

AUTOMATIC ROOF PLANE EXTRACTION FROM LIDAR DATA USING RANSAC ALGORITHM

RANSAC ALGORİTMASI KULLANARAK LİDAR VERİSİNDEN OTOMATİK ÇATI DÜZLEMİ ÇIKARILMASI

KHALİL KARBALAI MOHAMMADZADEH

PROF. DR. MUSTAFA TÜRKER

Supervisor

Submitted to Institute of Sciences of Hacettepe University as a
Partial Fulfillment of the Requirements
for the Award of the Degree of Master
in Geomatics Engineering

Jun 2018

This work named “**AUTOMATIC ROOF PLANE EXTRACTION FROM LIDAR DATA USING RANSAC ALGORITHM**” by **KHALIL KARBALAI MOHAMMADZADEH** has been approved as a thesis for the degree of **MASTER OF SCIENCE IN GEOMATICS ENGINEERING** by the below mentioned Examining Committee Members.


Assoc. Prof. Dr. Mehmet Ali HİNDİSTAN
Head



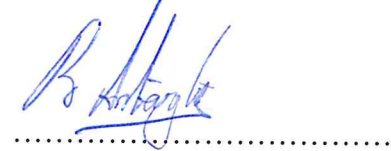
Prof. Dr. Mustafa TÜRKER
Supervisor



Asst. Prof. Dr. Kamil TEKE
Member



Asst. Prof. Dr. Berk ANBAROĞLU
Member



Asst. Prof. Dr. Baran USLU
Member



This thesis has been approved as a thesis for the Degree of **MASTER OF SCIENCE IN GEOMATICS ENGINEERING** by Board of Directors of the institute for Graduate studies in Science and Engineering

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU
Director of the Institute of
Graduate School of Science and Engineering

YAYINLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe Üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

- ☐ **Tezimin/Raporumun tamamı dünya çapında erişime açılabilir ve bir kısmı veya tamamının fotokopisi alınabilir.**

(Bu seçenekle teziniz arama motorlarında indekslenebilecek, daha sonra tezinizin erişim statüsünün değiştirilmesini talep etmeniz ve kütüphane bu talebinizi yerine getirirse bile, tezinin arama motorlarının önbelleklerinde kalmaya devam edebilecektir.)

- ☒ **Tezimin/Raporumun 30.07.2019 tarihine kadar erişime açılmasını ve fotokopi alınmasını (İç Kapak, Özet, İçindekiler ve Kaynakça hariç) istemiyorum.**

(Bu sürenin sonunda uzatma için başvuruda bulunmadığım takdirde, tezimin/raporumun tamamı her yerden erişime açılabilir, kaynak gösterilmek şartıyla bir kısmı ve ya tamamının fotokopisi alınabilir)

- ☐ **Tezimin/Raporumun tarihine kadar erişime açılmasını istemiyorum, ancak kaynak gösterilmek şartıyla bir kısmı veya tamamının fotokopisinin alınmasını onaylıyorum.**

- ☐ **Serbest Seçenek/Yazarın Seçimi**

02 / 07 / 2018



Khalil Karbalai Mohammadzadeh

ETHICS

In this thesis study, prepared in accordance with the spelling rules of Institute of Graduate Studies in Science of Hacettepe University,

I declare that

- all the information and documents have been obtained in the base of the academic rules,
- all audio-visual and written information and results have been presented according to the rules of scientific ethics,
- in case of the using other Works, related studies have been presented according the scientific standards,
- all cited studies have been fully referenced,
- I did not do any distortion in the data set,
- And any part of this thesis has not been presented as another thesis study at this or any other university.

07/06/2018



Khalil Karbalai Mohammadzadeh



To My Parents

ABSTRACT

AUTOMATIC ROOF PLANE EXTRACTION FROM LIDAR DATA USING RANSAC ALGORITHM

Khalil Karbalai Mohammadzadeh

Master of Science, Department of Geomatics Engineering

Supervisor: Prof. Dr. Mustafa TÜRKER

Jun 2018, 70 pages

Automatic image processing and object extraction from airborne data have become an important topic of research in the field of photogrammetry and remote sensing. The aerial laser scanning system, also known as LiDAR, has become the dominant technology for acquiring 3D spatial data from the earth surface with high speed and density. LiDAR's output is an unclassified and unstructured point cloud dataset. Thus, the main process to be performed on this dataset is to classify it into distinct classes. Then, the classified LiDAR data can be used as input to create 3D city models. This data has a number of unique properties that play a fundamental part in their classification process. The main properties include the geometric properties that are obtained through the processes carried out on 3D positions of the points in the cloud. Among these processes is the plane extraction, which is carried out through the most commonly used methods of RANSAC (Random Sample Consensus), Region growing, and Hough Transform.

In this study, the RANSAC algorithm was used to extract planes from building rooftops. The aim is to apply RANSAC on LiDAR point cloud data to extract planes from rooftops. The first and most important step in the extraction process of the planes from rooftops is to identify and distinguish buildings from the other features, such as terrain and vegetation. The second step is to apply the RANSAC algorithm on the point cloud data of the individual buildings. Based on the geometric position and the points' distance to the plane, the least squares method is used to cross the best plane through the candidate points that form the plane.

The experiments were carried out on the selected study areas located in the city of Bergama, Turkey using LiDAR point cloud data collected by the Reigl airborne scanner. The results show that RANSAC's performance is quite good for buildings which have complex roofs and also it has the ability to extract small planes in high density point clouds. Furthermore, the best extracted planes are properly adjusted to the raw point cloud data sets.

ÖZET

RANSAC Algoritması Kullanılarak LIDAR Verisinden Otomatik Çatı Düzlemi Çıkarılması

Khalil Karbalai Mohammadzadeh

Yüksek Lisans, Geomatik Mühendisliği Bölümü

Tez Danışmanı: Prof. Dr. Mustafa TÜRKER

Haziran 2018, 70 sayfa

Hava verilerinden otomatik görüntü işleme ve nesne çıkarımı fotogrametri ve uzaktan algılama alanlarında önemli bir araştırma konusu haline gelmiştir. LiDAR olarak da bilinen hava lazer tarama sistemi, yer yüzeyinden yüksek hız ve yoğunlukta 3D uzamsal verilerin elde edilmesinde kullanılan en temel teknolojidir. LiDAR'ın çıktıları, sınıflandırılmamış ve yapılandırılmamış nokta bulutudur. Dolayısıyla, bu veriler üzerinde gerçekleştirilecek temel işlem, onları ayrı kategoriler halinde sınıflandırmaktır. Sonra, sınıflandırılan LiDAR verileri 3B şehir modellerini oluşturmada giridi verisi olarak kullanılabilir. Bu veriler, sınıflandırma sürecinde kilit rol oynayan benzersiz özelliklere sahiptir. En temel özellikleri, bulut noktalarının 3D konumlarında gerçekleştirilen süreçler yoluyla elde edilen geometrik özellikleridir. RANSAC (Random Sample Consensus – Rastgele Örneklem Konsensüsü), bölge büyütme ve Hough dönüşümü gibi çeşitli yöntemler yoluyla yürütülen düzlemsel çıkarım, bu süreçler arasında yer almaktadır.

Bu çalışmada, bina çatılarından düzlem çıkarmada hızlı ve etkili bir algoritma olan RANSAC kullanılmıştır. Çalışmanın amacı, bina çatılarından düzlem çıkarmada RANSAC'ı, LiDAR nokta bulut verisi üzerinde uygulamaktır. Şehir modellerinin önemli bileşenlerinden olan bina çatılarının çıkarımının ilk ve en önemli adımı, binaları arazi ve bitki örtüsü gibi diğer yapılardan ayırıp tespit etmektir. İkinci adım, RANSAC algoritmasını tekil bina nokta bulutu verisine uygulamaktır. Geometrik konuma ve noktaların düzleme olan mesafelerine dayalı olarak, en küçük kareler yöntemi kullanılarak, düzlemi oluşturan aday noktalar yoluyla en iyi düzlem oluşturulur.

Türkiye'nin Bergama kentinden seçilen çalışma alanlarında Reigl tarayıcı ile toplanan LiDAR nokta bulut verileri kullanılarak testler gerçekleştirilmiştir. Elde edilen sonuçlar RANSAC'ın performansının, kompleks çatı yapısına sahip binalarda ve yüksek yoğunluklu nokta bulutlarında küçük düzlemleri çıkarma becerisi açısından iyi olduğunu göstermektedir. Ayrıca, en iyi çıkarımı yapılan düzlemler, ham nokta bulutu veri setlerine doğru şekilde uyarlanmıştır.

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my advisor Prof. Dr. Mustafa TRKER the continuous support of my Master study, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Master study.

I would also like to thank the experts who were involved in the validation survey for this research project: Assoc. Prof. Dr. Mehmet Ali HNDSTAN, Asst. Prof. Dr. Kamil TEKE, Asst. Prof. Dr. Berk ANBAROĖLU, Asst. Prof. Dr. Baran USLU. Without their passionate participation and input, the validation survey could not have been successfully conducted.

I would like to thanks the GENERAL COMMAND OF MAPPING for providing the point clouds of the study areas in this research.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

TABLE OF CONTENTS

	<u>Page</u>
<i>ABSTRACT</i>	<i>i</i>
<i>ÖZET</i>	<i>iii</i>
<i>TABLE OF CONTENTS</i>	<i>vi</i>
<i>LIST OF TABLES</i>	<i>viii</i>
<i>LIST OF FIGURES</i>	<i>ix</i>
<i>1.Introduction</i>	<i>1</i>
1.1. Thesis Objectives	2
1.2. Study areas	3
1.2.1. The first study area	3
1.2.2. The Second study area	5
1.3. Thesis outline	7
1.4. Software	7
<i>2.The literature Review</i>	<i>8</i>
2.1. Related to RANSAC	8
2.2. Related to LiDAR point cloud segmentation and classification	9
2.3. Related to ground filtering	9
<i>3. Methodology</i>	<i>11</i>
3.1. The ground filtering principle	13
3.2. Ground filtering and classifying non-ground points	16
3.3. Extracting the building class	17
3.4. Identifying the individual buildings	18
3.5. The RANSAC algorithm	20
3.5.1. Plane equation	24
3.5.2. Distance from a point to a plane	24
3.5.3. Least square estimation	25

4. Results and Analysis	28
4.1. The results of the first study area	28
4.1.1. Detection of the ground points	28
4.1.2. Ground filtering and classifying the non-ground points	29
4.1.3. Identifying the individual buildings	31
4.1.4. The extraction of the roof planes using the RANSAC algorithm	33
4.2. The results of the second study area.....	44
4.2.1. Detection of the ground points	44
4.2.2. Ground points filtering and non-ground point classifying	44
4.2.3. Identifying the individual buildings	46
4.2.4. The extraction of the roof planes using RANSAC.....	48
5. Conclusions and Recommendations	62
5.1. Conclusions.....	62
5.2. Recommendations	66
REFERENCES.....	66
CURRICULUM VITAE.....	70

LIST OF TABLES

	<u>Page</u>
Table 1 For the first study area, the number of points according to return numbers.....	3
Table 2 For the second study area, the number of points according to return number ...	5
Table 3 The ASPRS standard LIDAR point classes that are used in <i>LasClassify</i>	18
Table 4 The parameter values used for the trials conducted	30
Table 5 The classification parameters defined	45
Table 6 The processing time for RANSAC results	64

LIST OF FIGURES

	<u>Page</u>
Figure 1.1 First study area Google Earth images	3
Figure 1.2 The point cloud of the first study area	4
Figure 1.3 Three street level sample pictures in the first study area taken from the Google Earth images	4
Figure 1.4 The location of the second study area on Google Earth Images	5
Figure 1.5 The point cloud of the second study area	6
Figure 1.6 A, B and C represent three views of the second study area taken	6
Figure 3.1 The flowchart of the proposed plane extraction method	12
Figure 3.2 The seed points.....	13
Figure 3.3 A TIN surface generated over the seed points	13
Figure 3.4 a_1, a_2, a_3 , are the angles between the nodes of the triangle and the point...	14
Figure 3.5 The TIN surface that represents ground.....	14
Figure 3.6 The octree subdivision principle.....	19
Figure 3.7 The separated building's points with different colors	20
Figure 3.8 Classical RANSAC for detecting roof facets.....	22
Figure 3.9 The working mechanism of the RANSAC algorithm.....	23
Figure 3.10 The shortest distance D from point $p (P_x, P_y, P_z)$ to plane $P (a, b, c)$	25
Figure 3.11 Point cloud in two different coordinate system.....	26
Figure 4.1 The filtered ground points of the first study area	28
Figure 4.2 The separated non-ground points.....	29
Figure 4.3 The non-ground points classified as high vegetation and buildings.....	31
Figure 4.4 The points classified as buildings only.....	31
Figure 4.5 A , the first octree level (cell size=71.7365), B , The fourth (4 th) octree level .	32
Figure 4.6 The eighth (8 th) Octree level (cell size=0.560441)	32
Figure 4.7 The separated buildings that were stored individually.....	33

Figure 4.8 The comparison of the raw data and RANSAC results obtained using two different threshold values (10 cm and 20 cm) for building	34
Figure 4.9 Different objects on the roofs.	35
Figure 4.10 Comparison of the raw data and RANSAC results obtained using two different threshold values (10 cm and 20 cm) for building #20.....	36
Figure 4.11 The RANSAC results	41
Figure 4.12 The Google Earth's Picture and the RANSAC results for building #4.....	42
Figure 4.13 The Google Earth's Picture and the RANSAC Results for building #20.....	42
Figure 4.14 The Google Earth's Picture and the RANSAC Results for building #8.....	43
Figure 4.15 The Google Earth's Picture and the RANSAC Results for building #23.....	43
Figure 4.16 The ground points of second study area	44
Figure 4.17 The non-ground points of the second study area	44
Figure 4.18 The non-ground classified points	46
Figure 4.19 The classified building points	46
Figure 4.20 A, The first octree level (cell size=75.9458), B, 5 th octree level	47
Figure 4.21 A, Second test area 9 th octree level (cell size=0.296663).....	47
Figure 4.22 The separated buildings that were stored individually.....	48
Figure 4.23 The side view and top view of the building planes extracted through the RANSAC algorithm	57
Figure 4.24 The illustration of the planes extracted for Building #1.....	58
Figure 4.25 A, building (Building #10) with the extracted parallel planes.....	59
Figure 4.26 The planes extracted from Building #19. The gap within inside the plane ...	59
Figure 4.27 A. the building #28 with small window, B, the building # 14 with four facets type.....	60
Figure 4.28 Buildings #22, #28, and #7 with the overlapping tall trees.....	61
Figure 5.1 The comparison of the raw data (A) and RANSAC result (B) for buildings #1 and #8.....	65

1. Introduction

LIDAR (Light Detection and Ranging) congregates laser distance meter, IMU (Inertial Measurement Unit), and DGPS (Differential GPS) in a single combined unit [1]. Technology provides us with innovative solutions to obtain 3D real-time location data as well as novel ways of gaining high definition space-time territorial location data. In addition to promptly acquiring altitude information, LIDAR introduces several technological innovations in Remote Sensing mapping and other application areas. It bears a great potential of contributing to fields such as terrain surveying and mapping, city planning, environmental monitoring, oceanology, geology, planet science etc. [1]

LIDAR incorporates distinct assets. LIDAR has an operating laser pulse sensor and it keeps away from the impact of sun's shadow angle by diminishing their effect on obtaining information. If LIDAR is contrasted with photogrammetry, it is apparent that LIDAR has certain advantages, such as keeping all necessary data (from 3D to 2D), having precise altitude detection, and gaining multi-beam echo to obtain high-density data. Also, it makes considerable attenuation in ground control survey, aviation routes may immediately adapt, enhances the degree of mechanization, and swiftly and serially generate digital elevation models (DEM), digital surface models (DSM) and digital Ortho-photo maps (DOM) [1].

Two of the crucial functions of most surveying fields are rapidly gaining 3D data and self-regulating administration of data. Airborne laser scanning systems produce 3D data with perfect precision, high speed and density. Therefore, this method is predominantly applied in urban areas. There are two consecutive procedures for building up 3D urban models [2]. The initial procedure is automatic segmentation of the point cloud into three categories as terrain, buildings and vegetation. After the partitioning of the urban cloud, the modelling of buildings may be passed on.

Two methods, data-driven and model-driven exist in the doctrine in terms of building modelling. The model-driven methods basically review a database of basic building models to find out the most convenient one [3]. According to [3], there are a number of parameters to identify a basic building. Implicit in this assumption is the measurement of these parameters prior to building up the 3D model. When it comes to data-driven method, it attempts to generate the most accurate and closest multifaceted model by reproducing every single section of the buildings point cloud [1]. One of the most important functions

in the data-driven methods that generates more comprehensive models is the automatic detection of planes. Several techniques, including but not limited to growing, 3D Hough-transform and RANSAC (Random Sample Consensus), have been suggested to implement this process.

Although being rather straightforward, Region-Growing techniques have issues with segmentation and are over-responsive to data noise as well as being time consuming [4]. It gets challenging to identify a termination criterion in region-based techniques in the event that changes over across regions occurs easily [5].

The Hough transform technique is used for identifying parameterized objects as well as discovering flat and circular shapes in 2D circumstances. In terms of 3D point-cloud plane segmentation, each position in Hough space is associated with a plane in the space of the entity. The Hough transform technique is very successful in detecting shapes even in bad datasets. Nevertheless, this method entails a great deal of memory and CPU allocation [4].

The RANSAC (Random Sample Consensus) algorithm [6], which is a straightforward and robust method, is generally administered for identifying model parameters through data which might be distorted by extreme values. RANSAC is a convenient technique to deal with problems of robust estimation because it can perform well in the presence of large amounts of extreme values [7]. According to Tarsh-Kurdi et al. [2], RANSAC is a more powerful technique in comparison to Hough transform.

1.1. Thesis Objectives

The main objectives of the thesis are as follows:

- To extract roof planes of complex geometries using the RANSAC algorithm;
- To extract roof planes of complex buildings with polyhedral roofs by optimizing the threshold value of the RANSAC algorithm;
- To apply the algorithm in the selected test areas in Bergama, İzmir using the LIDAR point cloud data,
- To apply the Least Square estimation algorithms to LIDAR point cloud data sets

1.2. Study areas

1.2.1. The first study area

The first study area (Figure 1.1) was selected from the residential region of Bergama, which is located in the north of the city of Izmir. In Turkey, this is the only place where airborne LiDAR point cloud data exist. The LiDAR point cloud data was collected by the RIEGL scanner with 18.77 points density per square meter. The center coordinates of this study area are as follows: 39° 05' 57" N and 29° 09' 25" E.



Figure 1.1. The first study area Google Earth image.

The size of this study area is about 24858 square meters. Additionally, this area contains 466609 points (Figure 1.2) and regarding the return numbers, they can be categorized as given in Table 1. An emitted laser pulse in the LiDAR system can have up to five returns based on the features it is reflected. The first return will be denoted as the return number one, the second as the return number two, and so on.

Table 1. For the first study area, the number of points according to return numbers.

Return Number	Number of Points
1 st return	412512
2 nd return	48220
3 rd return	5543
4 th return	322
5 th return	12

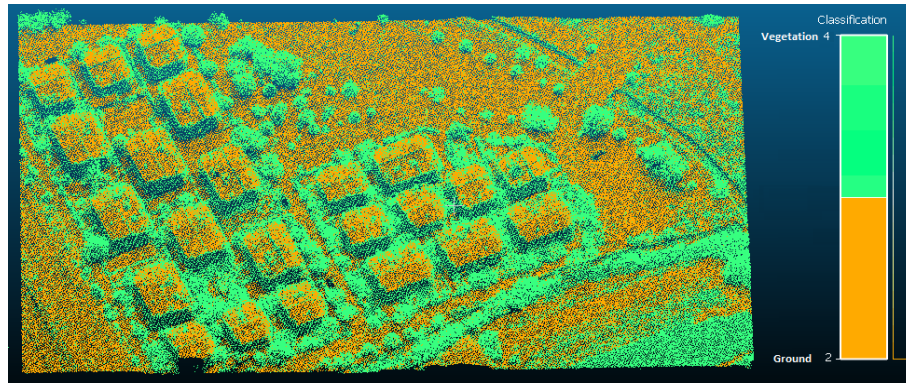


Figure 1.2. The point cloud of the first study area.

The area is an old residential region (Figure 1.3) and situated in the south of Bergama. Based on the Google earth pictures, there are 23 buildings with different roofing models in this study area.



Figure 1.3. Three street level sample pictures regarding the study area taken from the Google Earth Images.

1.2.2. The second study area

The second study area (Figure 1.4) was also selected from the residential part of the Bergama and is situated in the southwest part of Bergama. The LiDAR data was collected using the Reigle scanner. The center coordinates of this test area are as follows: $39^{\circ} 06' 28''$ N and $27^{\circ} 09' 55''$ E.



Figure 1.4. The location of the second study area on Google Earth Images.

This test area consists of 367512 points (Figure 1.5) with 19.28 point density per square meter and regarding the return numbers they can be categorized as shown in Table 2. Further, the size of this study area is about 19121 square meter.

Table 2. For the second study area, the number of points according to return numbers.

Return Number	Number of Points
1 st return	319353
2 nd return	43920
3 rd return	4038
4 th return	192
5 th return	9

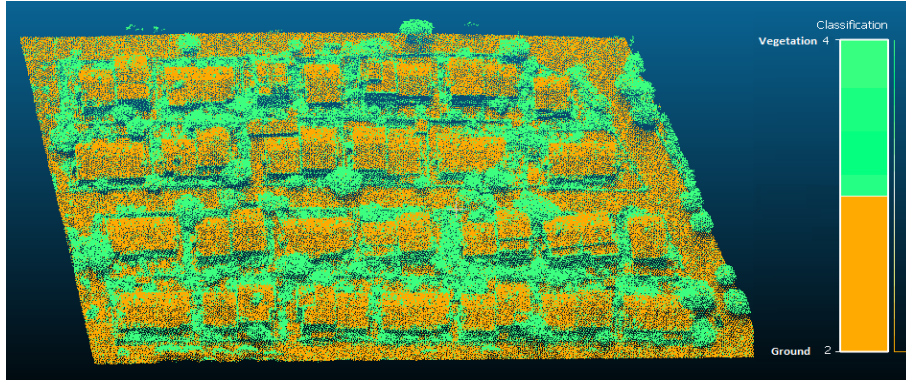


Figure 1.5. The point cloud of the second study area.

This area consists of 43 buildings with various shapes. Several views taken from Google Earth Images are illustrated in Figure.1.6.

