpolation method can be improved have better results in sparse and steep areas. In addition to planarity values, other covariance features can be considered to adapt this multi-resolution divide-and-conquer approach. Computation time can be enhanced by parallel processing and better software implementation. It is planned to implement the algorithm into C++ by using Point Cloud Library and compare the results with Python implementation.

REFERENCES

Achilleos, G. A. (2011). The Inverse Distance Weighted interpolation method and error propagation mechanism – creating a DEM from an analogue topographical map. *Journal of Spatial Science, 56*, 283-304. doi:10.1080/14498596.2011.623348

Arun, P. V. (2013, 12). A comparative analysis of different DEM interpolation methods. *The Egyptian Journal of Remote Sensing and Space Science, 16*, 133–139. doi:10.1016/J.EJRS.2013.09.001

Axelsson, P. (2000). DEM generation from laser scanner data using adaptive TIN models. *International archives of photogrammetry and remote sensing, 33*, 110–117.

Bailey, T. C., & Gatrell, A. C. (1995). *Interactive spatial data analysis* (Vol. 413). Longman Scientific & Technical Essex.

Bartels, M., Wei, H., & Mason, D. C. (2006). DTM generation from LIDAR data using skewness balancing. *Proceedings - International Conference on Pattern Recognition, 1*, 566–569. doi:10.1109/ICPR.2006.463

Becker, C., Rosinskaya, E., Häni, N., D'Angelo, E., & Strecha, C. (2018, 5). Classification of aerial photogrammetric 3D point clouds. *Photogrammetric Engineering and Remote Sensing, 84*, 287–295. doi:10.14358/PERS.84.5.287

Besl, P. J., & Jain, R. C. (1988). Segmentation Through Variable-Order Surface Fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 10*, 167–192. doi:10.1109/34.3881

Brovelli, M. A., Cannata, M., & Longoni, U. M. (2002). Managing and processing LIDAR data within GRASS. *Proceedings of the GRASS users conference, 29.*

Chandrasekar, L., & Durga, G. (2014). Implementation of Hough Transform for image processing applications. *2014 International Conference on Communication and Signal Processing*, (pp. 843-847). doi:10.1109/ICCSP.2014.6949962

Chung, S. Y., Venkatramanan, S., Elzain, H. E., Selvam, S., & Prasanna, M. V. (2019, 1). Supplement of Missing Data in Groundwater-Level Variations of Peak Type Using Geostatistical Methods. *GIS and Geostatistical Techniques for Groundwater Science*, 33–41. doi:10.1016/B978-0-12-815413-7.00004-3

Çınar, A., & Koçan, Y. (2019). DIGITAL TERRAIN MODEL GENERATION FROM HIGH RESOLUTION DIGITAL SURFACE MODEL BY USING CONDITIONAL ADVERSARIAL NETWORKS. *INTERNATIONAL SYMPOSIUM ON APPLIED GEOINFORMATICS.* Istanbul.

Danese, M., Nolè, G., & Murgante, B. (2009). Visual Impact Assessment in Urban Planning. *Studies in Computational Intelligence, 176*, 133–146. doi:10.1007/978-3-540-89930-3_8

Demantké, J., Mallet, C., David, N., Vallet, B., Demantké, J., Mallet, C., . . . Vallet, B. (2011, 9). Dimensionality Based Scale Selection in 3d LIDAR Point Clouds. *ISPAr, 3812*, 97–102. doi:10.5194/ISPRSARCHIVES-XXXVIII-5-W12-97-2011

Demantké, J., Vallet, B., & Paparoditis, N. (2012, 7). STREAMED VERTICAL RECTANGLE DETECTION IN TERRESTRIAL LASER SCANS FOR FACADE DATABASE PRODUCTION. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 1*, 99–104. doi:10.5194/ISPRSANNALS-I-3-99-2012

Deutsch, C. V. (2003, 1). Geostatistics. *Encyclopedia of Physical Science and Technology,* 700. doi:10.1016/B0-12-227410-5/00869-3

Domingo-Santos, J. M., de Villarán, R. F., Rapp-Arrarás, Í., & de Provens, E. C. (2011, 5). The visual exposure in forest and rural landscapes: An algorithm and a GIS tool. *Landscape and Urban Planning, 101,* 52–58. doi:10.1016/J.LANDURBPLAN.2010.11.018

Duda, R. O., & Hart, P. E. (1972, 1). Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM, 15*, 11–15. doi:10.1145/361237.361242

Elmqvist, M., Jungert, E., Lantz, F., Persson, A., & Söderman, U. (2001). TERRAIN MODELLING AND ANALYSIS USING LASER SCANNER DATA. *International Archives of Photogrammetry and Remote Sensing, XXXIV-3/W4,* 219-226.