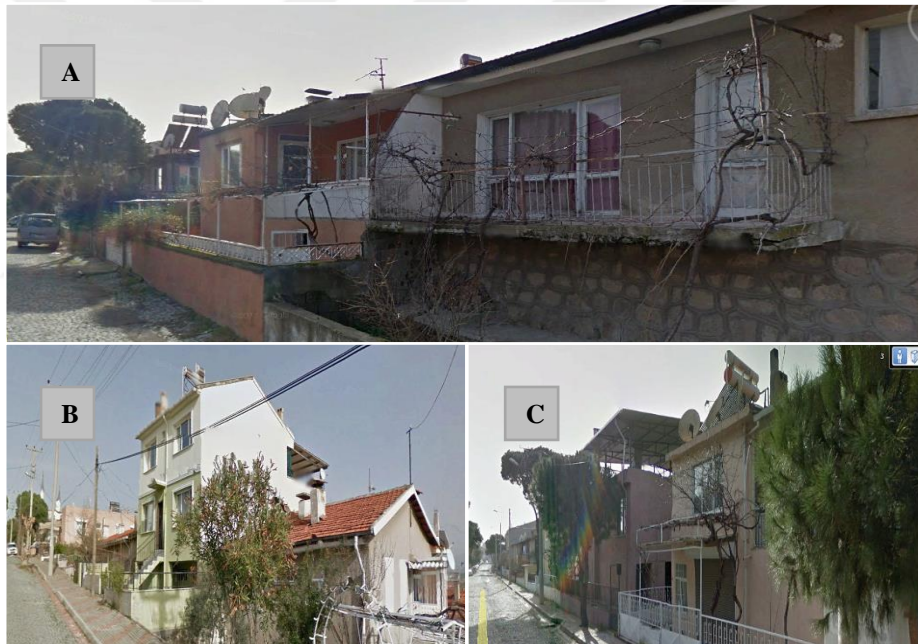


Figure 1.6. A, B and C represent three views of the second study area taken from Google Earth Images.

1.3. Thesis Outline

The remaining part of this thesis is categorized in four sections. Chapter 2 represents the review of the methods used for the extraction of roof planes from point cloud data sets using the RANSAC algorithm. Chapter 3 refers to the methodology used in the present case to extract building roof planes using the RANSAC algorithm. In Chapter 4, the results obtained from the experiments carried out on the study areas and discussions regarding the implementation of the RANSAC algorithm are given. Finally, in Chapter 5, the conclusions of this thesis study are provided.

1.4. Software

The software used to conduct this study are as follows:

- The RANSAC algorithm was implemented through a program written in the *MATLAB* programming environment.
- The ground filtering and point cloud classification operations were carried out using various functions of the *LASTOOLS* software.
- The point cloud data sets were separated to sub-clouds using the *Cloud Compare* software.

2. The Literature Review

2.1. Related to RANSAC

Region growing segmentation algorithms are not always intelligible and are not implemented on a regular basis. For that reason, in a study carried out by [2], 3D Hough transform and RANSAC algorithm were used to automatically establish 3D roof planes from LIDAR point cloud data. Since Hough-transform is very delicate to segmentation parameter values, the extended RANSAC algorithm was used to segment the building planes. It was observed that the extended approach gives satisfying results for weak point density and different levels of building complexity.

In the work of [8], the boundary of a lake was automatically extracted from LIDAR point cloud data. To do that water surface was established using the RANSAC algorithm. It was suggested that since lake areas are mostly planar their areas can be easily established using the RANSAC algorithm.

In a study conducted by [9] to establish plain floor surface or details of any building etc., best candidates for planes that fix the buildings and land surfaces were determined by implementing the RANSAC algorithm, which is a more efficient technique than the other methods. The results obtained by the RANSAC algorithm were reported to be satisfactory.

In a study carried out by [10], the surface of shapes such as a plane, cone, cylinder and ball that takes place in 3D topographic LIDAR cluster were automatically established through RANSAC algorithm. To test the validity of the process, the manually and automatically extracted surfaces were compared. It was concluded that the RANSAC algorithm is very efficient in automatically establishing the surfaces from 3D topographic LIDAR point cluster data. It was suggested that the RANSAC algorithm can be successfully used in establishing planes of the objects that contain plane surfaces.

2.2. Related to LIDAR point cloud segmentation and classification

In a study carried out by [11], segmentation problems, such as losing data during the transformation of irregular point clouds into other models were resolved and the high computational cost was reduced by optimization.

In the work of [12], the planes were extracted by means of segmentation over the point cloud dataset. It was stated that the method can simultaneously extract multiple roof planes and the spatial relationship between data points is intended.

In the work of [13], building regions were determined automatically. Then, curvature-based segmentation technique was used to establish the roof planes. To construct many-sided structural models, the shapes of boundaries of roof planes were determined by grouping the roof planes. In addition, in order to enhance the reconstructed models' geometric quality, they examined the subject of integrating air images for the reconstruction process.

In the work of [14], a new approach that involves roof segmentation and roof model reconstruction process were developed to establish roof planes from LiDAR point cloud data. As segmentation is effective in establishing the geometric structure of roofs, reconstruction was observed to be better in determining the contiguity and integrity of the roofs. The multiphase level set method was tested on two laser data sets. The study concludes that, if sufficient amount of points do not exist to represent roof structures, segmentation will not express the objects adequately. The study further mentions that the suggested approach can be tested using the data of different roof structures and complicated buildings.

In a study carried out by [15], 3D roof models were constructed using the LIDAR data only. Based on the results achieved it was suggested that higher quality results can be obtained by incorporating 3D lines established through using the images.

2.3. Related to ground filtering

The study conducted by [16] presents algorithms for the ground filtering to generate ground surface, the classification of the non-ground points, and the generation of DEMs. This study also identifies the ground points from the point cloud data using the Minimum Description Length criterion parameter, which is declared to be an important parameter to identify ground points. The advantages of the algorithms presented in their work were that,

(i) the ground surface was created from the connected points in a TIN, (ii) the points of low ground surface were always contained and (iii) the original data points were used in the creation of the surface.

In a study conducted by [17], several analyses were carried out by defining three search windows at small, medium, and large scales over the threshold elevation difference to filter terrain points. This method is said to be robust to different problems such as resolution of data, objects complexity and diversity slopes.



3. Methodology

The goal of this research study was to extract roof planes from LiDAR point cloud data. The steps followed in the methodology are shown in a flowchart in Figure 3.1. The raw LiDAR point cloud data to be processed is available in *.Laz* or *.txt* format. First, the ground points are identified and separated from the point cloud data. Then, the remaining non-ground points are classified into buildings and tall trees. Next, the building class that includes integrated information about all buildings is extracted through a classification process. Since the building information is required separately the building class is divided into sub-building cloud points therefore.

The MATLAB scripts code reads the data points separately, one-by-one. The roof planes are extracted from the point cloud as follows: Three random points are chosen to form an imaginary plane to which the distances from all points are computed. By comparing the distances with the specified threshold, the points closer than the threshold to the imaginary plane are found and saved. The best estimation for the plane with the highest number of points is approximated and these points are extracted from the raw dataset. The process is repeated until less than three points remain in the data set and the roof planes are saved with the building number as the final result.

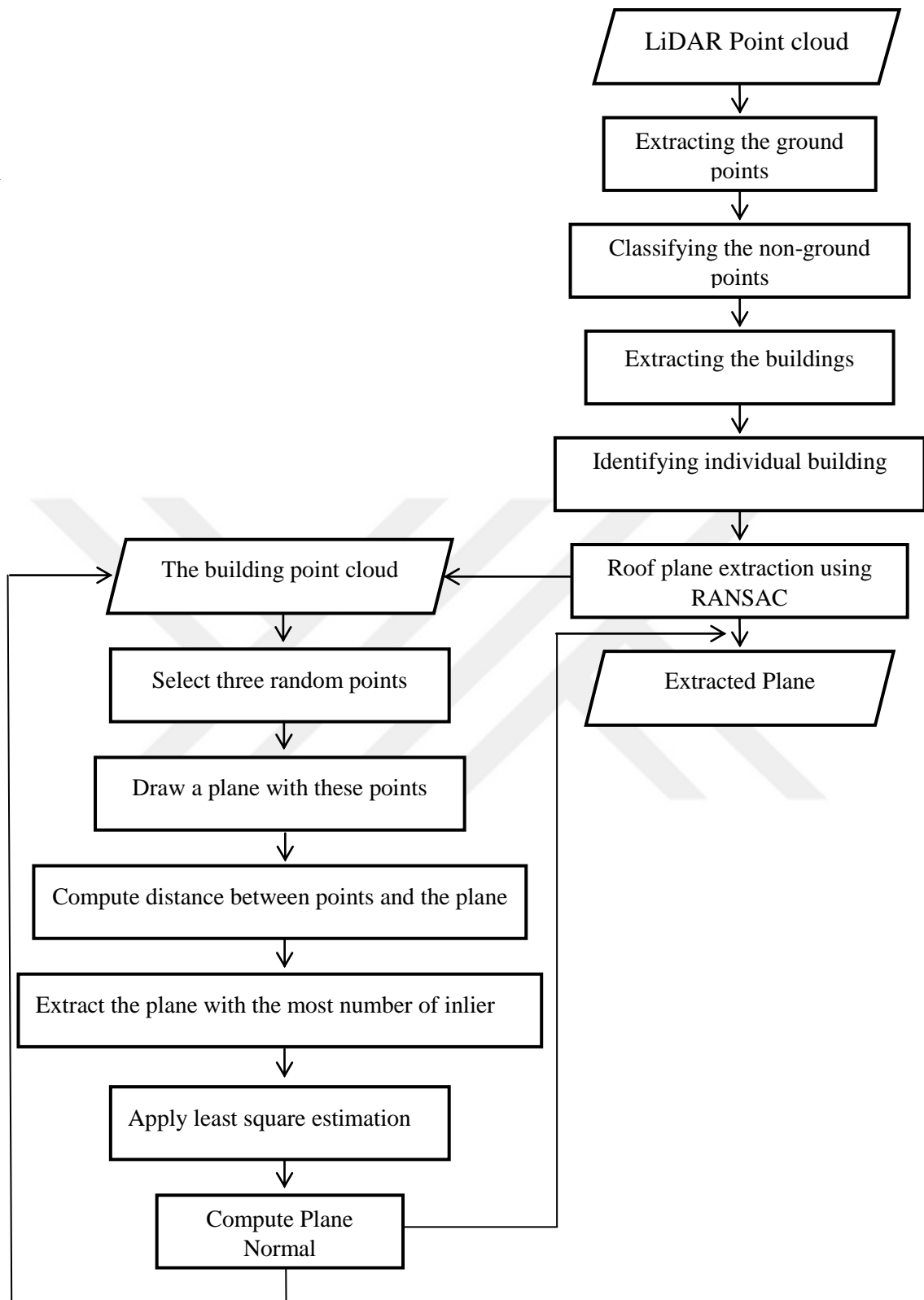


Figure 3.1. The flowchart of the proposed plane extraction method.

3.1. The Ground filtering principle

Detecting the ground points is the first step in this study. To extract objects and render a Digital Elevation Model (DEM), the LIDAR data points are classified into two classes. The first class consists of the ground points, which are extracted from point cloud data and are categorized as ground. Then, the remaining points are categorized as vegetation and building.

A surface is generated below to the randomly distributed laser points using the triangulated irregular network (TIN) processing. Statistics characterizing the distance to the TIN facets are computed for all data points [16]. Based on the computed statistics, seed points are chosen, such as the red points shown in the example in Figures 3.2 and 3.3.

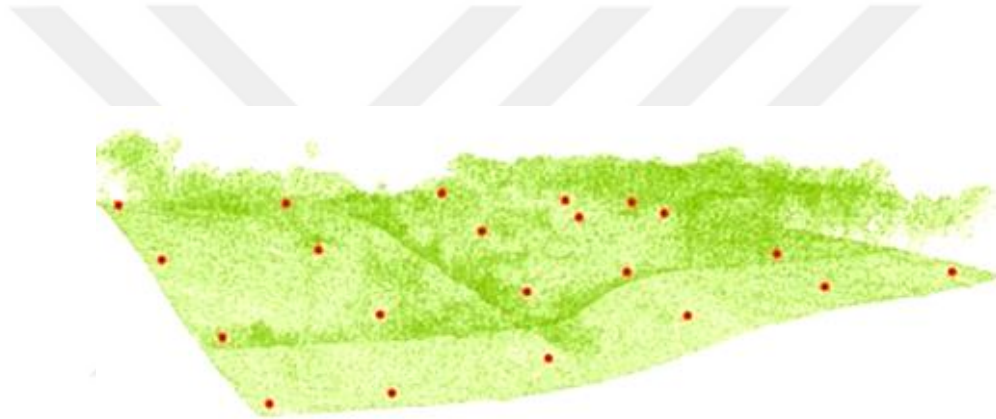


Figure 3.2. The seed points.

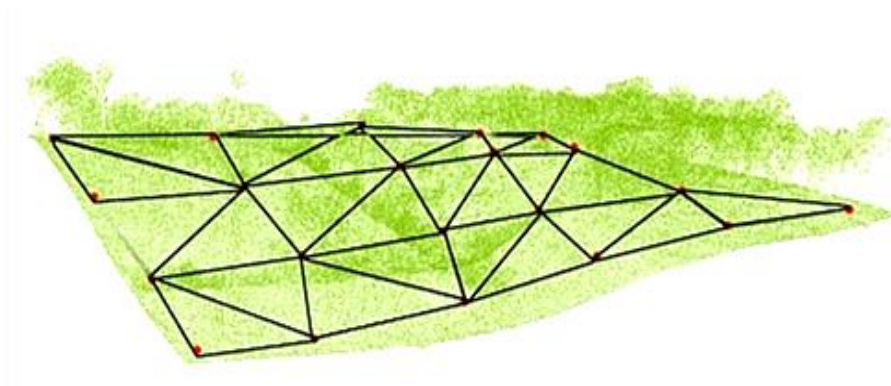


Figure 3.3. A TIN surface generated over the seed points.

In the next stage, new points join the terrain class and the surface, based on the selected threshold values, is iteratively densified. There are two threshold parameters to be used for this densification process: (i) angle to the nodes, and (ii) distance to the TIN faces [16] (Figure 3.4). At the end, the final TIN that represents the ground is achieved (Figure 3.5).

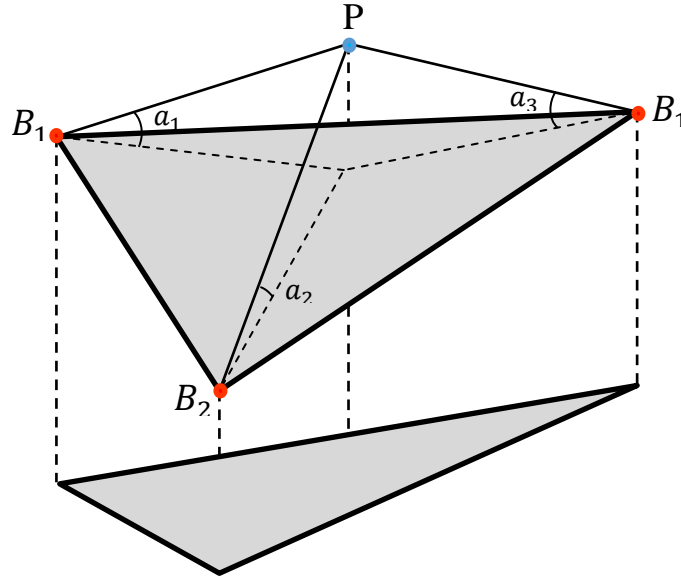


Figure 3.4. a_1, a_2, a_3 , are the angles between the nodes of the triangle and the point being considered and B_1, B_2, B_3 are TIN nodes.

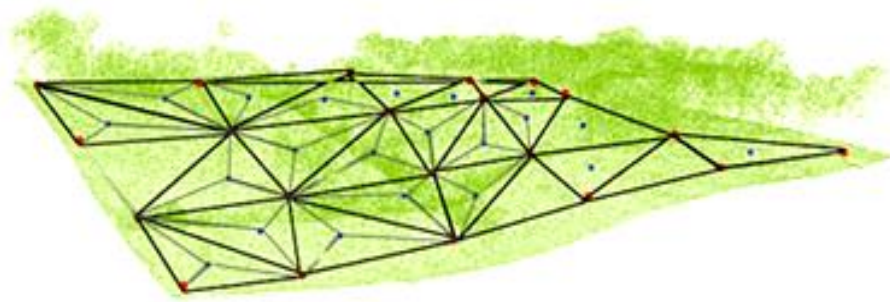


Figure 3.5. The TIN surface that represents ground.

In this study, the ground filtering process was carried out using the *LasGround* function of the *LasTools* Software [18]. Completely, the *LasGround* function consists of four important parameters that control the filtering results [19].

- The first and important parameter is the step size. This parameter is defined according to the region type and the object dimensions in the area to be filtered. The default step size is 5 meters which is most suitable for mountains and forest areas. For urban areas or flat terrains, this value is increased to 10 meters. In the present case, after several trials it was explored that the value of 25 meters gives excellent results for urban areas and warehouses. Therefore, in this study, the step size parameter value was taken as 25 meters.
- The second parameter is the spike value. This parameter has two sub-parameters that are up-spike and down-spike. In the coarsest TIN, these parameters remove the points that stay above and below the defined threshold values.
- The third parameter is the Maximum offset. This parameter defines an offset distance from the ground surface over which the points are not included.
- The fourth parameter is the Standard Deviation. This parameter defines the maximum standard deviation in centimeter for the planar patches.

All of these parameters can be set manually. However, they can also be predefined using the default strategies defined as follow:

- ✓ Forest or hills
- ✓ Town or flats
- ✓ City or warehouses
- ✓ Metropolis

Parameters for these strategies are set based on the practices carried out for various land cover types. Furthermore, there are five standards describing the violence of the terrain that might be used irrespective of the strategy. These five standards are defined as two groups. The first group simplifies the search and is suitable for the flat terrains ('Coarse' or 'Extra coarse'), while the second group amplifies the search for initial ground points and is used for the very inclined hills ('Fine', 'Extra fine' or 'Ultra fine').

Therefore, for the test areas, after selecting the required suitable parameter values for the *LasGround* function, the ground points were detected from the point cloud.

3.2. Ground filtering and classifying non-ground points

In this step, first the ground points are eliminated from the point cloud data sets using the *Las2Las* function with the ‘Dropping with classification number’ option [20]. Then, the non-ground points are classified into different classes using the *LasClassify* function.

In general, categorizing non-ground points into different classes that include vegetation (tall trees) or buildings is quite challenging. To classify non-ground point cloud, LIDAR data rely on a geometrical parameter that allows one to measure the roughness of planar and ruggedness objects, including buildings and trees. Moreover, the planarity of the neighboring points (x_i) around a point, x_0 can be estimated using the principle components analysis (PCA) based on surface points with covariance matrix [21]. A positive, semi-definite covariance matrix is computed as follows [22]:

$$C = \sum_{i=1}^k (x_i - \bar{x})(x_i - \bar{x})^T \quad i = 1, \dots, k \quad (1)$$

Where, C denotes the mean of x (cloud points). The PCA of matrix C generates real eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_2$ with the corresponding Eigen vectors v_0 , v_1 and v_2 on an orthogonal basis of R^3 . The eigenvalue λ_0 evaluates the variance of x_i in v_0 direction, and v_0 approximates the surface normal of x_i . A constant surface changes of the k -nearest neighbors in point cloud data sets is given by [22]:

$$l = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \quad (2)$$

Where, l denotes the planarity and $\lambda_0, \lambda_1, \lambda_2$ represent the Eigen values. With a plane terrain surface $\lambda_0 \ll (\lambda_1 \approx \lambda_2)$, the parameter l is close to zero. Therefore, it is necessary to select a suitable threshold (T) to extract the planar and rugged surfaces accurately. For example, points where $l > T$ can be identified as trees. The threshold is a function of surface roughness and noise in the data and changes for different sets of data.

Although the threshold value is proven to work for extracting buildings, it may fail to identify pruned trees in urban areas. For example, trimming the trees near overhead power lines changes the shape of their canopies and consequently, their roughness. That is contingent upon LIDAR classification criteria assuming trees as rugged surfaces. Furthermore, large trees with small surface roughness may be identified as buildings.

The **LasClassify** function classifies buildings and tall trees into the *las/laz* file [23]. This requires ground points and the elevation of each point from the ground, calculated in **LasHeight**. The function attempts to identify the neighboring points at 2 meter above the ground. By default, *planarity* and *rugged* values are 0.1 and 0.4, respectively. Furthermore, the elevation threshold can be changed via *ground offset*. In case of noisy data, identifying planar surfaces is difficult; therefore, '*planarity*' is changed according to the regional conditions to extract better results.

Moreover, a low point density leads to inaccurate results and **LasClassify** to fail. In other words, the minimum number of pulses required to identify a building roof is 2 per square meter. That is, if pulses per square meter are less than 2, the search window value, which is 2 m by default, is increased to 4 m [23].

The function is run after selecting the best parameters. With this operation, the points that belong to buildings are classified as building class and are given the class number of 6 based on the default values of the **LasTools** software.

3.3. Extracting the building class

The **Las2Las** function reads and writes LIDAR data in *Laz*, *Las*, and *ACSII* formats to filter, project, transform, thin, etc. on point cloud data [20]. One of the most used option in this study is filter. The filter option includes two types of flags. These flags are defined based on the classification number, return number, and point cloud features and the data set is classified based on these flags. For example, keeping or dropping the building points using the 'keep/drop with the classification number' flags. Some of the other examples for the Keep/drop flags are as follow;

- The points that are inside a defined rectangle,
- The Points which are between determined height,
- The points that are a certain return, and
- The points that scale angle has above certain threshold or between some intensity.

As a result, in this study, the points that belong to building class were separated from the previously classified non-ground points by using the ‘Keep with classification number’ option. To do that the default class numbers of the *LasClassify* Function of the *LasTools* software were used (Table 3).

Table 3. The ASPRS Standard LIDAR Point Classes that are used in *LasClassify* [24].

Class Type	Class No.	Class Type	Class No.
Never Classified	0	Water	9
Unclassified	1	Rail	10
Ground	2	Road Surface	11
Low Vegetation	3	Overlap	12
Medium Vegetation	4	Wire Guard	13
Height Vegetation	5	Wire Conductor	14
Building	6	Tower	15
Noise	7	Wire Connector	16
Key point	8	Bridge deck	17

3.4. Identifying the individual buildings

Next, the integrated point cloud dataset corresponding to buildings are separated to individual building’s points by the ‘Label connected component’ function of the Cloud Compare software. In this tool, there are two parameters [25]. Of these parameters, the first and important one is the Octree level.

Octree is a regular and recursive subdivision in three dimensional space. The cubical box that belongs to points set is divided into 8 similar cubes. This operation is recursively performed for each cube. The process stops when no points are in the cube or predefined threshold level is reached. Such a structure gives, the capability to quickly determine what points is in a determined cube and in its surrounding cubes. Hence, the nearest neighbors derivation and equivalent processes become very fast [26].

It is possible to assort selected point cloud by the Octree which consists of many groups of points adequately distant from the others. This common algorithm is normally utilized in 2D binary but has been extended to 3D binary grid. The function exports one entity for each sub cloud of points that belong to building. The higher level of Octree will produce smaller distance threshold, subsequently more subgroups will be extracted. More memory is needed for the larger octree levels. Therefore, the octree level is a sensitive parameter and is difficult to adjust priori, regardless of the experience. Hence, it is necessary to reach an approach by trial and error.

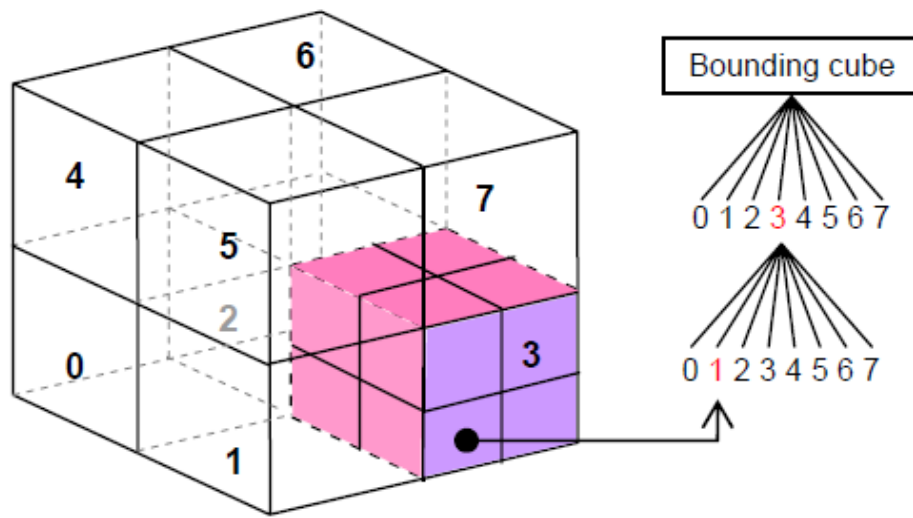


Figure 3.6. The octree subdivision principle [26].

The second important parameter is the minimum number of points for each component. If the number of points in a portion of the defined value is lower than defined value, they are extracted as a separate building. Moreover, the minimum number of points is determined according to the density of the points and the dimensions of the smallest building.

Further, the Random Colors option describes the cloud compare to provide random colors to each new cloud. At the end, the point cloud data sets that belong to individual buildings are extracted and saved separately with different colors [25].

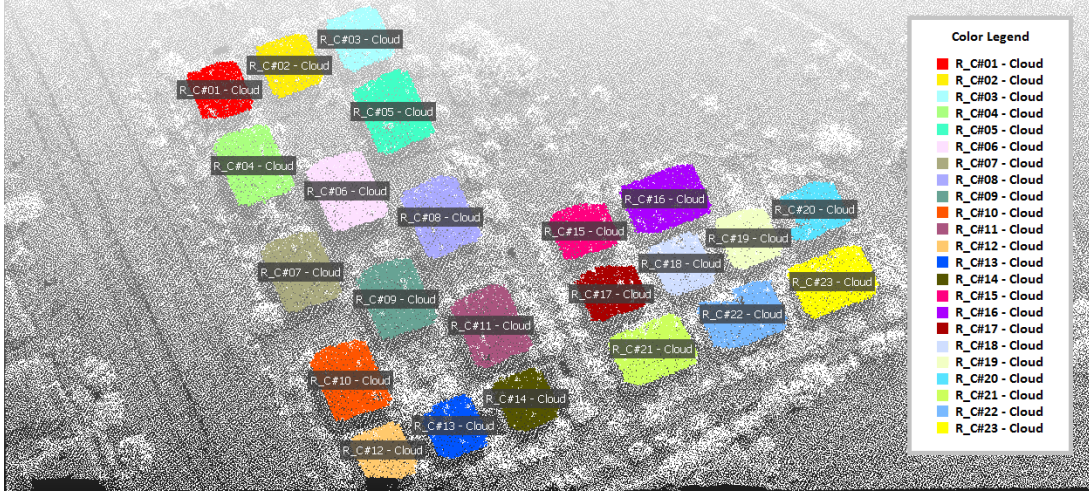


Figure 3.7. The separated building's points with different colors.

3.5. The RANSAC Algorithm

In this section, the RANSAC algorithms was applied on building point cloud data, one building at a time, to extract the roof planes. Ficher and Bolles [6] introduced the RANSAC algorithm, which is a general robust approach in order to estimate model parameters. To achieve an initial solution, instead of utilizing a number of data and then try to remove invalid points, RANSAC applies the smallest possible data set, such as the point cloud of a building, and from this data set, the possible large planes are crossed. The steps for the extraction of the roof planes using the RANSAC algorithm are as follows [27]:

- It randomly selects planar surfaces P from the LIDAR data points S and maintains a number of points where the Euclidean distance from the plane is less than the critical distance d_{cr} .
- The Least Square Estimation of the final plane (p_{final}) is performed along the set of supporters maximum cardinal (M_k).
- The set M_k is eliminated from the initial point cloud S .
- The algorithm operates as long as $card(S) < 3$, where $card(S)$ is the set S cardinal.

Theoretically, the assured best plane is drawn at each repetition, every triplets of S should be tried. Therefore, N_{th} can be defined as (Eq. 3) [27]:

$$N = \frac{Card(S)!}{3!(Card(S)-3)!} \quad (3)$$

Where, S refers to data points.

Furthermore, the detection of the planes can be highly time consuming. Mostly, it is not feasible to apply all possible draws. In other words, the probability of drawing t a correct plane P (which is three points without outlier), considering all points are inlier (w^3 for 3 points), the probability w can be maximized. A relationship between t , w and N can be computed using Eq. 4 [27] as follows:

$$(1 - t) = (1 - w^3)^N \Leftrightarrow N = \frac{\log(1-t)}{\log(1-w^3)} \quad (4)$$

Therefore, the number of draws N might be directly calculated from t and w . If t is constant as 0.99, w has to be estimated with a former knowledge. The general idea of this approach is to enhance the efficiency of performance of a classical RANSAC approach with the focus on the drawing of triplets over previously separated point cloud data. In this study, the directions of the extracted major planes correspond to roof planes orientations. The RANSAC algorithms are given below.

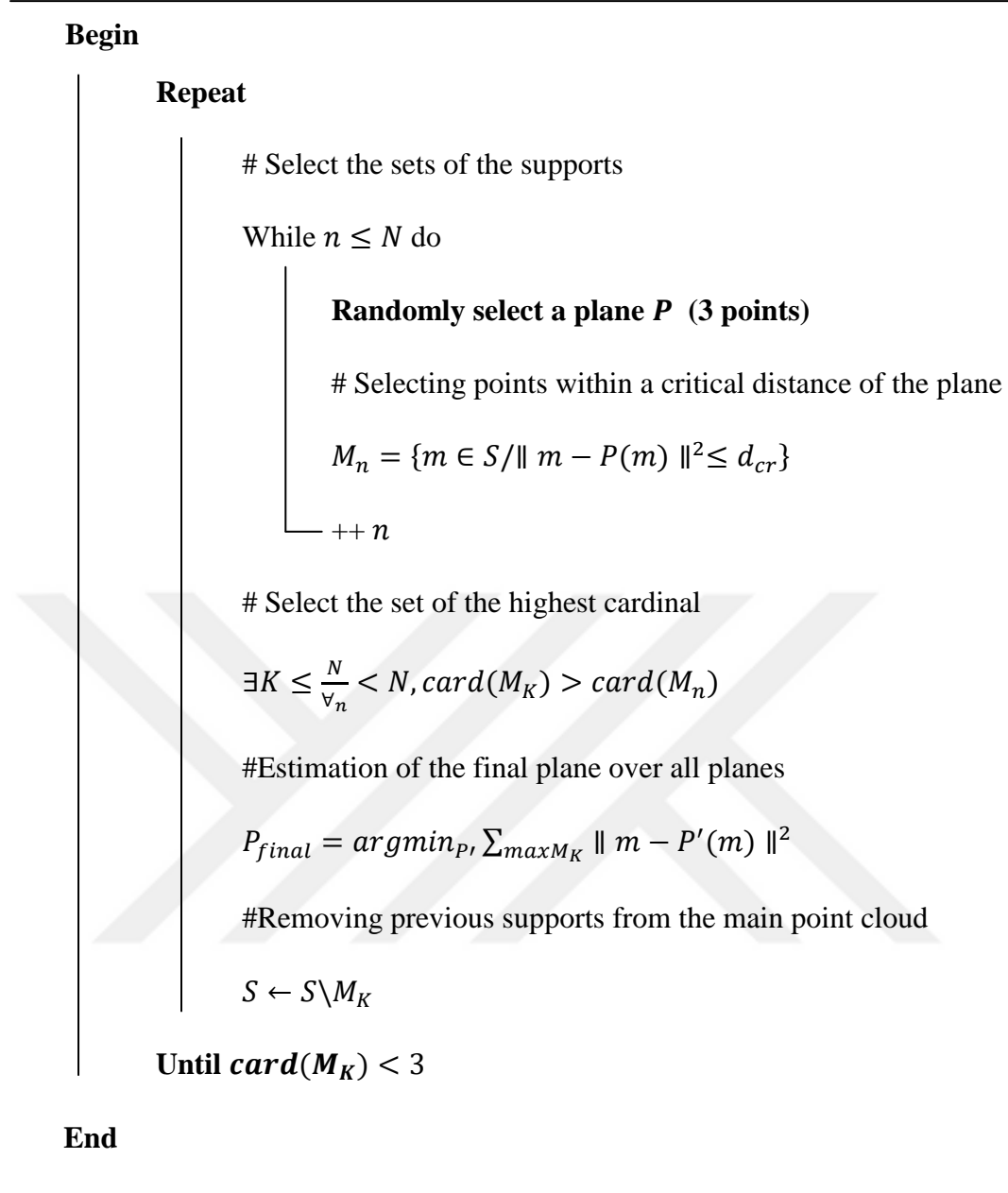


Figure 3.8. Classical RANSAC for detecting roof facets.

In order to implement the algorithm, a code was written in MATLAB programming environment. In this algorithm, the distance threshold is an important and sensitive parameter and directly effects the results. The working mechanism of the RANSAC algorithm is illustrated in Figure 3.9. In Figure 3.9 (A), the threshold value is very low. In this case, a new parallel plane can be formed with the assumption plane, In Figure 3.9(B), the threshold value is defined very high, in this situation. The normal vector of the extracted plane is not due to the normal vector of the building roof in the same direction

because it approximates a wide range of points to the plane, and finally in Figure 3.9(C), the distance threshold between points and the plane is determined according to the structure of the building roof and density of the points. In this study, the threshold value is determined based on the method of trial and error.

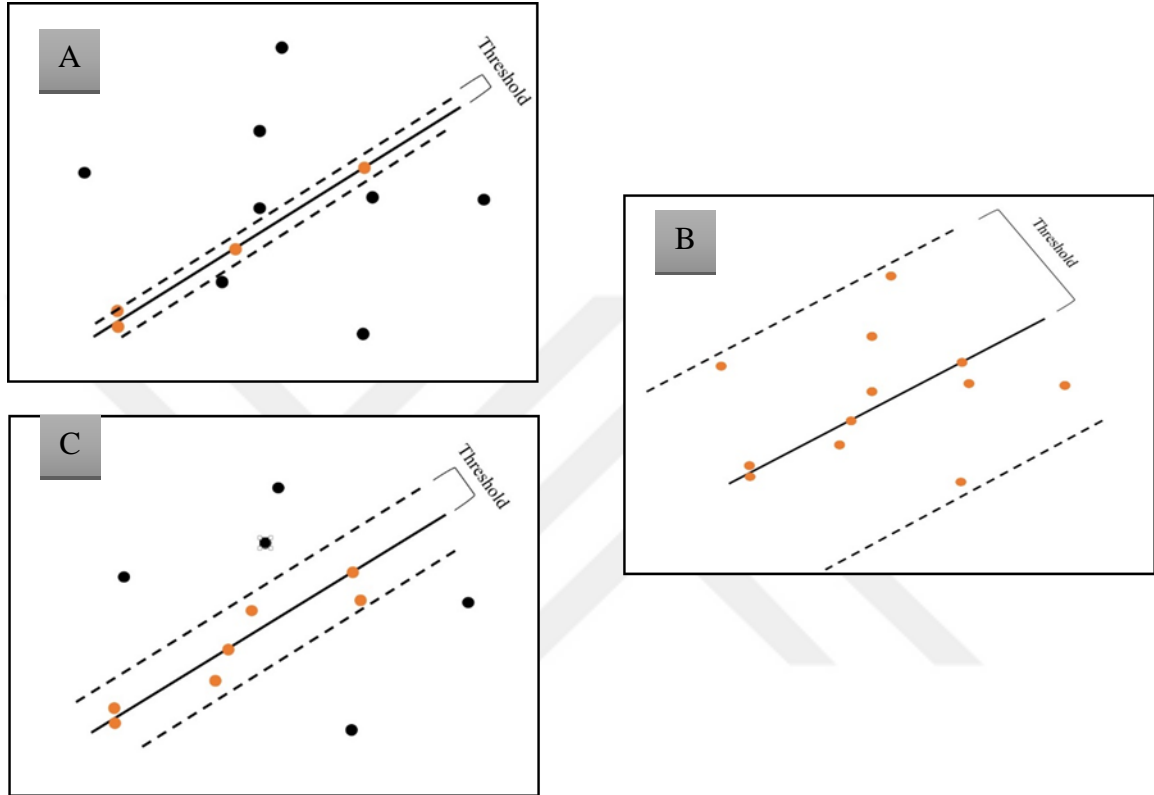


Figure 3.9. The working mechanism of the RANSAC algorithm. The distance threshold from the assumption plane in a test point cloud data.

For a building, the RANSAC MATLAB script performs the following steps to extract planes of the building.

3.5.1. Plane equation

First, three random points are selected from the point cloud data set and a plane crosses from these points. The properties of the extracted plane such as normal vector and distance to the origin of the coordinate system are also computed. This operation is carried out N times for the point cloud that belong to a building being considered. The iteration number is computed using the equation given in Eq. 4

The Cartesian equation of a plane which crosses through three non-collinear points- $p_1(x_1, y_1, z_1)$, $p_2(x_2, y_2, z_2)$, $p_3(x_3, y_3, z_3)$, is shown as follows [28];

$$a \cdot x + b \cdot y + c \cdot z + d = 0 \quad (5)$$

Where, (a, b, c) are the normal vector \vec{N} elements and d denotes the distance to origin of the coordinate system. Three points (p_1, p_2, p_3) can label two distinct vectors $\overrightarrow{p_1p_2}$ and $\overrightarrow{p_1p_3}$. As the two vectors is on the plane, their cross product can be used as the normal to the plane

3.5.2. Distance from a point to a plane

After computing the plane elements, the shortest distance from each point to the detected plane are calculated. For a point $p_1 = (x_1, y_1, z_1)$ that is out of the plane $P(ax + by + cz + d = 0)$, the shortest distance D from p_1 to plane P (Figure 3.10) is computed as follows [29]:

$$D = \frac{|ax_1 + by_1 + cz_1 + d|}{\sqrt{a^2 + b^2 + c^2}}. \quad (6)$$

The point, p_1 lies in the defined plane, P if and only if $D = 0$. If $\sqrt{a^2 + b^2 + c^2} = 1$ which means that a, b and c are normalized and then the equation becomes

$$D = |ax_1 + by_1 + cz_1 + d|. \quad (7)$$

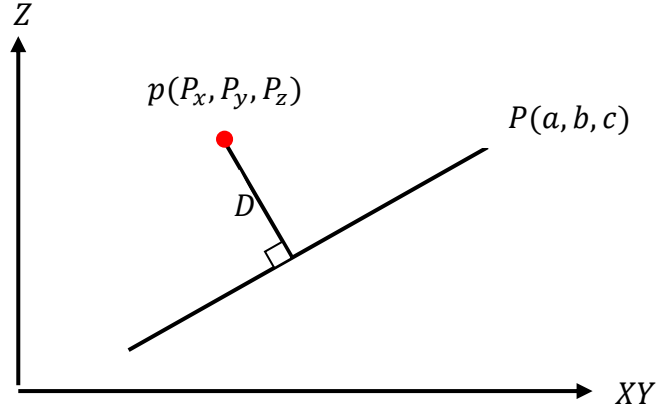


Figure 3.10. The shortest distance D from point $p (P_x, P_y, P_z)$ to plane $P (a, b, c)$

Next, the computed distances are compared based on the previously defined threshold value. Then, the points whose distances are below the threshold value are selected as the inlier points. This operation is carried out N times. Finally, the plane which includes the most inlier points and its number of inlier points are extracted as the candidate points to implement the Least Square estimation algorithms.

3.5.3. Least Square Estimation

After extracting the inliers points that belong to a roof plane, the next processing step is to approximate the best fitting plane. This is carried out using the least square estimation, which is a method for estimating the parameters by minimizing the squared discrepancies between the points that belong to the plane. The objective is to adjust the parameters of a model function to achieve the best fit for the point cloud data set. In this study, the data set consists of n points with x , y , z coordinates, ($i = 1, \dots, n$), where, x_i and y_i are independent variables and z_i is a dependent variable the value of which is computed from the raw point cloud.

First, the mean values \bar{X} , \bar{Y} and \bar{Z} are computed from the x_i , y_i and z_i coordinates of the points that belong to plane. Next, the variance-covariance matrix C_i is compute for each data point using Eq. 8 [22].

$$P_i(P_{xi}, P_{yi}, P_{zi}) \quad , \quad \bar{P}(\bar{X}, \bar{Y}, \bar{Z})$$

$$C_i = \frac{1}{k} \sum_{i=1}^k (P_i - \bar{P})(P_i - \bar{P})^T \quad i = 1, 2, \dots, k \quad (8)$$

$$C_i = \begin{bmatrix} cov(x_i, x_i) & cov(x_i, y_i) & cov(x_i, z_i) \\ cov(y_i, x_i) & cov(y_i, y_i) & cov(y_i, z_i) \\ cov(z_i, x_i) & cov(z_i, y_i) & cov(z_i, z_i) \end{bmatrix} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix} \quad (9)$$

Where, P_i is a data point with x_i, y_i, z_i coordinates, $\bar{P} (\bar{X}, \bar{Y}, \bar{Z})$ is the mean vector of all points, and k is the number of data points.

Next, the eigenvector is computed. To do that a total variance-covariance matrix, \bar{C}_P is computed by summing up the previously calculated variance-covariance matrices.

$$\bar{C}_P = \sum_{i=1}^k C_i \quad (10)$$

$$\bar{C}_P = \begin{bmatrix} \overline{\sigma_{xx}} & \overline{\tau_{xy}} & \overline{\tau_{xz}} \\ \overline{\tau_{yx}} & \overline{\sigma_{yy}} & \overline{\tau_{yz}} \\ \overline{\tau_{zx}} & \overline{\tau_{zy}} & \overline{\sigma_{zz}} \end{bmatrix} \quad (11)$$

Then, the eigenvector matrix is computed from the calculated total variance-covariance matrix. The first column of this eigenvector matrix represents the normal of the plane which best approximates the data points. The number λ is an eigenvalue of \bar{C}_P if and only if $\bar{C}_P - \lambda I$ is singular [30].

$$\det(\bar{C}_P - \lambda I) = 0 \quad (12)$$

The Eigen values $\lambda_1, \lambda_2, \lambda_3$ are computed of solve equation (12). Also an Eigen vector, is extracted of solve the $(\bar{C}_P - \lambda I)X = 0$ for each $(\lambda_1, \lambda_2, \lambda_3)$.

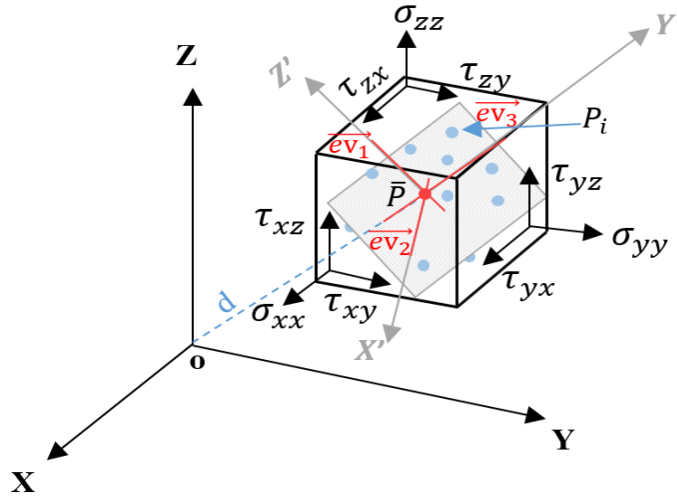


Figure 3.11. Point cloud in two different coordinate systems.

As shown in Figure 3.11, $\vec{ev}_1 = \lambda_1 = \vec{N}(a, b, c)$, $\vec{ev}_2 = \lambda_2$, $\vec{ev}_3 = \lambda_3$, and o denotes the origin of the Cartesian coordinate system, and d is the distance between the origin of the Cartesian coordinate system and origin \bar{P} of the rotated coordinate system $X'Y'Z'$.

Next, the distance d from, $\bar{P} (\bar{X}, \bar{Y}, \bar{Z})$ to the origin of the coordinate system is computed using the inner multiplication of the normal vector \vec{N} and \bar{P} . Then, the x and y coordinates are kept constant and the z coordinates are calculated from the plane equation (Eq. 14).

$$d = \vec{N} * \bar{P} \rightarrow d = [a \quad b \quad c] * \begin{bmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{bmatrix} \quad (13)$$

Thus, according to the plane equation (Eq. 5) the new Z values can be calculated as [29]:

$$Z_i = \frac{-(a*x_i+b*y_i+d)}{c} \quad (14)$$

After computed the new Z values for the point cloud data sets, the best plane coordinates are extracted as (P_{xi}, P_{yi}, P_{zi}) .

4. Results and Analysis

4.1. The result of the first study area

4.1.1. Detection of the Ground Points

The first step in the extraction of building plane is the detection of the ground points. The detection of ground points was carried out using the *LasGround* function of the *LasTools* LiDAR data processing software. As described inspection 3.1, the detection of the ground points depends on several parameters that are determined by default based on the characteristics of the study area and its topography.

The selected first study area is a small part of the city of Bergama and with smooth terrain. The *LasGround* function produces better results in the towns with smaller building dimensions rather than their step size. Due to the existence of large size of buildings in this area, the value of the step size parameter has been changed. Based on this knowledge, the Town and Flat options of *LasGround* were selected for the detection of ground points. Furthermore, the height of every single point was calculated from the ground with the *Compute Z* flag or *LasHeight* function [19]. The extracted ground points in this study area are illustrated in Figure 4.1.