Fernandes, L. A., & Oliveira, M. M. (2008, 1). Real-time line detection through an improved Hough transform voting scheme. *Pattern Recognition, 41*, 299–314. doi:10.1016/J.PATCOG.2007.04.003

Fortune, S. (1995). Voronoi Diagrams and Delaunay Triangulations. In D.-Z. Du, & F. Hwang, *Computing in Euclidean Geometry* (pp. 225–265). doi:10.1142/9789812831699_0007

Fortune, S. (2017). Voronoi Diagrams and Delaunay Triangulations. In C. D. Toth, J. O'Rourke, & J. E. Goodman, *Handbook of discrete and computational geometry* (p. 706). CRC press.

Fowler, R. J., & Little, J. J. (1979, 8). Automatic extraction of Irregular Network digital terrain models. *Proceedings of the 6th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1979,* 199–207. doi:10.1145/800249.807444

Garnero, G., & Godone, D. (2013, 1). Comparisons between different interpolation techniques. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 40,* 139-144. doi:10.5194/isprsarchives-XL-5-W3-139-2013

Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th ed.). Pearson.

Grussenmeyer, P., & Al Khalil, O. (2002, 10). Solutions for Exterior Orientation in Photogrammetry: A Review. *The Photogrammetric Record, 17*, 615–634. doi:10.1111/0031-868x.00210

Gupta, R. P. (2018). Digital Elevation Model. In *Remote Sensing Geology* (pp. 101–106). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-662-55876-8_8

Hackel, T., Wegner, J. D., & Schindler, K. (2016, 6). Fast Semantic Segmentation of 3D Point Clouds with Strongly Varying Density. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 3*, pp. 177–184. Copernicus GmbH. doi:10.5194/isprs-annals-III-3-177-2016

Hirt, C. (2014). Digital Terrain Models. In *Encyclopedia of Geodesy* (pp. 1–6). Cham: Springer International Publishing. doi:10.1007/978-3-319-02370-0_31-1

Höhle, J., & Potucková, M. (2011). *Assessment of the Quality of Digital Terrain Models.* Frankfurt: EuroSDR.

Hough, P. V. (1962, 3). *US Patent No. 3,069,654.*

Howes, D., & Gatrell, A. (1993). Visibility analysis in GIS: Issues in the environmental impact assessment of windfarm developments. *Proceeding of the Fourth European Conference and Exhibition on Geographical Information Systems,* (pp. 861–870).

Hu, X., & Yuan, Y. (2016). Deep-Learning-Based Classification for DTM Extraction from ALS Point Cloud. *Remote Sensing, 8*. doi:10.3390/rs8090730

Hubert, M., Rousseeuw, P. J., & Vanden Branden, K. (2005, 2). ROBPCA: A new approach to robust principal component analysis. *Technometrics, 47,* 64–79. doi:10.1198/004017004000000563

Journel, A. G., & Huijbregts, C. J. (1976). *Mining geostatistics.* United Kingdom: Academic Press,London, England.

Jutzi, B., & Gross, H. (2009). Nearest neighbour classification on laser point clouds to gain object structures from buildings. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 38,* 4–7.

Kraus, K., & Pfeifer, N. (1998, 8). Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing, 53,* 193–203. doi:10.1016/S0924-2716(98)00009-4

Li, Z., Zhu, Q., & Gold, C. (2004). *Digital terrain modeling - principles and methodology.* CRC-Press.