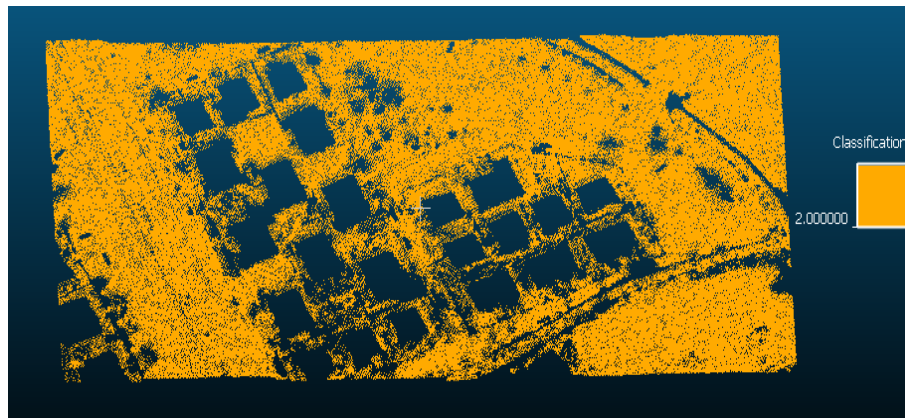


Figure 4.1. The filtered ground points of the first study area.

4.1.2 Ground filtering and classifying the non-ground points

The cloud points were divided into two groups, which are (i) ground and (ii) non-ground after identifying them. Based on the *LasTools* software [18], the first group (ground) was labeled as class 2, while the second group (non-ground) was classified as buildings, tall trees etc. The filtering of the points can be carried out with ‘filter with drop classification number’ flag in *Las2Las* script [20]. Therefore, with the use of this function, the ground points were filtered and extracted from the dataset as shown in Figure 4.2.

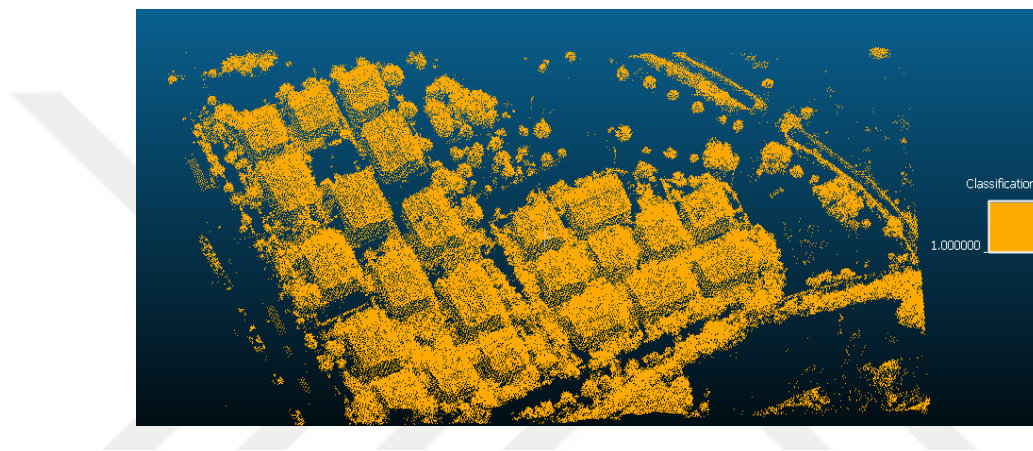


Figure 4.2. The separated non-ground points.

Next, the non-ground points were classified as buildings and tall trees. As mentioned in section 3.1, the classification of non-ground points (carried out using the *LasClassify* function of *LasTools*) depends on the below given several important parameters [23]:

- 1) The parameter ‘search area size’ detects neighboring points that are located at the minimum defined from the ground. If the density of the points is less than two pulses per square meter, the default value of this parameter is increased to 4 meters. Since in the first study area the density of the points is more than two pulses per square meter, the value for this parameter was not changed.
- 2) The second parameter is ‘planarity’. This parameter computes standard deviation for the point cloud data sets in order to extract the roughness of the surface. For this study area, the value for this parameter was increased due to terrain characteristics and the point density. Moreover, good results were achieved for the planer surfaces when the parameter value for the planarity was selected as 0.2.

- 3) The third parameter is ‘forest ruggedness’. The value for this parameter varies with the amount of vegetation and tall trees. Due to low density and low number of trees present in this study area, the value for this parameter was not modified in order to detect and classify the plants that fall in this area.
- 4) Accurate determination of the value for the parameter ‘ground offset’ reduces the duration of the classification and increases the classification accuracy. Based on the calculations in *LasHeight* function, the roof of buildings in this study area is higher than the pre-defined value for this parameter. Therefore, to achieve satisfactory results, the value of this parameter was chosen close to the heights of the buildings.

Therefore, after conducting several trials, the most appropriate classification parameter values for the first study area were determined as given in Table 4.

Table 4. The parameter values used for the trials conducted. The parameter values of the 3rd trial have been defined to provide the optimum value set.

TRIAL	1	2	3	4	5
SEARCH AREA SIZE (M)	1.5	2	2	3	2
BUILDING PLANARITY	0.1	0.1	0.2	0.1	0.3
FOREST RUGGEDNESS	0.4	0.2	0.4	0.6	0.4
GROUND OFFSET (M)	2	2	3	2	1

After determining the best parameter values for the *LasClassify* function, the non-ground points were classified as shown in Figure 4.3. Next, the building class was separated from the classified non-ground points. The separated final building class is shown in Figure 4.4.

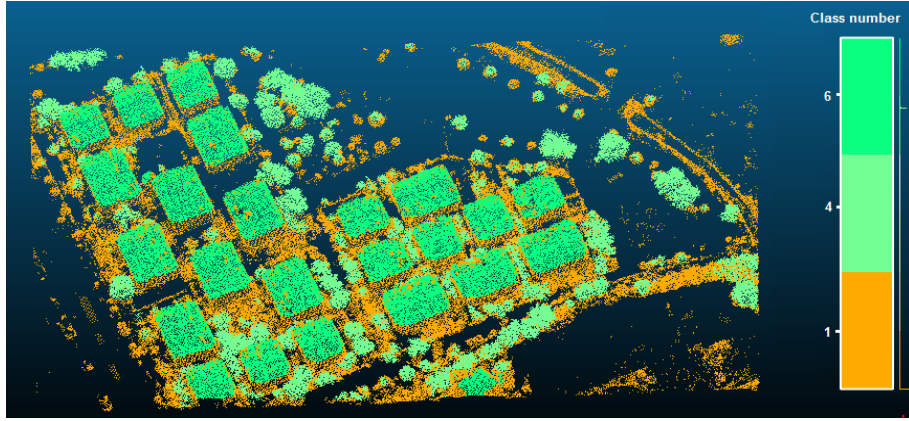


Figure 4.3. The non-ground points classified as high vegetation and buildings.

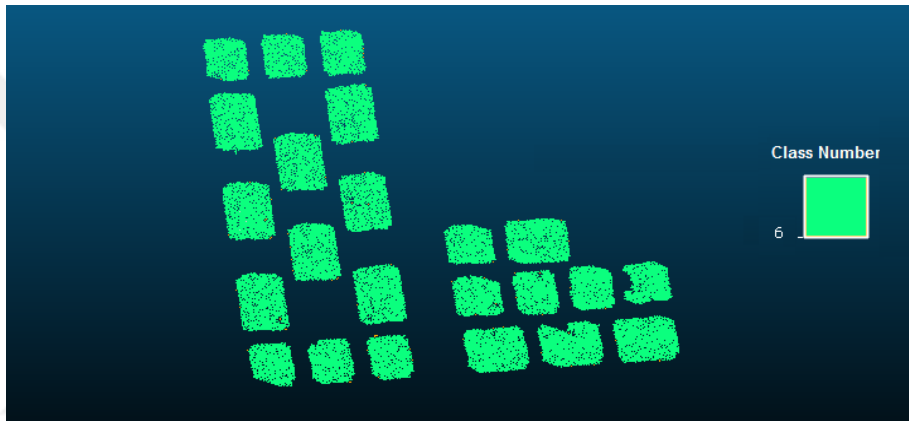


Figure 4.4. The points classified as buildings only.

4.1.3 Identifying the individual buildings

In the extracted building class dataset, the buildings are not separate and all are stored as integrated. However, to extract roof planes of the buildings the building point cloud dataset should be processed one building at a time in order to effectively apply the RANSAC algorithm. To do that the point cloud that belongs to each building was separated from the building point cloud dataset. This was performed using the Cloud Compare software. To do that, the ‘Label connected component’ option of the Cloud Compare software was used [25]. This option has two parameters that are the ‘octree level’ and the ‘minimum points per component’.

The parameter ‘Octree Level’ defines the minimal threshold between the compact sub-clouds. In the Cloud Compare software, based on the point density and sample size, the first level of this parameter was calculated. The variation of the Octree level has a direct relationship with the number of identified sub-clouds [25]. In other words, the number of the extracted sub-clouds is increased if the value of the octree level increases. In this study area, the point density is 18.77 points per square meter and also the sample size is computed automatically. Based on this knowledge therefore, the *eight* octree level was found to be the most suitable level for the extraction of individual buildings in this study area (Figure 4. 6). Also the first and fourth octree levels were calculated for the illustrative purpose Figure 4. 5

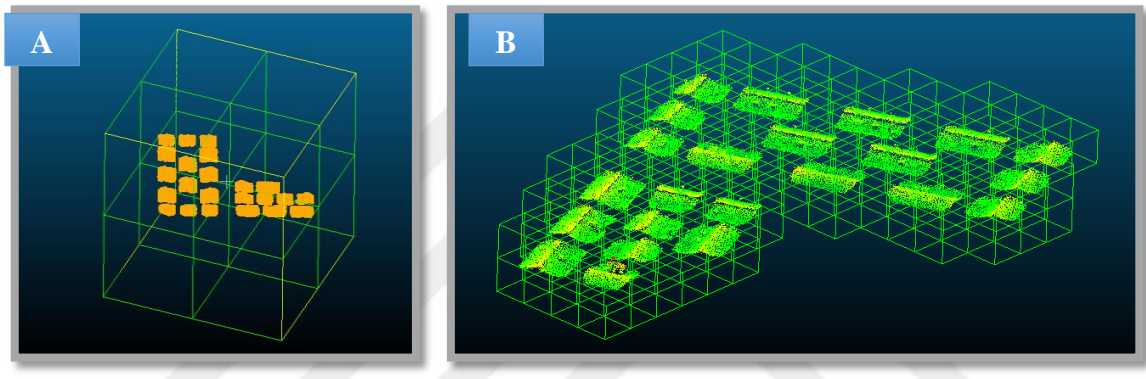


Figure 4.5. A, the first octree level (cell size=71.7365), B, The fourth (4th) octree level (cell size=8.96706).

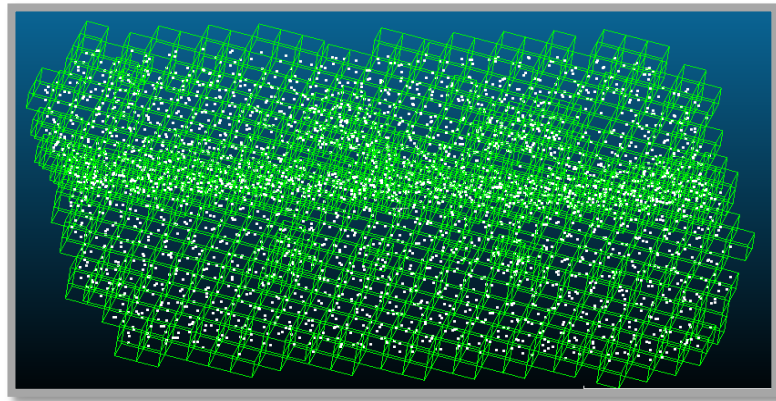


Figure 4.6. The eighth (8th) Octree level (cell size=0.560441).

The minimum number of points per component is denoted by the ‘Min. points per components’ parameter. If this value is lower than the defined threshold, which is previously computed based on the dimensions of the buildings that fall in the study

area, the sub-cloud points cannot be detected as separate buildings. For example, if the minimum dimensions of the buildings to be separated are 2 m x 3 m then, the number of points to be extracted for each building would be about 100 (18.77 points x 3 m x 2 m). Therefore, for this study area, the minimum number of points for each component was selected to be 100. And, the extracted individual buildings were saved separately with the use of this parameter value, as shown in Figure 4.7.

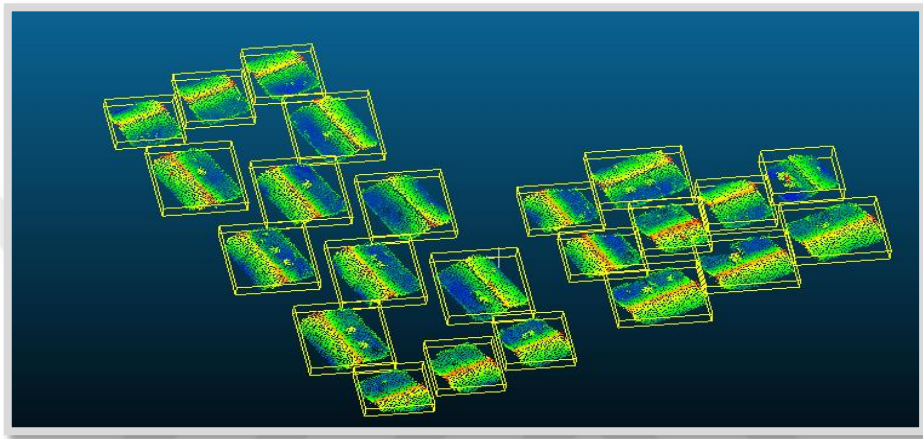


Figure 4.7. The separated buildings that were stored individually.

4.1.4. The extraction of the roof planes using the RANSAC algorithm

The point cloud dataset that belongs to each individual building was imported to the developed MATLAB script for the RANSAC algorithm. There are three important parameters for the RANSAC algorithm written in the MATLAB script, that are the threshold, probability of drawing the correct plane, and the probability of choosing an inlier [27].

- The first parameter is ‘Threshold’, which refers to Euclidean distance from each point to the plane. The point density is the most important factor for defining the value for this parameter. For this study area, the RANSAC algorithm was run using two threshold values of 10 cm and 20 cm.



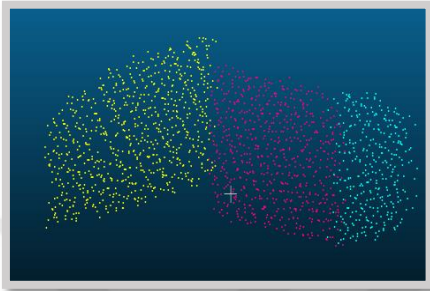
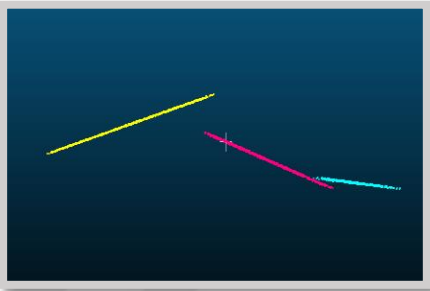
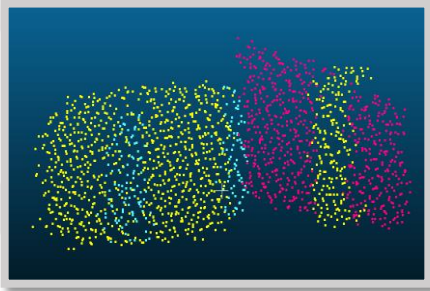
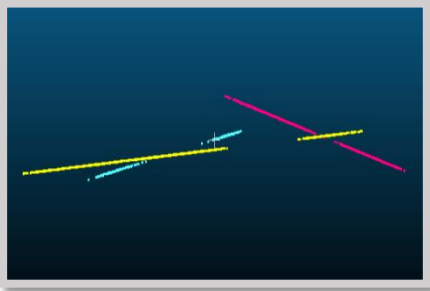
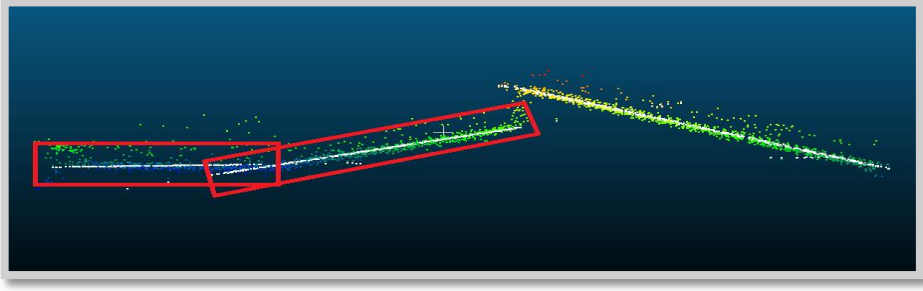
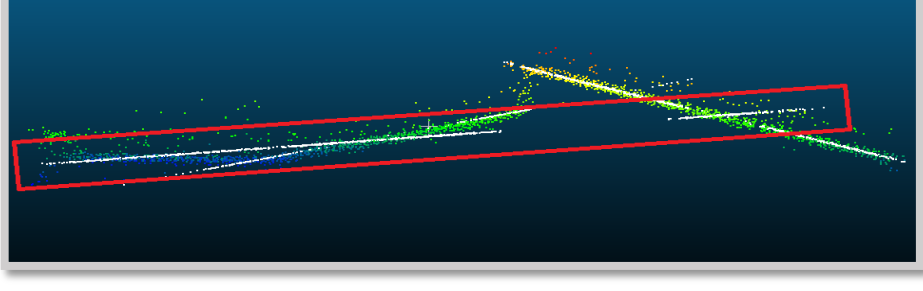
	R_C#03	
Google Earth 'Picture		
Threshold = 10 cm		
Threshold = 20 cm		
Threshold 10 cm compare with raw data		
Threshold 20 cm compare with raw data		

Figure 4.8. The comparison of the raw data and RANSAC results obtained using two different threshold values (10 cm and 20 cm) for building # 3.

RANSAC would be applied as the best approximation on the points that they have the same normal vector in the defined threshold tolerance. As for the shape of the buildings and different planes intersection with the increasing the value of threshold, the probability of applying approximation for points that in the same direction is high. The extracted roof planes of building #3 using the threshold values of 10 cm and 20 cm are shown in Figure 4.8. As seen in the figure, the threshold value of 20 cm extracts the planes incorrectly due to high approximation of the tolerance. In other words, if a high threshold values is used the number of inliers and the distribution area increase causing therefore the wrong plane cross.

The threshold value to be selected depends on several factors, including the roof materials, chimneys and the other objects that are located on the roofs as well as the edges of the buildings as shown in Figure 4.9.



Figure 4.9. Different objects on the roofs

- The second parameter is the probability of drawing the correct plane which is denoted by 't'. For this study area, the value for this parameter was selected as 0.99.[27]
- The third parameter is the probability of choosing an inlier. This parameter is represented by 'w'. The time for plane extraction from the point clouds depends on this parameter. The value for this parameter was obtained based on the trial and error method. For this study area, the value for this parameter was chosen as 0.3.

As shown in the below given Figure 4.10, due to the solar panels and chimneys on the roofs of the buildings, the threshold value of 20 cm produced better results than the threshold value of 10 cm.

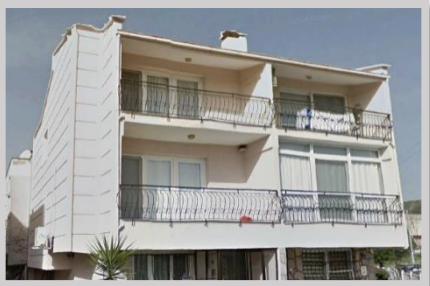
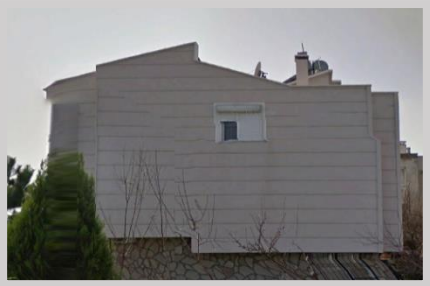
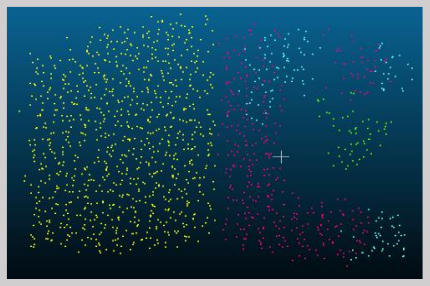
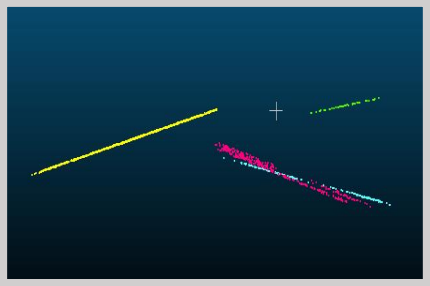
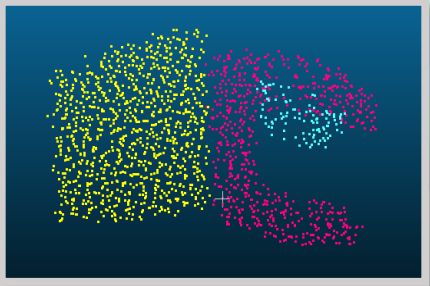
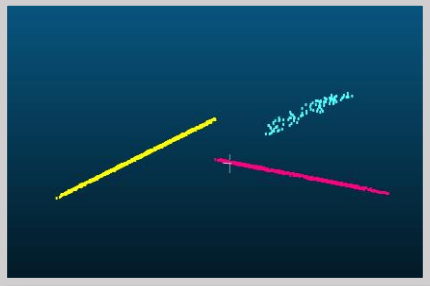
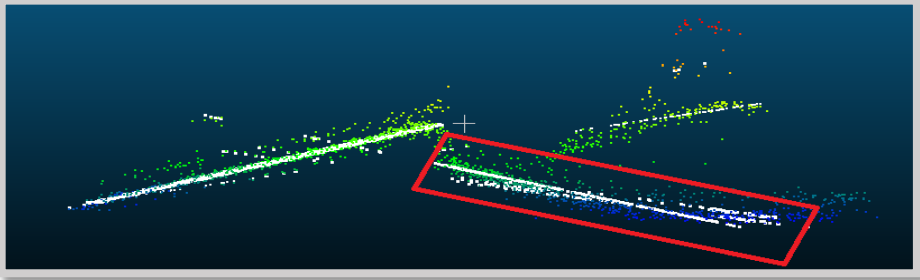
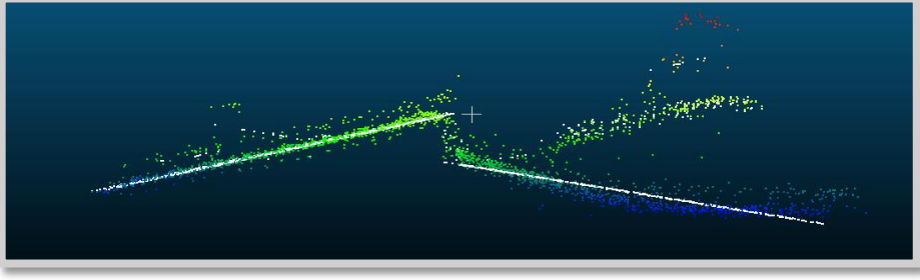
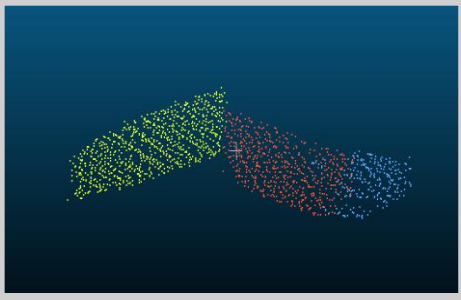
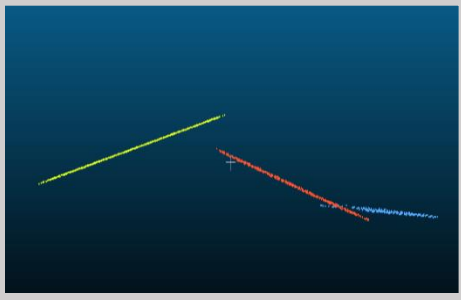
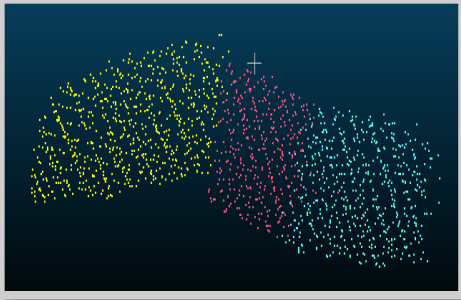
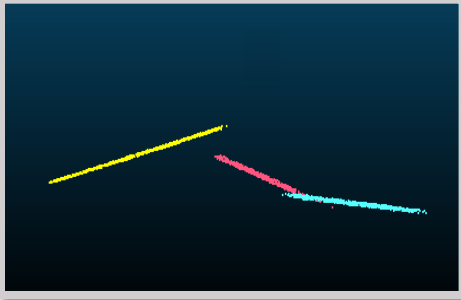
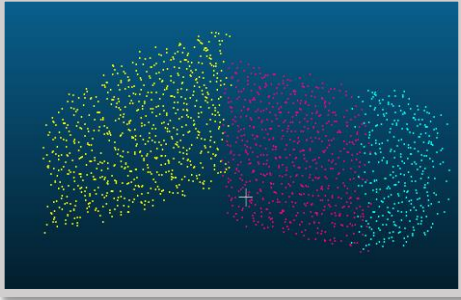
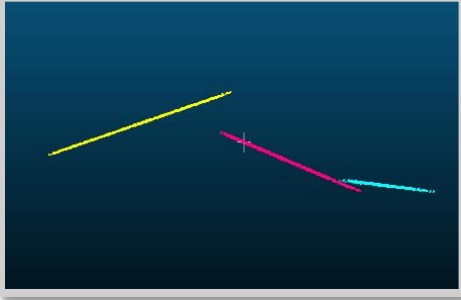
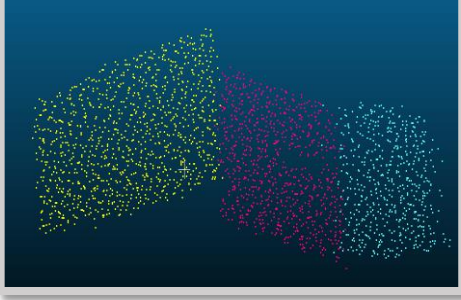