Limberger, F. A., & Oliveira, M. M. (2015, 6). Real-time detection of planar regions in unorganized point clouds. *Pattern Recognition, 48,* 2043–2053. doi:10.1016/J.PATCOG.2014.12.020

Linder, W. (2009). *Digital photogrammetry: A practical course.* Springer Berlin Heidelberg. doi:10.1007/978-3-540-92725-9

Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2005). *Geographical Information Systems and Science* (2nd ed.). John Wiley & Sons, Ltd.

Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2005). *Geographical Information Systems and Science* (2nd ed.). John Wiley & Sons, Ltd.

Mark, L. H. (2010, 1). *Principal Component Analysis For ICP Pose Estimation Of Space Structures.* mathesis, Ryerson University, Ottawa.

Matheron, G. (1963, 12). Principles of geostatistics. *Economic Geology, 58*, 1250. doi:10.2113/GSECONGEO.58.8.1246

Mitas, L., & Mitasova, H. (2005). Spatial Interpolation. In P. A. Longley, M. F. Goodchild, D. J. Maguire, & D. W. Rhind, *Geographic Information Systems: Principles, Techniques, Management and Applications* (2nd ed., Vol. 1, pp. 481-490). Wiley.

Mongus, D., & Žalik, B. (2012). Parameter-free ground filtering of LiDAR data for automatic DTM generation. *ISPRS Journal of Photogrammetry and Remote Sensing, 67*, 1-12. doi:https://doi.org/10.1016/j.isprsjprs.2011.10.002

Nar, F., Yilmaz, E., & Camps-Valls, G. (2018, 10). Sparsity-driven digital terrain model extraction. *International Geoscience and Remote Sensing Symposium (IGARSS). 2018-July*, pp. 1316–1319. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/IGARSS.2018.8517569

Nurunnabi, A., Belton, D., & West, G. (2012). Robust segmentation in laser scanning 3D point cloud data. *2012 International Conference on Digital Image Computing Techniques and Applications, DICTA 2012*. doi:10.1109/DICTA.2012.6411672

Oude Elberink, S., & Maas, H.-G. (2000, 1). The use of anisotropic height texture measures for the segmentation of airborne laser scanner data. *IAPRS, XXXIII*, 678-684.

PDAL Contributors. (2018, November). PDAL Point Data Abstraction Library. doi:10.5281/zenodo.2556738

Pouliou, T., Kanaroglou, P. S., Elliott, S. J., & Pengelly, L. D. (2008, 2). Assessing the health impacts of air pollution: a re-analysis of the Hamilton children's cohort data using a spatial analytic approach. *International journal of environmental health research, 18*, 21-22. doi:10.1080/09603120701844290

Putz, V., & Zagar, B. G. (2008). Single-Shot Estimation of Camera Position and Orientation using SVD. *2008 IEEE Instrumentation and Measurement Technology Conference*, (pp. 1914-1919). doi:10.1109/IMTC.2008.4547360

Pyysalo, U., Oksanen, J., & Sarjakoski, T. (2009). Viewshed analysis and visualization of landscape voxel models. *24th international cartographic conference, Santiago, Chile, 15.*

Rabbani, T., Heuvel, F. A., & Vosselman, G. (2006, 1). Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36*.

Remondino, F. (2003). From point cloud to surface: the modeling and visualization problem. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXIV-5/W10*. doi:10.3929/ETHZ-A-004655782

Rovers, A., de Vreede, I., Rook, M., Psomadaki, S., & Nagelkerke, T. (2015). *Semantically Enriching Point Clouds The Case of Street Level.* Technical Report, TU Deflt.

Rusu, R. B., & Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). *2011 IEEE International Conference on Robotics and Automation*, (pp. 1-4). doi:10.1109/ICRA.2011.5980567

Snavely, N., Seitz, S. M., & Szeliski, R. (2008). Modeling the world from Internet photo collections. *International Journal of Computer Vision, 80,* 189–210. doi:10.1007/s11263-007-0107-3

Sohn, G., & Dowman, I. (2002, 1). Terrain surface reconstruction by the use of tetrahedron model with the MDL criterion. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 34*.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., Hough-Transform, P. G., Ransac, E., Tarsha-Kurdi, F., . . . Grussenmeyer, P. (2007). *Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data. ISPRS Workshop on Laser Scanning.* Tech. rep. Retrieved from https://halshs.archives-ouvertes.fr/halshs-00264843