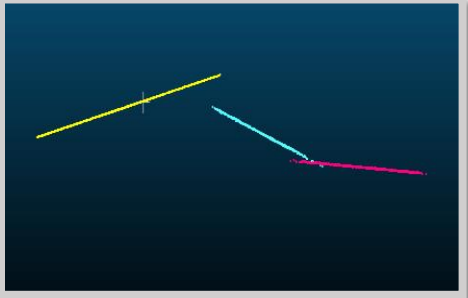
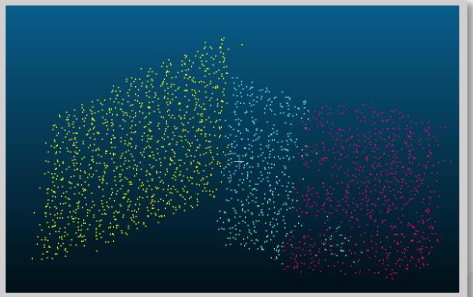
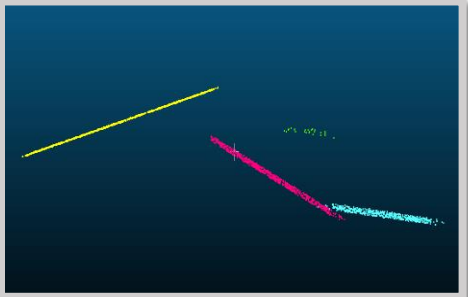
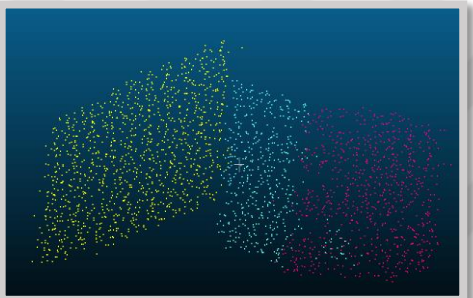
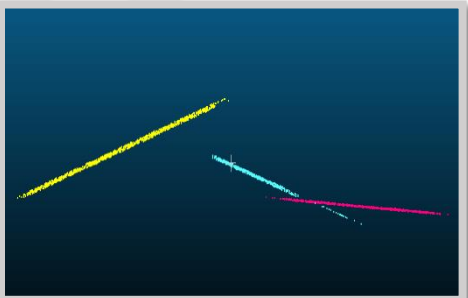
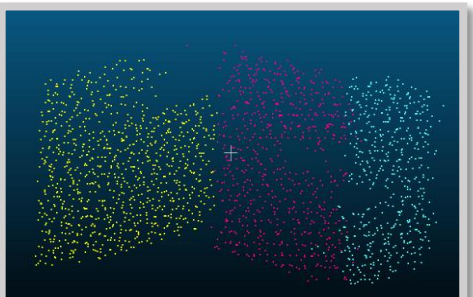
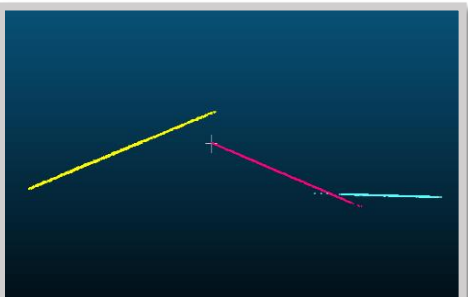
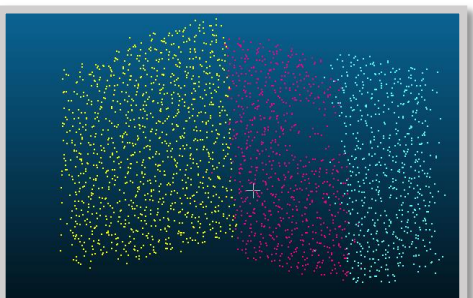

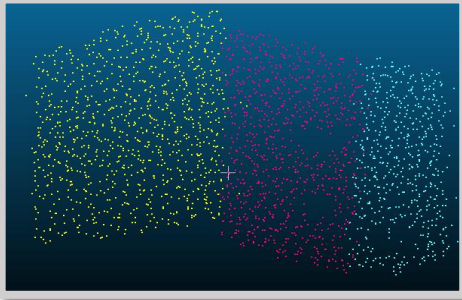
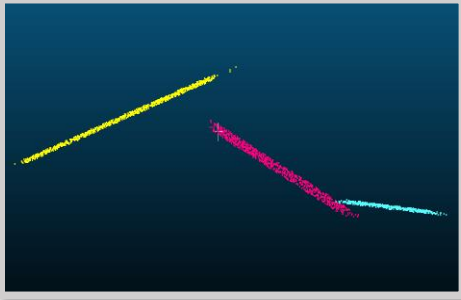
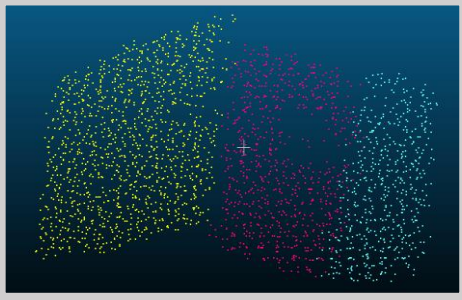
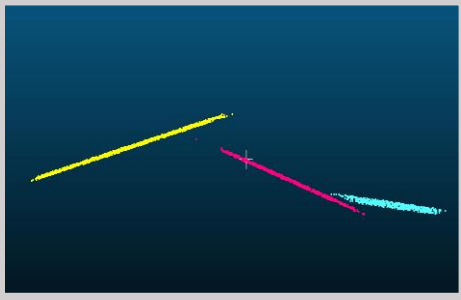
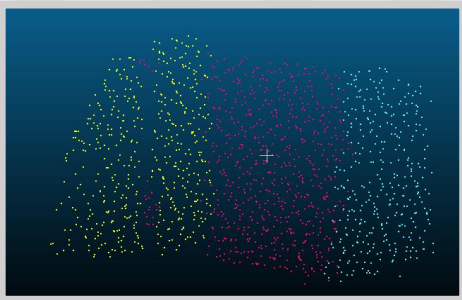
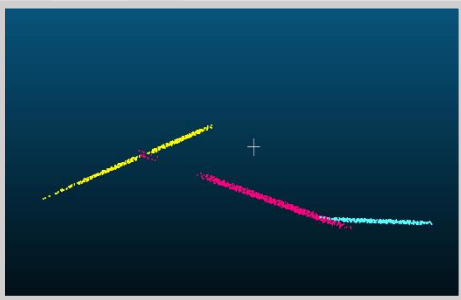
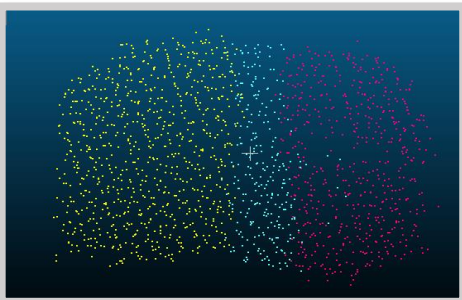
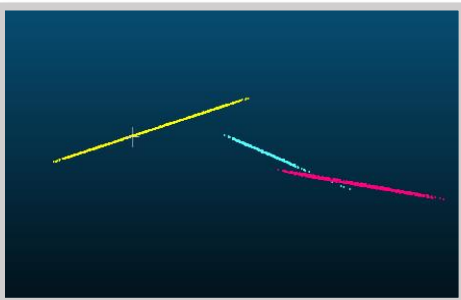
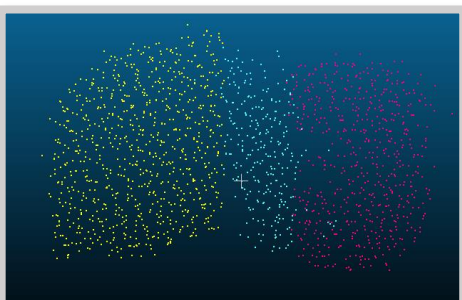

	R_C#20	
Google Earth 'Picture		
Threshold = 10 cm		
Threshold = 20 cm		
Threshold 10 cm compare with raw data		
Threshold 20 cm compare with raw data		

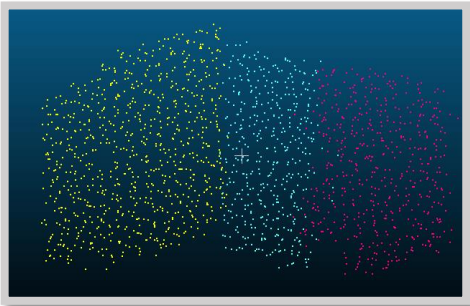
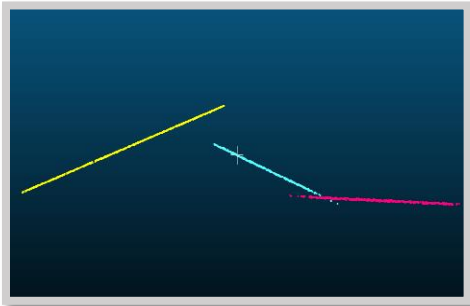
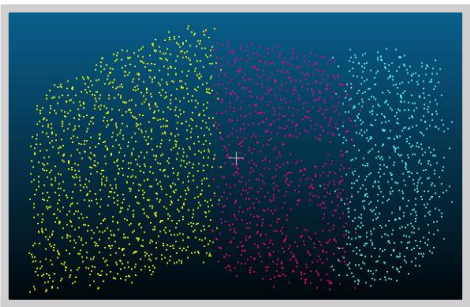
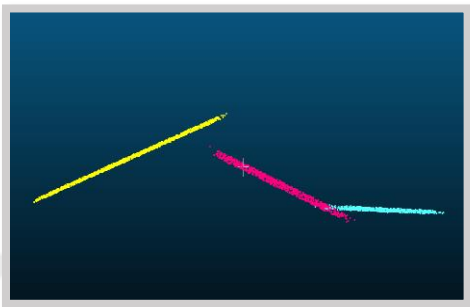
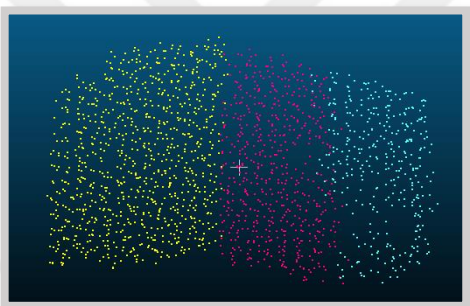
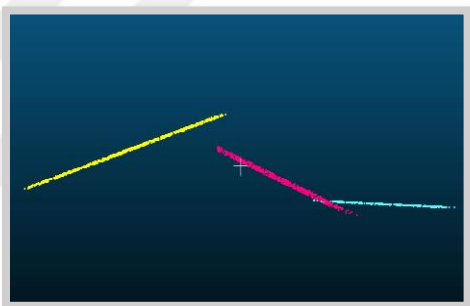
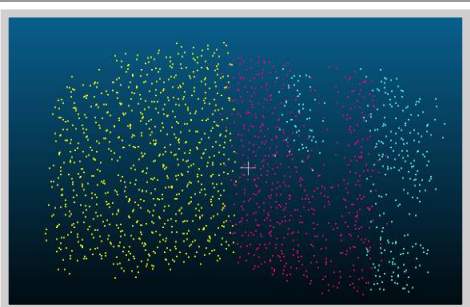
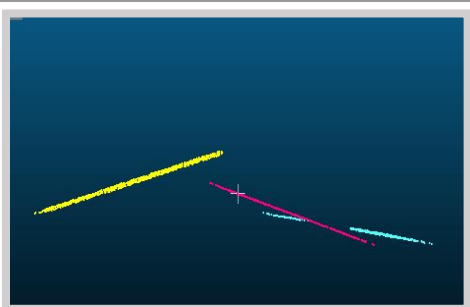
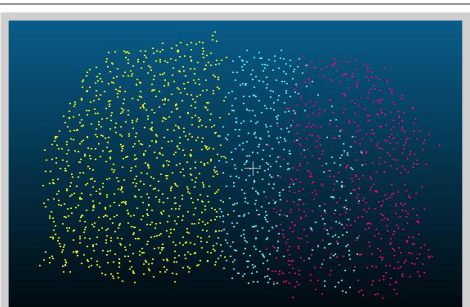
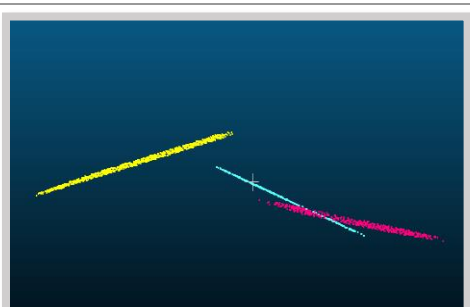
Figure 4.10. Comparison of the raw data and RANSAC results obtained using two different threshold values (10 cm and 20 cm) for building #20.

Therefore, the RANSAC algorithm was applied on the point cloud dataset of each of the separated buildings and, for each building point cloud dataset, the planes were extracted. The top and side views of the extracted planes of all buildings (#1–#23) are show in Figure 4.11.

Roof Code	Top view	Side view
R_C # 01		
R_C # 02		
R_C # 03		
R_C # 04		

R_C # 05		
R_C # 06		
R_C # 07		
R_C # 08		
R_C # 09		

R_C # 10	 A scatter plot showing a distribution of points in a 2D space. The points are colored yellow, magenta, and cyan. A white crosshair is centered in the plot.	 A line plot showing three distinct segments: a yellow line, a magenta line, and a cyan line. A white crosshair is positioned above the magenta segment.
R_C # 11	 A scatter plot showing a distribution of points in a 2D space. The points are colored yellow, magenta, and cyan. A white crosshair is centered in the plot.	 A line plot showing three distinct segments: a yellow line, a magenta line, and a cyan line. A white crosshair is positioned above the magenta segment.
R_C # 12	 A scatter plot showing a distribution of points in a 2D space. The points are colored yellow, magenta, and cyan. A white crosshair is centered in the plot.	 A line plot showing three distinct segments: a yellow line, a magenta line, and a cyan line. A white crosshair is positioned above the magenta segment.
R_C # 13	 A scatter plot showing a distribution of points in a 2D space. The points are colored yellow, magenta, and cyan. A white crosshair is centered in the plot.	 A line plot showing three distinct segments: a yellow line, a magenta line, and a cyan line. A white crosshair is positioned above the magenta segment.
R_C # 14	 A scatter plot showing a distribution of points in a 2D space. The points are colored yellow, magenta, and cyan. A white crosshair is centered in the plot.	 A line plot showing three distinct segments: a yellow line, a magenta line, and a cyan line. A white crosshair is positioned above the magenta segment.

R_C # 15		
R_C # 16		
R_C # 17		
R_C # 18		
R_C # 19		

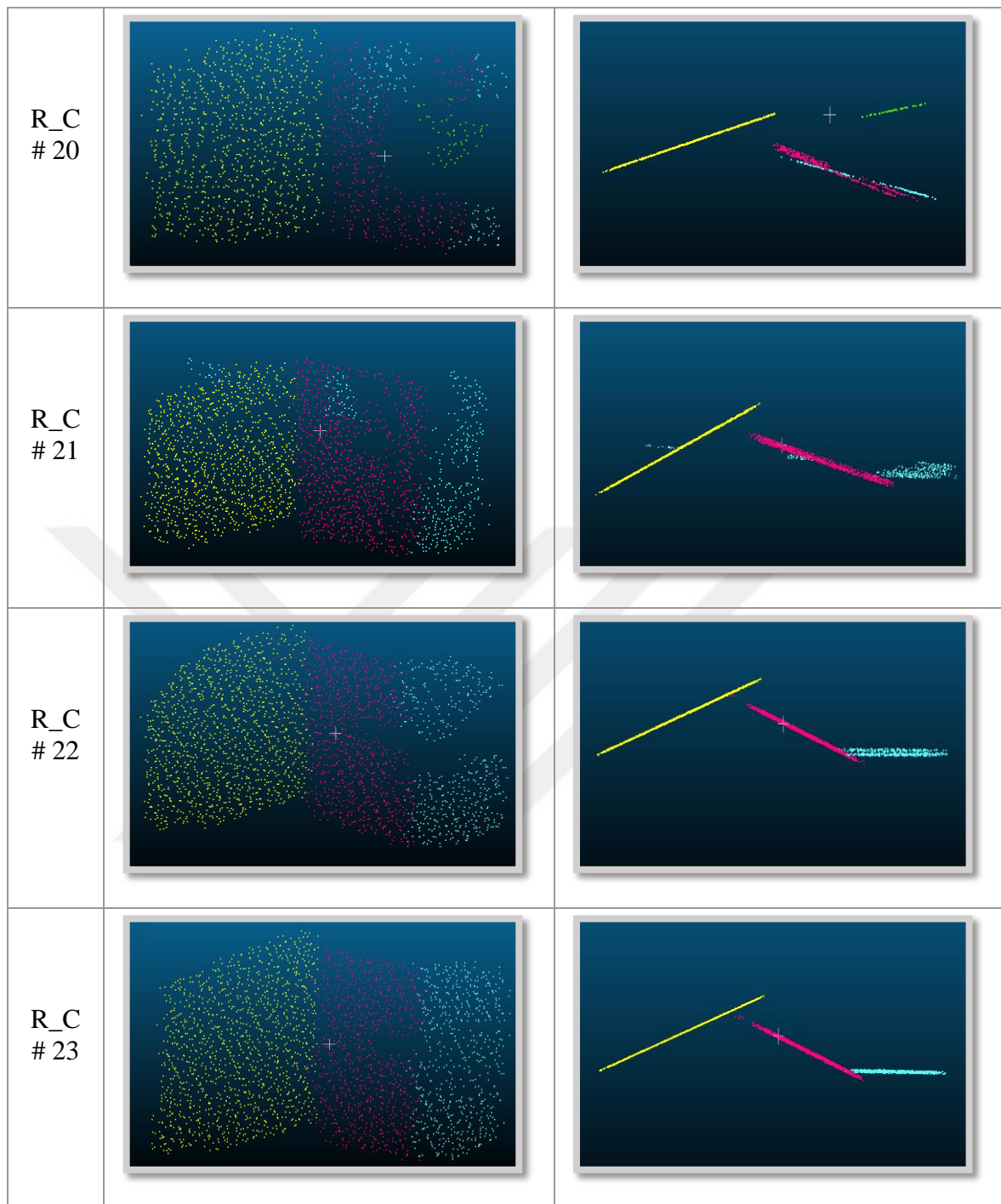


Figure 4.11. The RANSAC results.

According to the raw data and the results achieved using the RANSAC algorithm, there are several effective issues with regard to the analysis of 23 individual buildings in this study area.

A chimney is evident in Figure 4.12.A, which is associated with building #4. On the other hand, in Figure 4.12.C, the same chimney is represented as an empty region (i.e. a gap) in the results of the extracted plane using the RANSAC algorithm. The results show that depending on their dimensions the chimneys on building roofs are detected as either empty regions or a plane.

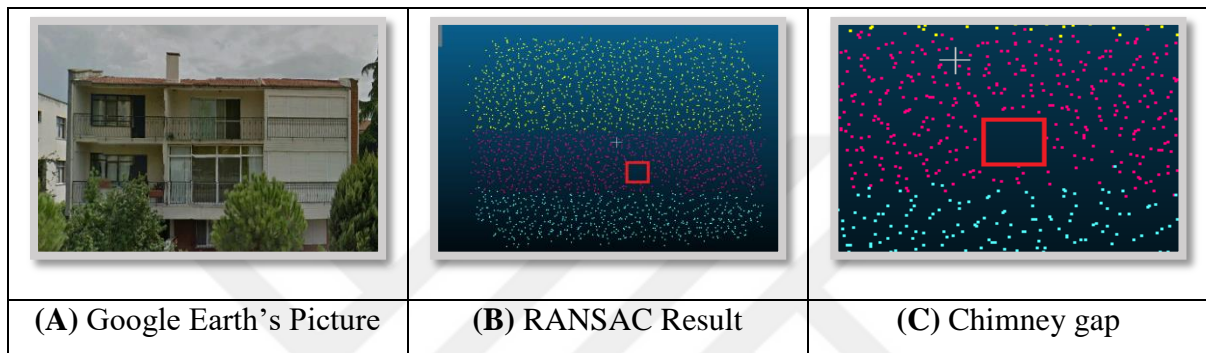


Figure 4.12. The Google Earth's picture and RANSAC results for building #4.

Similar to chimneys, solar panels also affect the results. Figure 4.13.C shows the plane of a solar panel (Figure 4.13.A and B) extracted by the RANSAC algorithm. Such objects, if located at the same level as the building edge and/or other objects that are present on the roof, may affect the RANSAC results and trick the algorithm into detecting unnecessary planes along inappropriate directions.

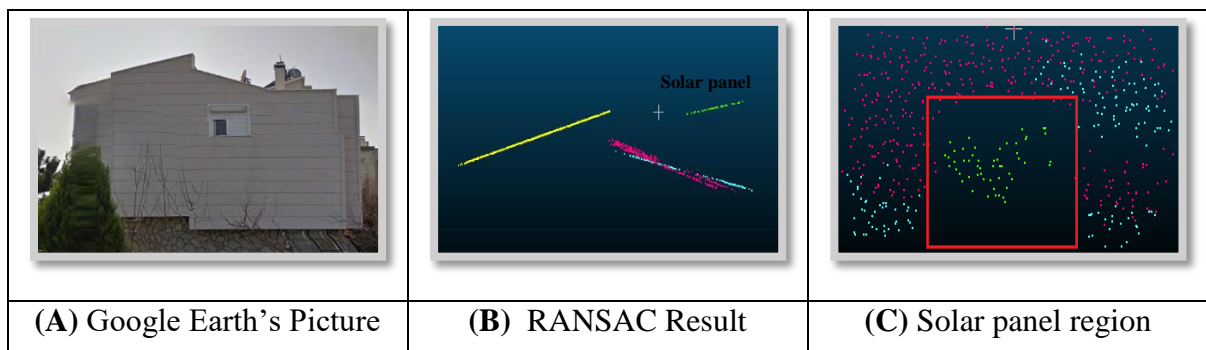


Figure 4.13. The Google Earth's picture and RANSAC results for building #20

Trees and tall plants are also among the main factors that affect the plane extraction process. Some trees are at the same height as buildings and therefore they effect the classification of non-ground points. This means that the buildings may be classified as trees and trees may be classified as buildings. This problem causes either to have an empty region or the formation of irregular shapes in the extracted planes. For building #8, the effect of trees located close to the building in roof extraction process is illustrated in Figure 4.14.

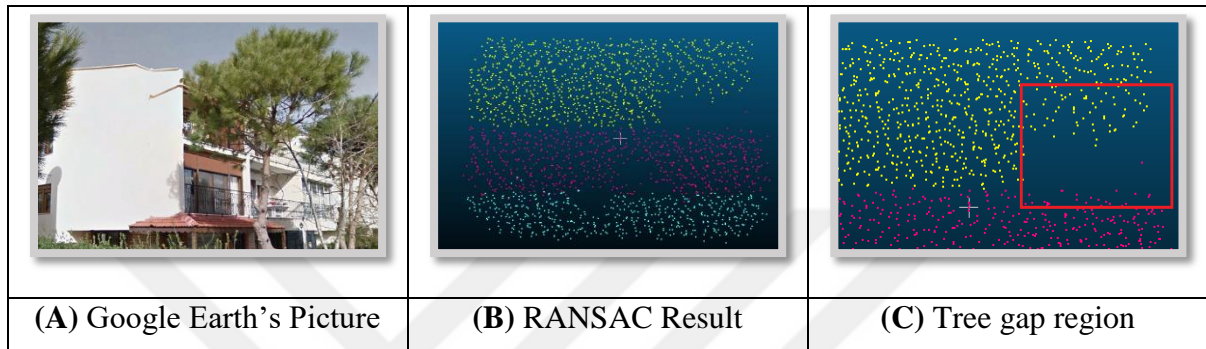


Figure 4.14. The Google Earth's picture and RANSAC results for building #8

The extracted building roofs were classified into different types. The RANSAC results obtained in this study area indicate that the buildings are not of gable type as they are composed of three planes. If the inner angle between the adjacent two planes is large then it may be difficult for the algorithms to extract these planes as separate. However, the developed script correctly extracts the planes in such conditions. As an example, the correctly extracted planes of building #23 are shown in Figure 4.15.C.

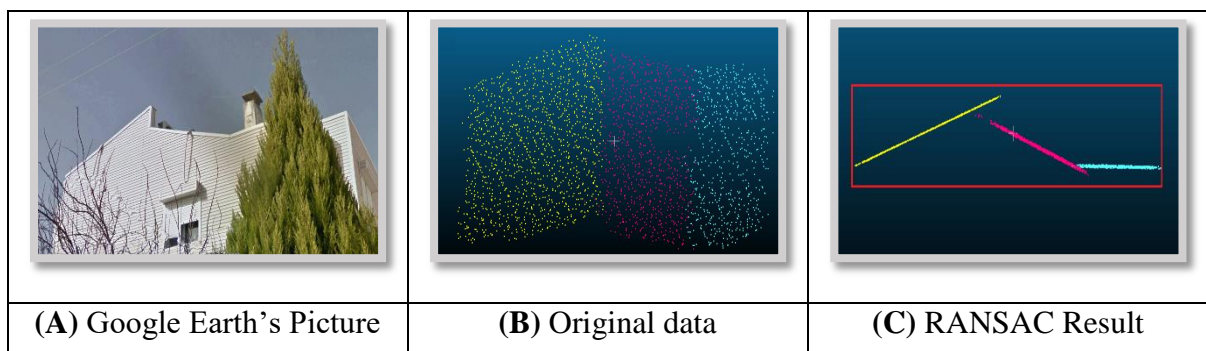


Figure 4.15. The Google Earth's picture and RANSAC results for building #23.

4.2. Second study area

4.2.1. Detection of the ground points

Similar to first study area the first step of data processing was to extract ground points from the point cloud dataset. For this process, several parameters were used. According to topography and the density of high vegetation in this study area, the ground extraction parameters of the *LasGround* function were changed. Furthermore, in the predefined parameters for ground extraction, the ‘nature’ flag was selected. The obtained result is shown in Figure 4.16.

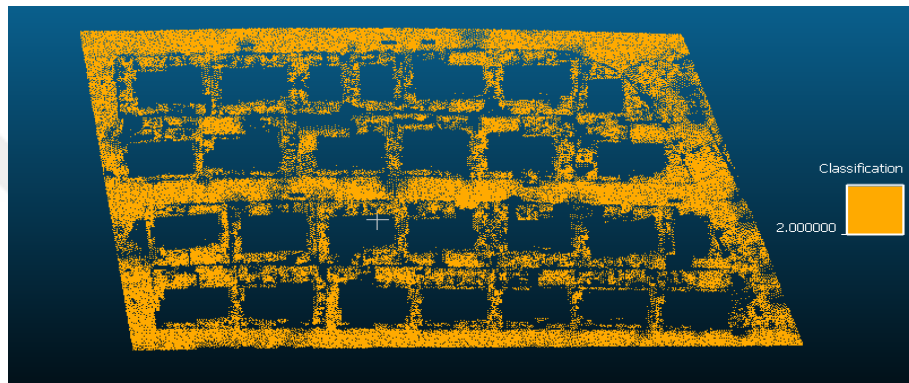


Figure 4.16. The ground points of the second study area.

4.2.2. Ground points filtering and non-ground points classifying

After detecting the ground points, they were separated from the point cloud dataset and then the non-ground points were classified into two distinct classes that are buildings and tall trees. As shown in Figure 4.17 the non-ground points, before classification, were stored as one class. Next, the classification process of the non-ground points was carried out.

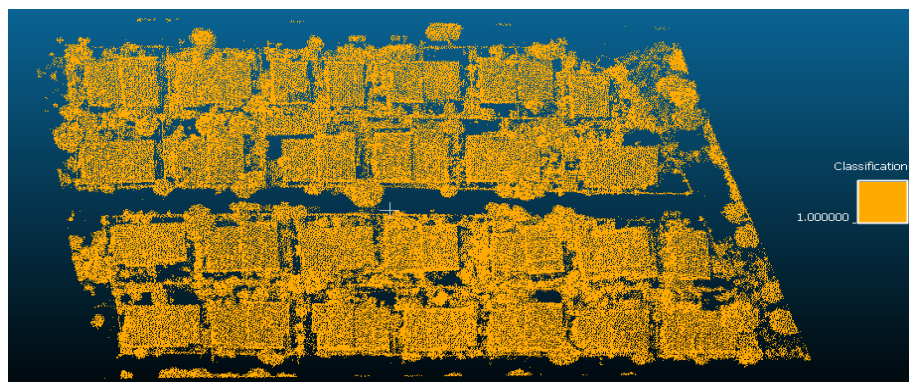


Figure 4.17. The non-ground points of the second study area.

There are four factors in classifying the non-ground points into two classes [19];

- 1) The search area size depends on the density of points. Due to the, point density in this test area (19.22 points per square meter), the dimensions of search area window was not changed;
- 2) From the geometric point of view, the point cloud is divided into planer and non - planer classes according to the value of the standard deviation and predefined threshold value. As to this classification, the buildings are in the first class and the tall trees are in the second class. In some wrong cases, the trees due to smooth cutting can be identified as building. Accurate determination of these parameters in the point cloud classification is important. In this test area the value for this parameter was selected as 0.1;
- 3) The third parameter in the classification is 'ruggedness vegetation' parameter. Tall trees and plants are identified with this parameter. There are many tall trees the building level in this test area. Therefore, to get the best results, the value of this parameter was increased to 0.8;
- 4) The buildings average height in the second test area based on the *LasHeight* calculation, is closer to the default value of this parameter. Therefore, the 'ground offset ' remained unchanged.

In this study, the classification parameter values were defined by trial and error and are shown in Table 5. The parameter values for trial 4 was found to be the best and used for the following processing operations.

Table 5. The classification parameters defined

TRIAL	1	2	3	4	5
SEARCH AREA SIZE (M)	1.5	2	2	2	2
BUILDING PLANARITY	0.1	0.3	0.1	0.1	0.1
FOREST RUGGEDNESS	0.4	0.4	0.2	0.8	0.3
GROUND OFFSET (M)	2	2	2	2	3

Next, by using the best parameter values set and the classification function, the non-ground points were classified as shown in Figure 4.18.

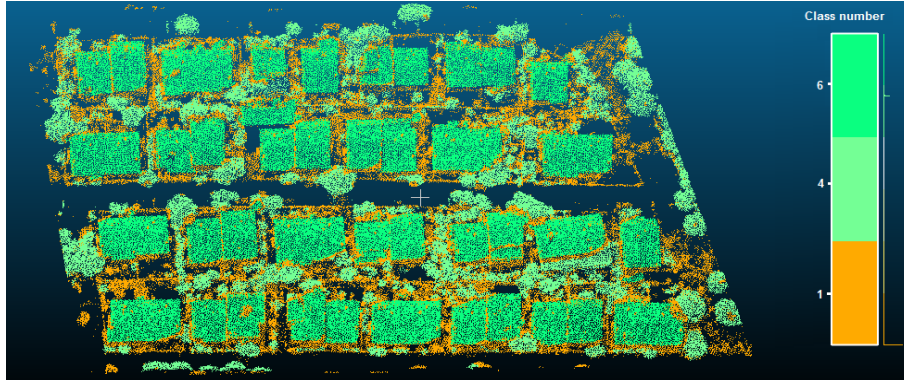


Figure 4.18. The non-ground classified points.

After classifying the non-ground points, the point cloud that belongs to building class was separated and saved as a different class as shown in Figure 4.19.

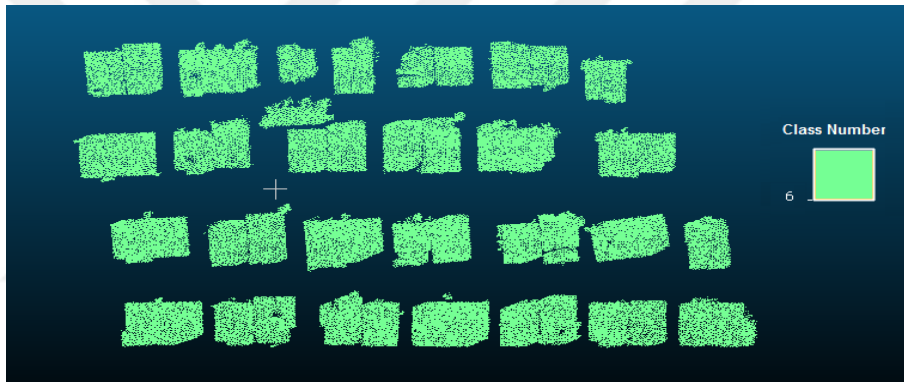


Figure 4.19. The classified building points.

4.2.3. Identifying the individual buildings

The previously extracted point cloud data set that belongs to building class was saved as one file. However, the developed RANSAC MATLAB script requires that the point cloud that belongs to building class should be processed one building at a time. Therefore, to separate the point clouds of individual buildings the ‘Label connected component’ function of the Cloud Compare software was used [25]. There are two important parameters for building separation in the Cloud Compare software.

1) The first parameter is ‘Octree Level’. The shortest distance between the sub-clouds depends on the density of the points. Similarly, the grid size and the number of sub-clouds change with the increase of the octree level. As shown in Figures 4.20 and 4.21, the value of the first level of the octree parameter and the grid size, based on the point cloud statistics, were automatically computed in the Cloud Compare software.

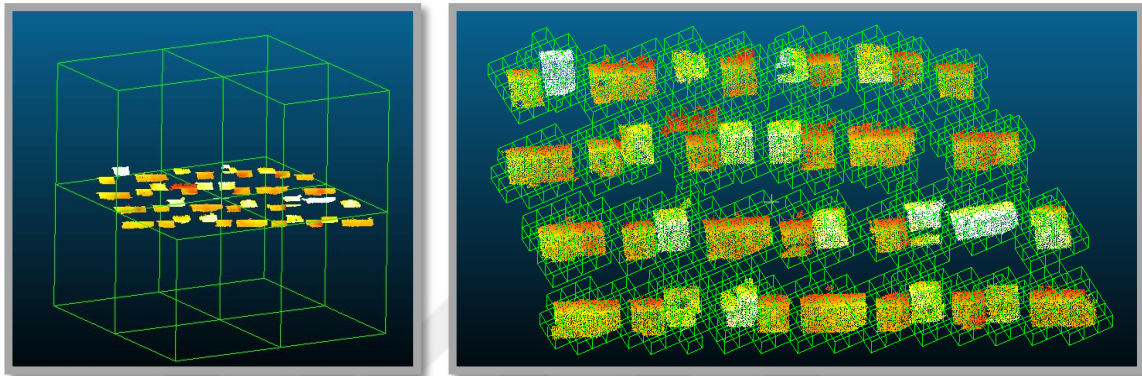


Figure 4.20. A, The first octree level (cell size=75.9458), B, 5th octree level (cell size=4.74661).

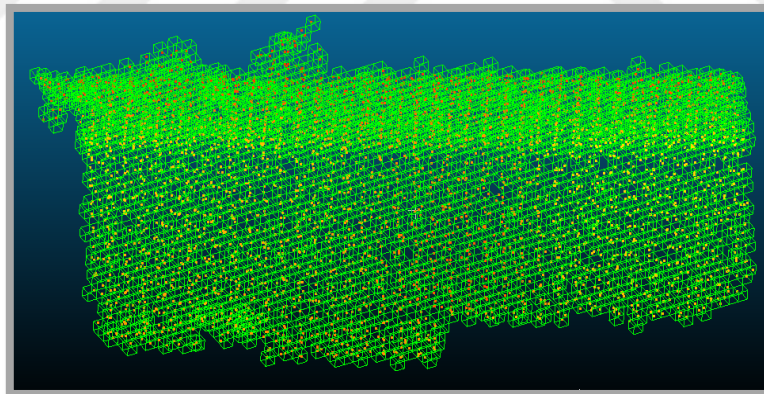


Figure 4.21. A, Second test area 9th octree level (cell size=0.296663).

2) The ‘Minimum points per component’ is the second parameter. For this study area, the octree level based on the minimum dimensions of the buildings present in this region was computed as 9. Fits it, the cell size was computed automatically in the CC software (Figure 4.22).

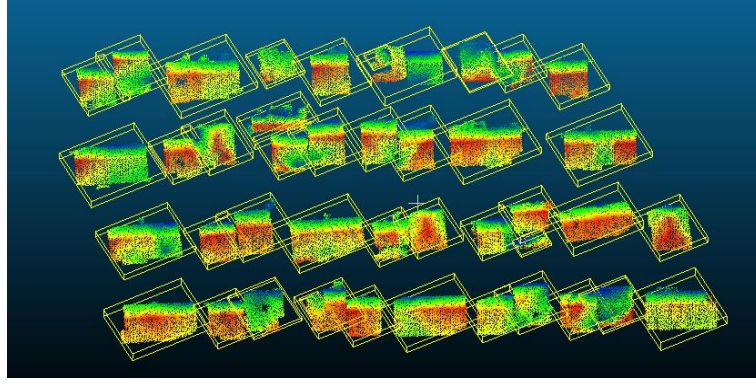
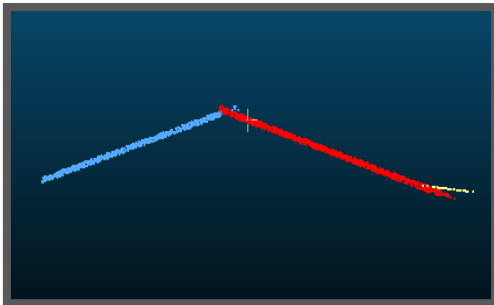
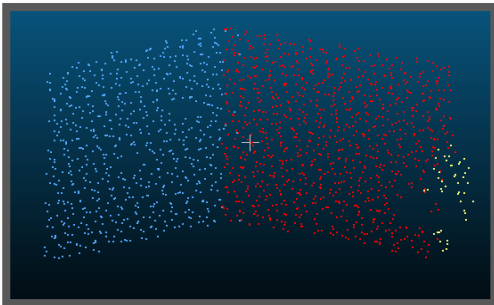
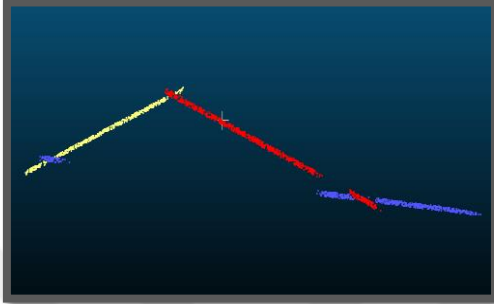
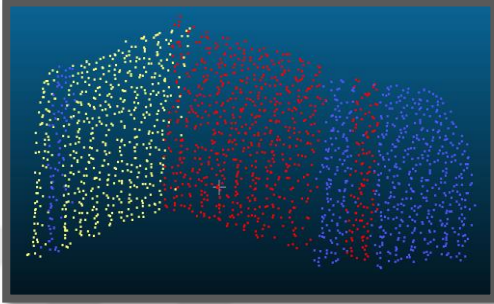
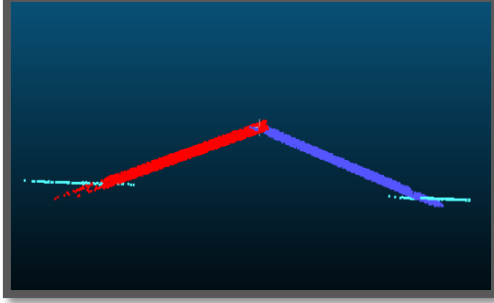
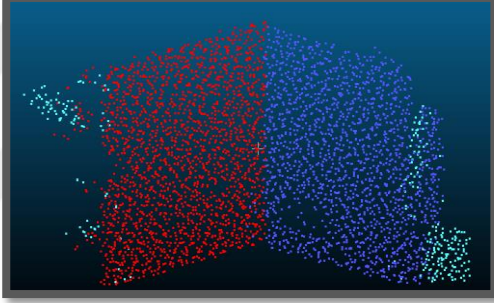
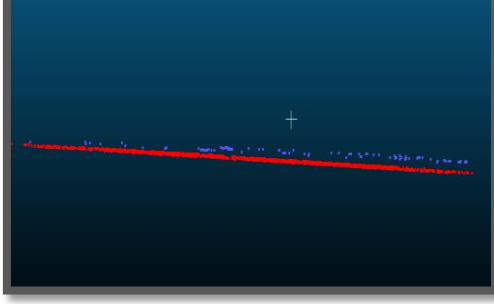
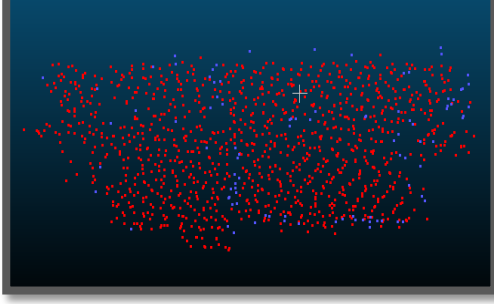

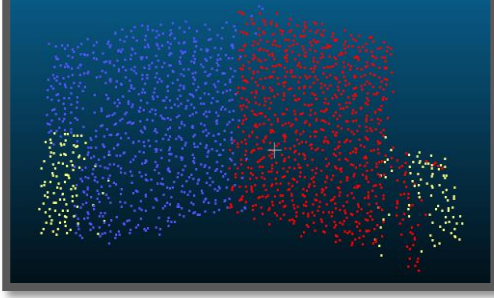


Figure 4.22. The separated buildings that were stored individually.