The American Society for Photogrammetry and Remote Sensing. (2004). *ASPRS Guidelines Vertical Accuracy Reporting for Lidar Data.* ASPRS Lidar Committee (PAD). Retrieved from https://www.asprs.org/a/society/committees/lidar/Downloads/Vertical_Accuracy_Reporting_for_Lidar_Data.pdf

The American Society for Photogrammetry and Remote Sensing. (2019). *LAS Specification 1.4 - R14.* Bethesda, Maryland: The American Society for Photogrammetry and Remote Sensing. Retrieved from http://www.asprs.org/wp-content/uploads/2019/03/LAS_1_4_r14.pdf

VanHorn, J. E., & Mosurinjohn, N. A. (2010). Urban 3D GIS Modeling of Terrorism Sniper Hazards. *Social Science Computer Review, 28*, 482-496. doi:10.1177/0894439309360836

Vieira, M., & Shimada, K. (2005, 11). Surface mesh segmentation and smooth surface extraction through region growing. *Computer Aided Geometric Design, 22*, 771–792. doi:10.1016/J.CAGD.2005.03.006

Vo, A. V., Truong-Hong, L., Laefer, D. F., & Bertolotto, M. (2015, 6). Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing, 104*, 88–100. doi:10.1016/J.ISPRSJPRS.2015.01.011

Vosselman, G. (2000, 1). Slope based filtering of laser altimetry data. *IAPRS, XXXIII.*

Wack, R., & Wimmer, A. (2002). Digital terrain models from airborne laserscanner data-a grid based approach. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34*, 293–296.

Waldhauser, C., Hochreiter, R., Otepka, J., Pfeifer, N., Ghuffar, S., Korzeniowska, K., & Wagner, G. (2014, 4). Automated Classification of Airborne Laser Scanning Point Clouds. *Springer Proceedings in Mathematics and Statistics, 97*, 269–292. doi:10.1007/978-3-319-08985-0_12

Wang, L., Zhen, Z., Zhang, X., & Sato, M. (2016, 5). Adaptive camera control method for efficient stereoscopic photography. *Proceedings of the IEEE International Conference on Industrial Technology, 2016-May*, 726–731. doi:10.1109/ICIT.2016.7474840

Wang, Y., & Feng, H.-Y. (2015, 5). Outlier detection for scanned point clouds using majority voting. *Computer-Aided Design*, *62*, 31-43. doi:10.1016/j.cad.2014.11.004

Watson, D. F., & Philip, G. M. (1984, 5). Systematic triangulations. *Computer Vision, Graphics, and Image Processing, 26*, 217–223. doi:10.1016/0734-189X(84)90184-1

Weinmann, M., Urban, S., Hinz, S., Jutzi, B., & Mallet, C. (2015, 6). Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. *Computers and Graphics (Pergamon), 49*, 47–57. doi:10.1016/j.cag.2015.01.006

Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., & Reynolds, J. M. (2012, 12). 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology, 179*, 300–314. doi:10.1016/j.geomorph.2012.08.021

Wheatley, D., & Gillings, M. (2000). Vision, Perception and GIS: some notes on the development of enriched approaches to the study of archaeological visibility. *Beyond the map: archaeology and spatial technologies*, 1–27.

Woodget, A. S., Austrums, R., Maddock, I. P., & Habit, E. (2017, 7). Drones and digital photogrammetry: from classifications to continuums for monitoring river habitat and hydromorphology. *WIREs Water, 4,* 1222. doi:10.1002/wat2.1222

Xu, Z., & Yang, Z. (2018, 4). EIGENENTROPY BASED CONVOLUTIONAL NEURAL NETWORK BASED ALS POINT CLOUDS CLASSIFICATION METHOD. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-3,* 2017-2022. doi:10.5194/isprs-archives-XLII-3-2017-2018

Zhang, K., Chen, S. C., Whitman, D., Shyu, M. L., Yan, J., & Zhang, C. (2003, 4). A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Transactions on Geoscience and Remote Sensing, 41,* 872–882. doi:10.1109/TGRS.2003.810682

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., & Yan, G. (2016). An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sensing, 8*. doi:10.3390/rs8060501