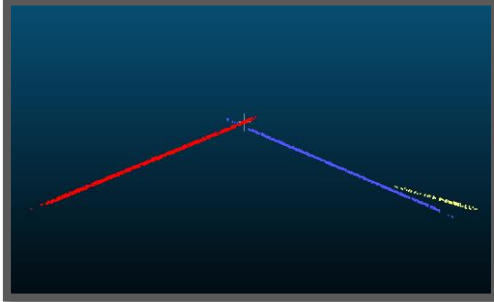
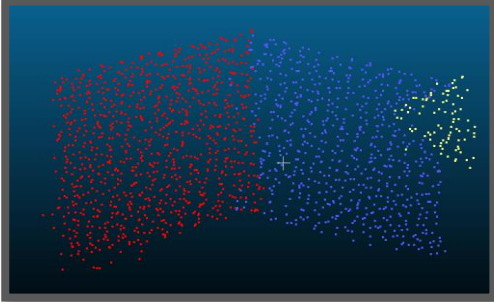
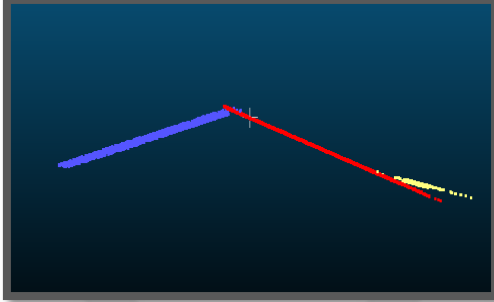
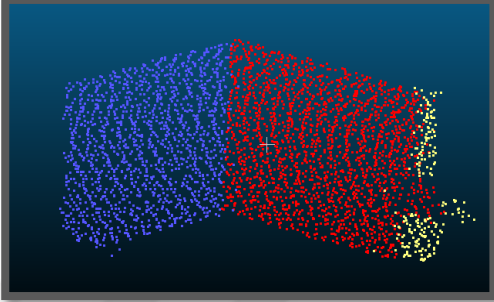
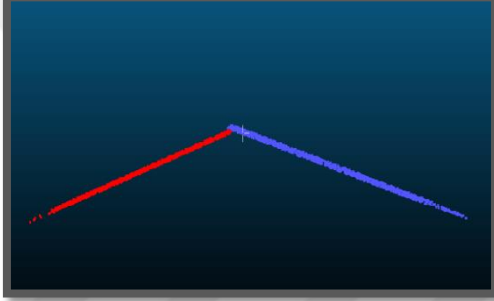
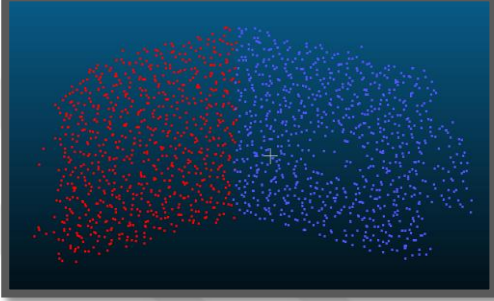
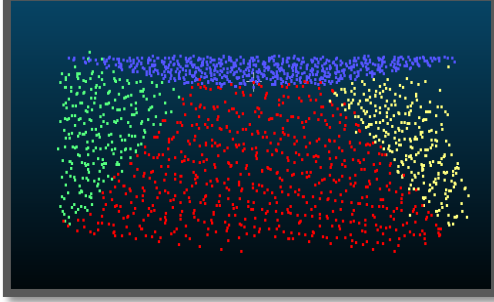
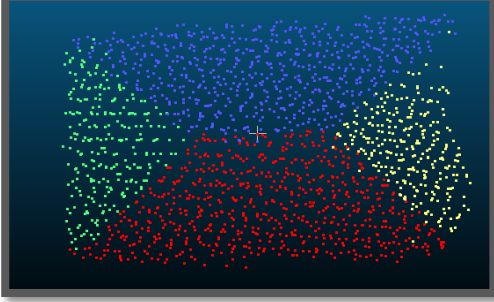
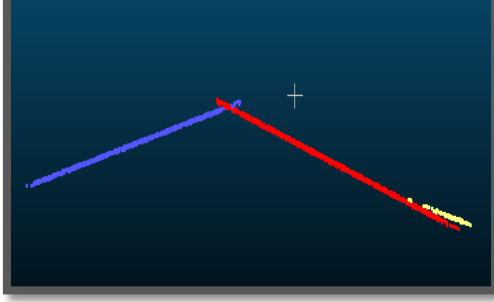
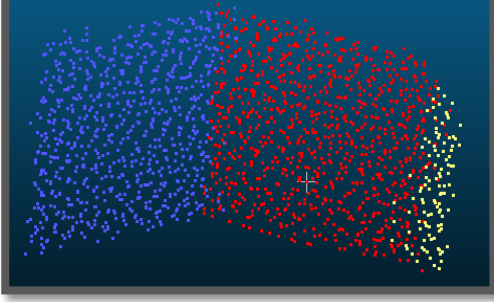
4.2.4. The extraction of the roof planes using RANSAC

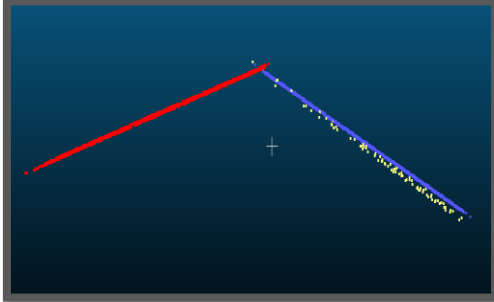
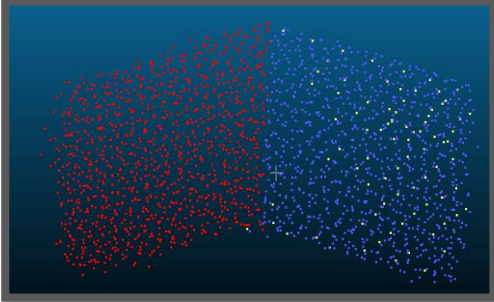
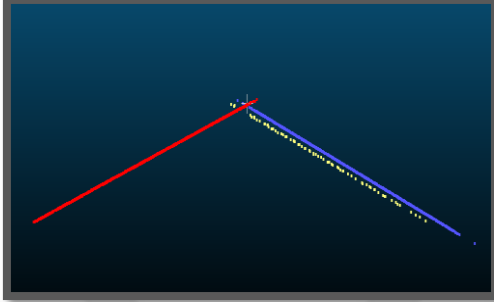
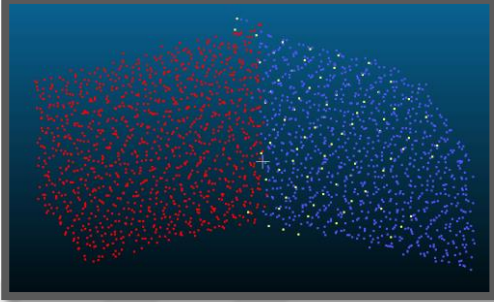
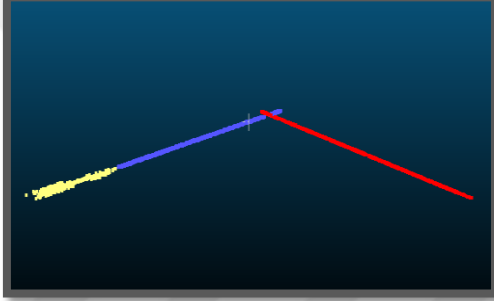
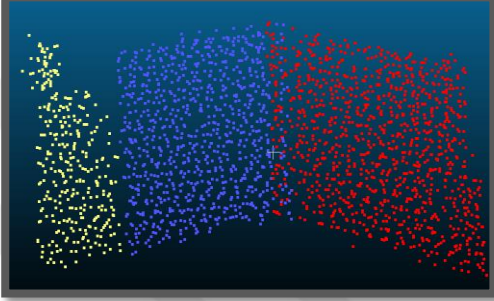
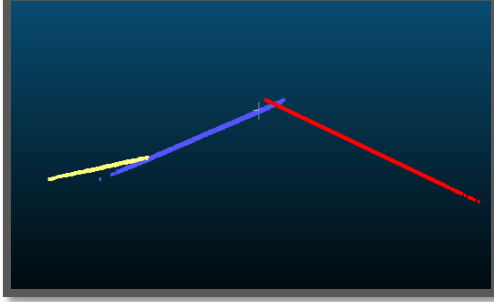
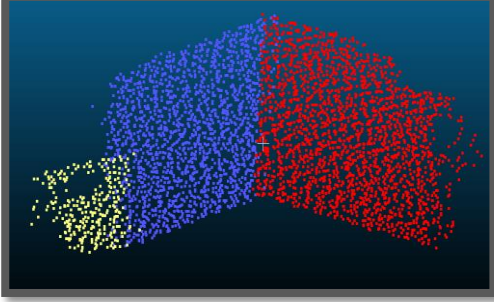
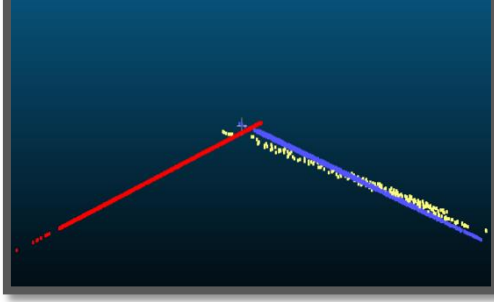
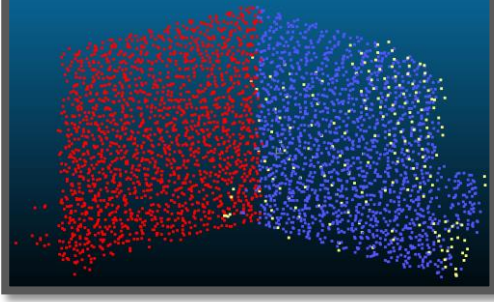
In this step, the RANSAC algorithm was used to extract roof planes of the buildings. There are three important parameters in the developed RANSAC MATLAB script. The first parameter is the Euclidian distance threshold. The point cloud data set for this study area has been collected using the same scanner as the first study area. However, there are different roof types in this study area. Therefore, this caused the value of threshold to be increased until that supported the roof roughness. For this study area, the value of distance threshold was selected as 15 centimeters. The second parameter is 't', which refers to probability of the drawing the correct plane in the RANSAC algorithms. This parameter was also used in Eq. 4. For the extraction of the number of the iterations, the default value of 0.99 was used for this parameter [27]. Finally, the third parameters is 'w', which denotes to probability of being the inlier points in the RANSAC algorithms. The value of this parameter was defined based on the trial and error method. For this study area, the value of this parameter was selected as 0.3. Furthermore, this parameter also depends on the duration of the RANSAC calculation.

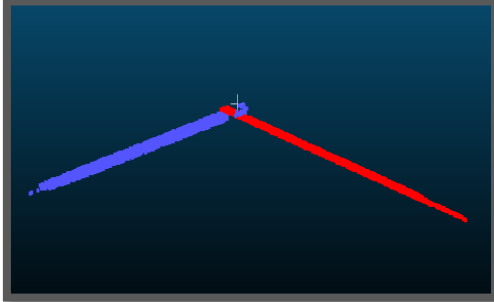
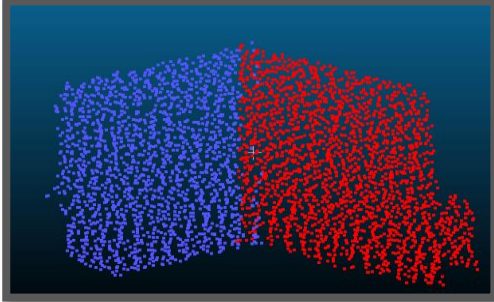
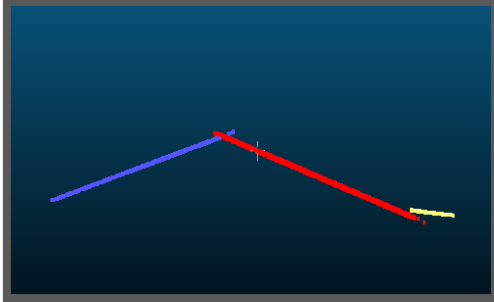
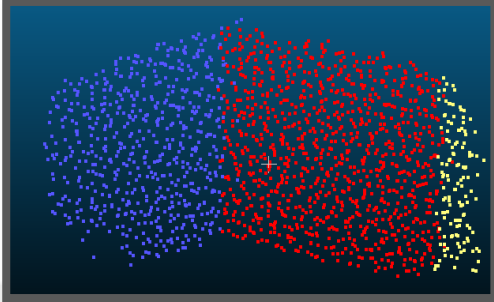
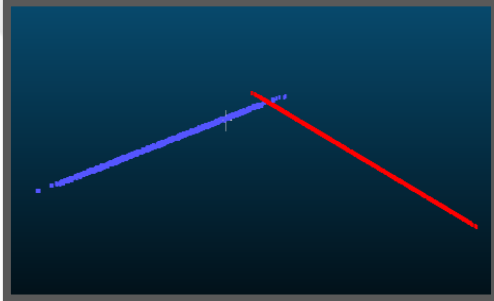
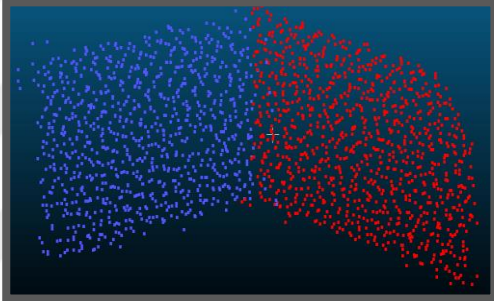
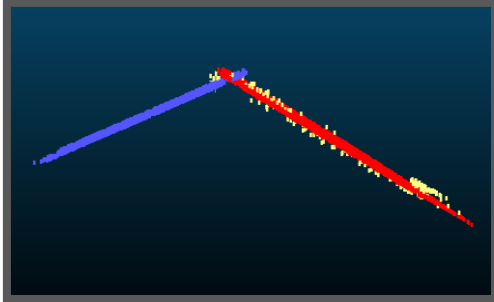
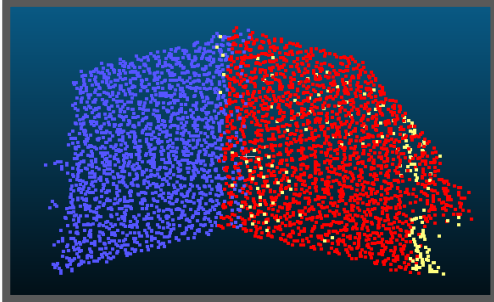
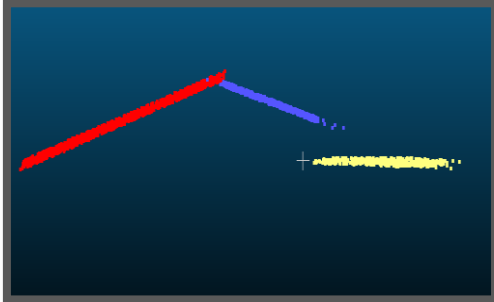
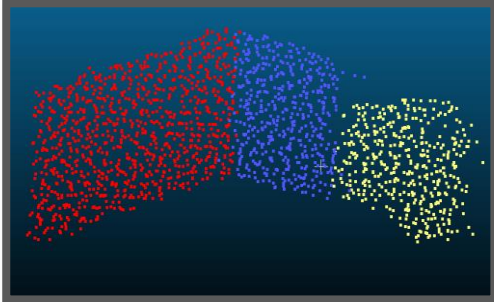
After selecting the suitable parameters, the RANSAC algorithms was applied on the point clouds of individual buildings to extract the planes. The side view and the top view of the extracted building planes are shown in Figure 4.23.

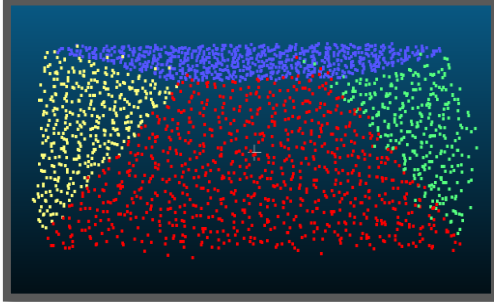
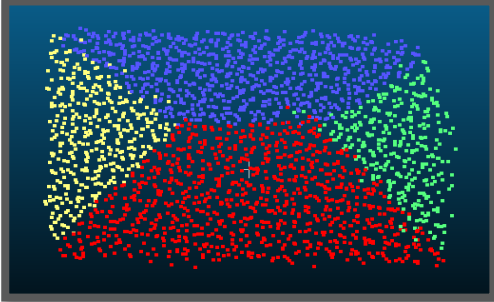
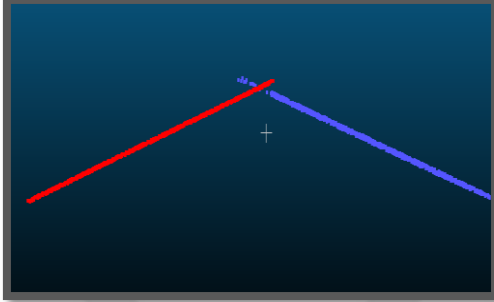
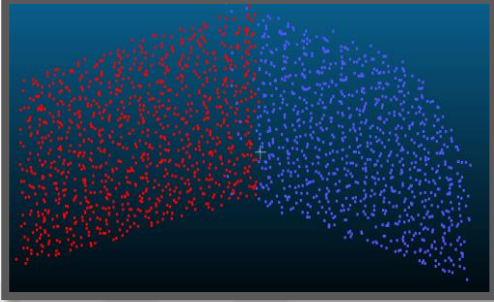
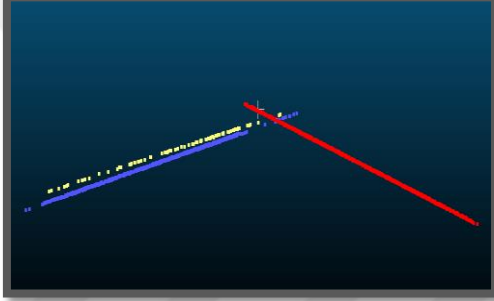
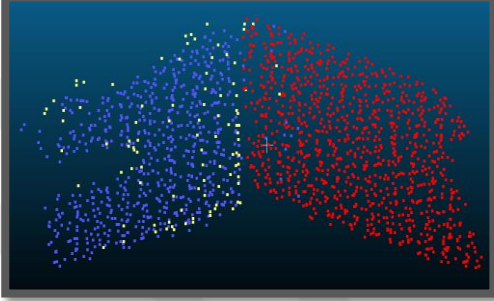
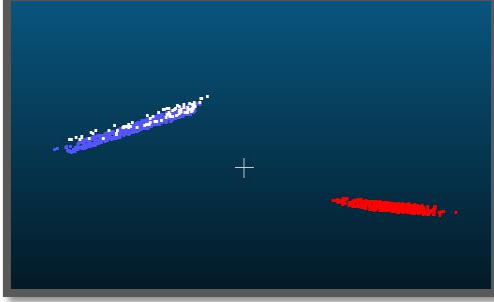
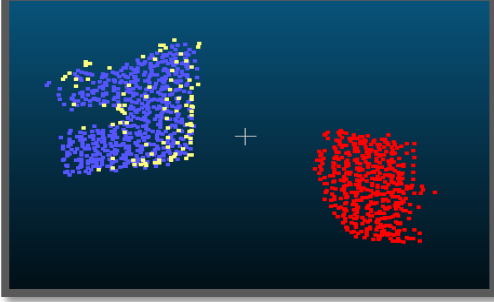
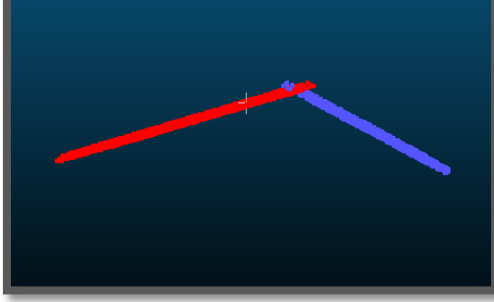
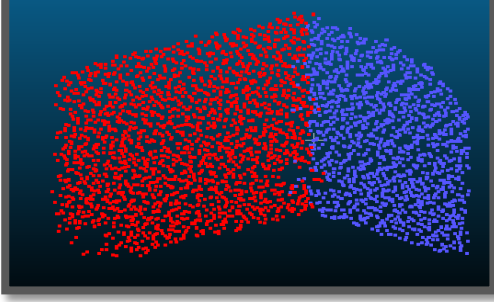
ROOF CODE	Side view	Top View
R-C #01		
R-C #02		
R-C #03		
R-C #04		
R-C #05		

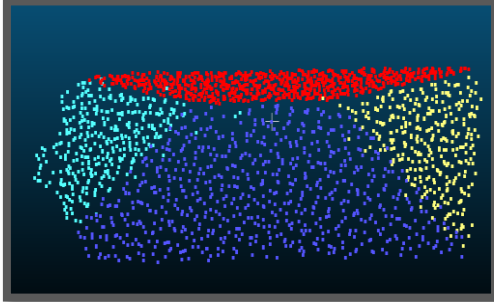
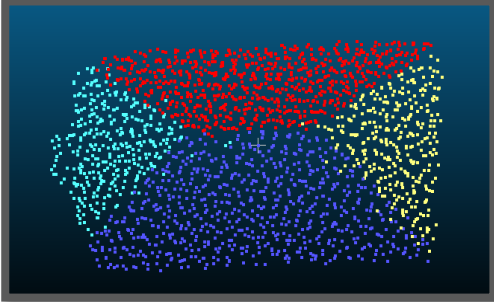
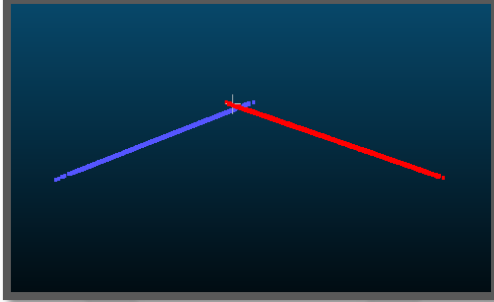
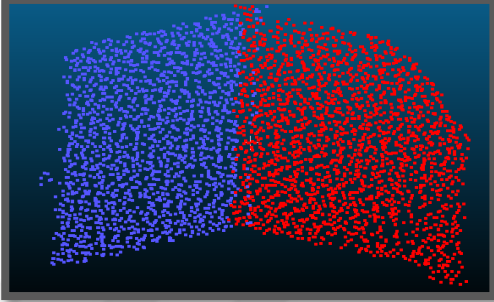
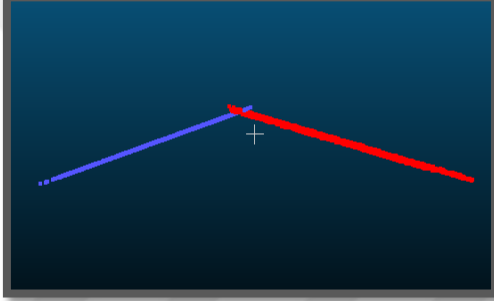
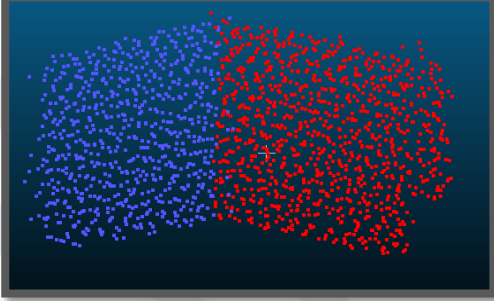
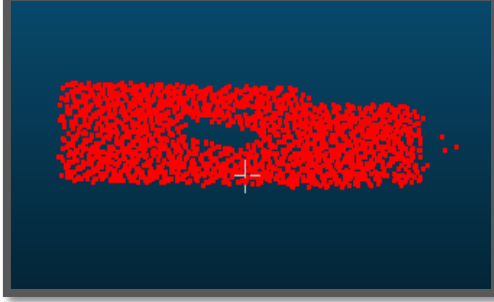
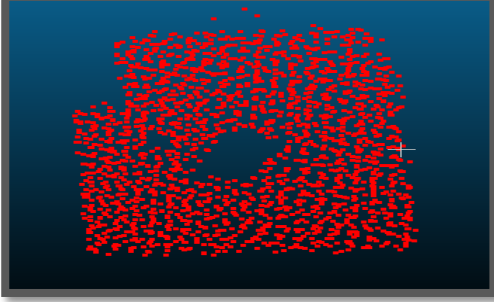
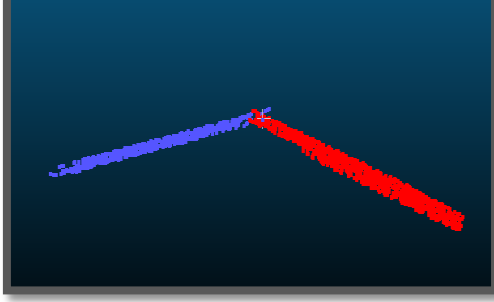
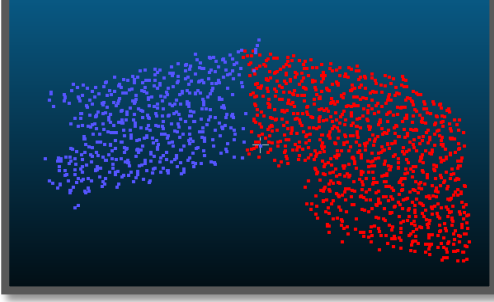
R-C #06		
R-C #07		
R-C #08		
R-C #09		
R-C #10		

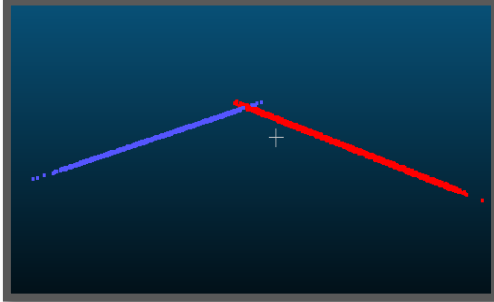
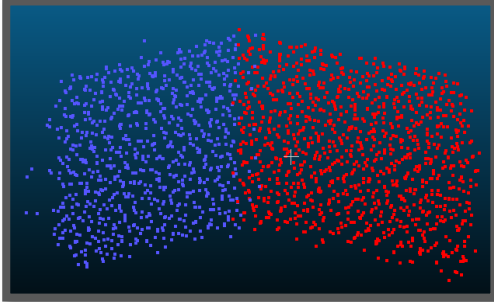
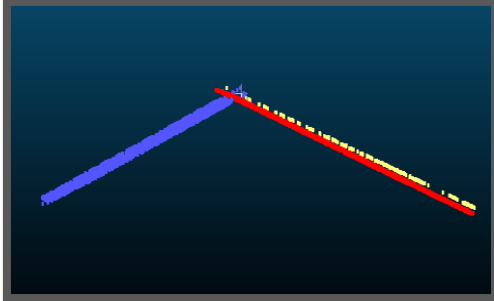
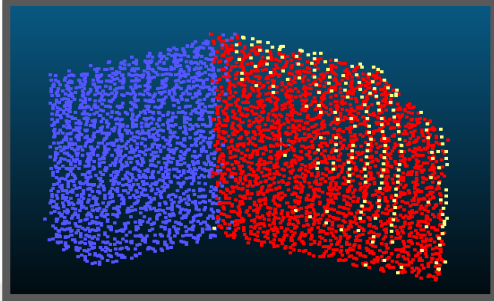
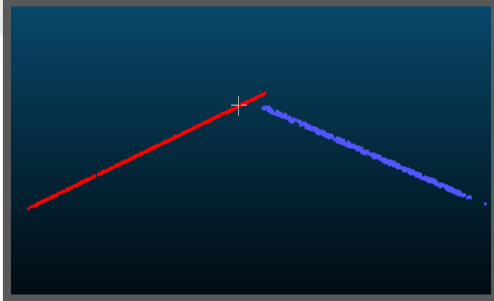
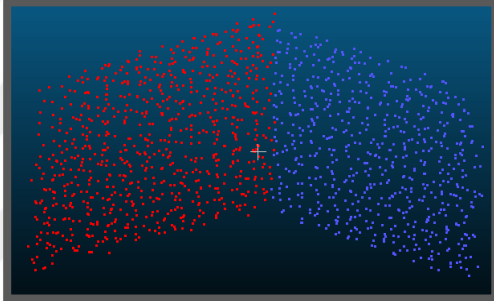
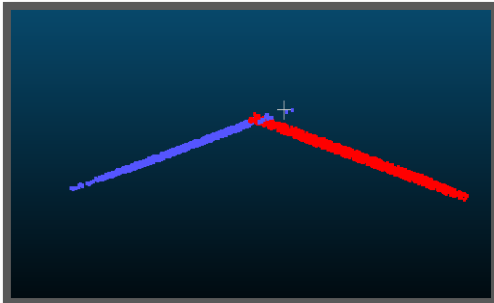
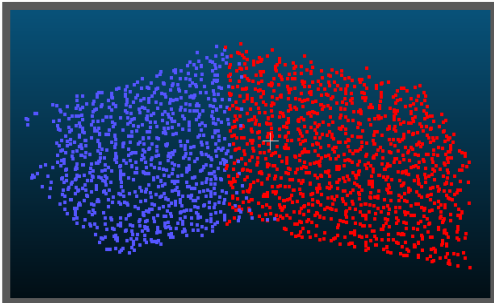
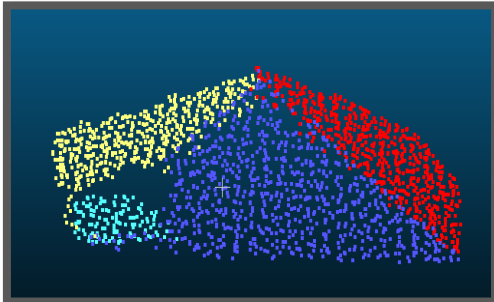
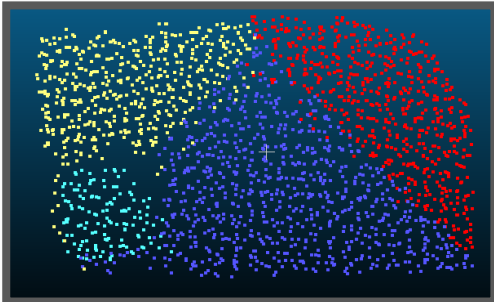
R-C #11	 A line plot on a dark blue background showing two intersecting lines. The left line is red and the right line is blue. They meet at a central point marked with a small white cross. The lines extend downwards and outwards from this point.	 A scatter plot on a dark blue background showing two distinct clusters of points. The left cluster is red and the right cluster is blue. They are separated by a vertical gap, with a small white cross marking the midpoint between them.
R-C #12	 A line plot on a dark blue background showing two intersecting lines. The left line is blue and the right line is red. They meet at a central point marked with a small white cross. The lines extend downwards and outwards from this point.	 A scatter plot on a dark blue background showing two distinct clusters of points. The left cluster is blue and the right cluster is red. They are separated by a vertical gap, with a small white cross marking the midpoint between them.
R-C #13	 A line plot on a dark blue background showing two intersecting lines. The left line is red and the right line is blue. They meet at a central point marked with a small white cross. The lines extend downwards and outwards from this point.	 A scatter plot on a dark blue background showing two distinct clusters of points. The left cluster is red and the right cluster is blue. They are separated by a vertical gap, with a small white cross marking the midpoint between them.
R-C #14	 A scatter plot on a dark blue background showing a dense, rectangular region of points. The points are colored in a gradient from green on the left to yellow on the right, with red in the center. A small white cross is located in the center of the region.	 A scatter plot on a dark blue background showing a dense, rectangular region of points. The points are colored in a gradient from green on the left to yellow on the right, with red in the center. A small white cross is located in the center of the region.
R-C #15	 A line plot on a dark blue background showing two intersecting lines. The left line is blue and the right line is red. They meet at a central point marked with a small white cross. The lines extend downwards and outwards from this point. A small white cross is also visible above the intersection point.	 A scatter plot on a dark blue background showing two distinct clusters of points. The left cluster is blue and the right cluster is red. They are separated by a vertical gap, with a small white cross marking the midpoint between them.

R-C #16		
R-C #17		
R-C #18		
R-C #19		
R-C #20		

R-C #21		
R-C #22		
R-C #23		
R-C #24		
R-C #25		

R-C #26		
R-C #27		
R-C #28		
R-C #29		
R-C #30		

R-C #31		
R-C #32		
R-C #33		
R-C #34		
R-C #35		

R-C #36		
R-C #37		
R-C #38		
R-C #39		
R-C #40		

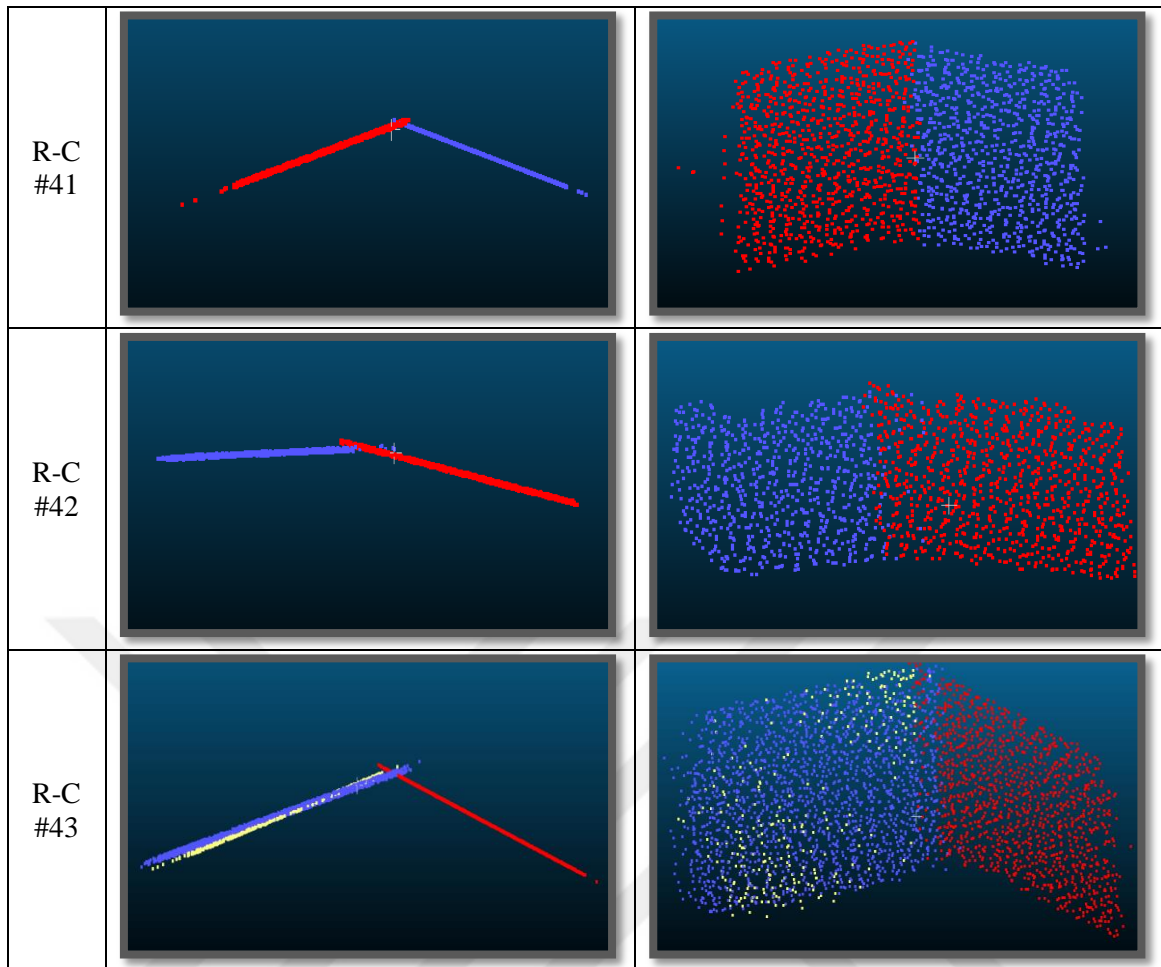


Figure 4.23. The side view and top view of the building planes extracted through the RANSAC algorithm.

Several important factors affect the results achieved for this study area.

- An example of a roof of a building with the chimney is shown in Figure 4.24. As can be seen in the figure, the chimney was also extracted. Therefore, objects such as the chimney on the roofs generate gaps on the planes.



Figure 4.24. The illustration of the planes extracted for Building #1.

- In some cases, the RANSAC algorithm extracts a plane parallel to the main roof plane due to roof materials used in the buildings as shown in Figure 4.25. This problems usually occur for the galvanized roofs with a sinusoidal model. Since some of the buildings in the Bergama region have galvanized roofs with sinusoidal patterns of long amplitudes, in some of the results, an extra plane parallel to the main plane was extracted. On the other hand, since only a limited number of buildings are of this type and a single threshold should be applied to all buildings, the threshold was not significantly modified to match the few aforementioned buildings.



Figure 4.25. A, building (Building #10) with the extracted parallel planes (the plane of white data points and the plane of red data points).

- Similar to first study area, the building roofs in this study region also include solar panels that affect the results (Fig. 4.10). As can be seen in Figure 4.26, the gap inside the plane of yellow color data points shows that the solar panel on the roof was not detected as a plane.

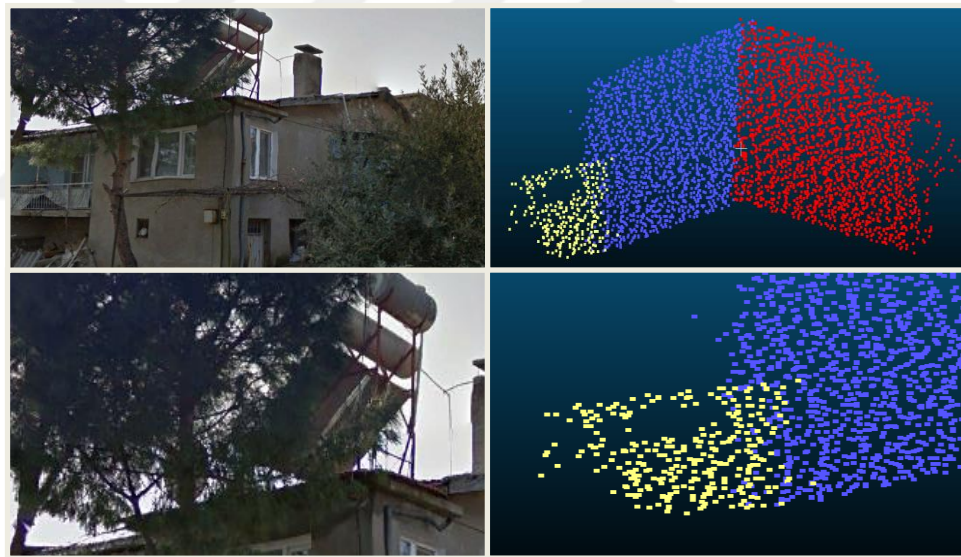


Figure 4.26. The planes extracted from Building #19. The gap within inside the plane of yellow color data points show the location of a solar panel which was not extracted as a roof plane.

- In this study area, the building roofs are very diverse and have up to 4 roof planes, as shown in Figure 4.27. B. Moreover, some building roofs contain dorms, receivers etc. If the sizes of the planes of these objects are smaller than the predefined threshold value then, these planes are not extracted and therefore gaps occur in the point cloud of the corresponding positions. As shown in Figure 4.27. A, the planes of the dorm of building #28 were not extracted causing a gap in RANSAC results.

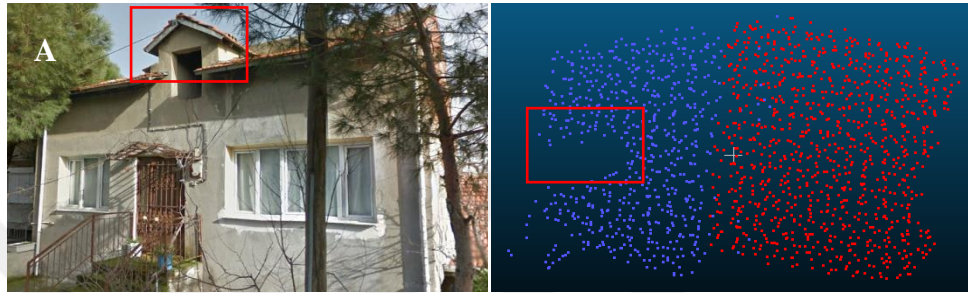


Figure 4.27. A. the building #28 with small window.

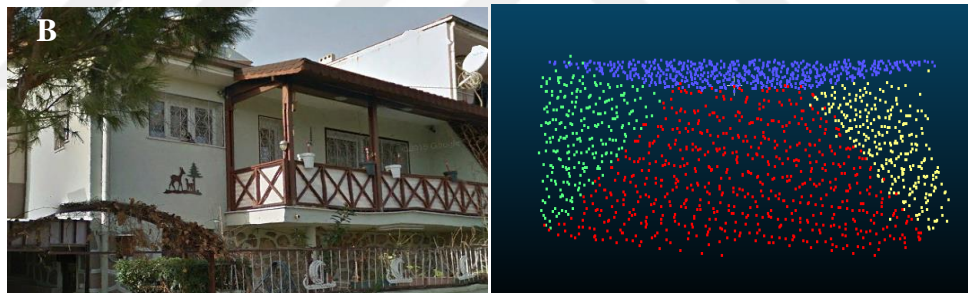


Figure 4.27. B, the building # 14 with four facets type.

- Dense vegetation and tall trees that exist in this study area were taken into consideration when performing the classification of the non-ground points. In this study area, the buildings are located on slopes, enabling tall trees to cover the roofs of some of the buildings. As it was mentioned previously in the first test area, the overlapping of trees and building roofs leads to incorrect classification, which in turn causes problems in the extraction of the planes. Figure 4.28 shows examples for the overlapping of trees and building roofs, where the plane associated with the building roof is incorrectly and irregularly extracted.

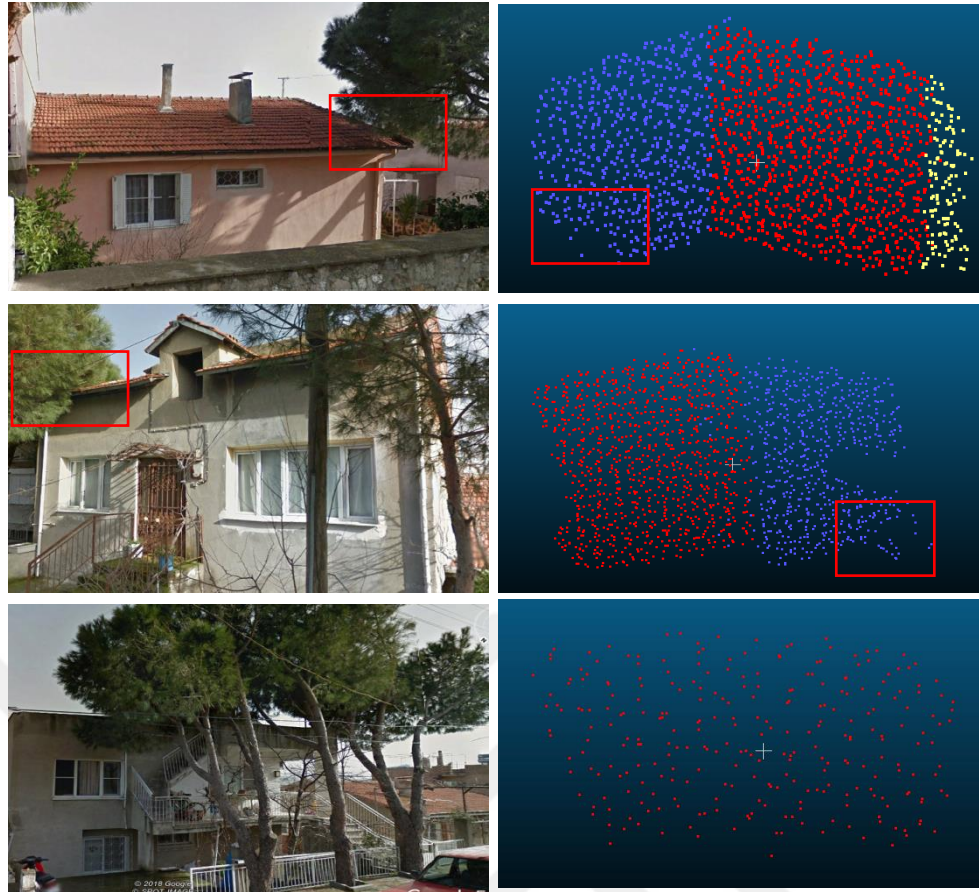


Figure 4.28. Buildings #22, #28, and #7 with the neighboring tall trees that partially occlude the roofs.

5. Conclusions and Recommendations

5.1. Conclusions

In this study, building roof planes were automatically extracted from airborne LiDAR point cloud data using the RANSAC algorithm, which is known to be one of the most powerful plane extraction algorithms available. To do that a script was developed in the MATLAB programming environment. As the inputs, the algorithm receives a distance threshold between the points, the number of iterations, and dimensions of the smallest plane to be extracted from the point cloud being processed. The raw LiDAR data is initially pre-processed. The pre-processing includes ground filtering, classify the non-ground points and separate the building class to individual data sets. The final plane is obtained by applying the least square estimation method on candidate points that form a plane in the RANSAC algorithm. For the experiments, two study sites with different characteristics were selected from the city of Bergama. The LiDAR point cloud data sets used were collected through the Riegl scanner. The results achieved for both study areas are quite promising. Most of the roof planes were correctly extracted using the proposed method. The conclusions reached during this research study are as follows:

The conclusions reached from the processings of the point cloud data sets in the first study area are as follows:

- It was concluded that the plane extraction process is affected by the large-sized diverse objects such as solar panels, chimney, and satellite dishes that are present on rooftops. For instance, a solar panel can be identified as a plane if located at the same level as one of the planes on the roof. Moreover, the panel can be extracted as a void if it has a smaller size.
- The number of extracted planes is dependent on the distance threshold between the planes and the points. In this study, two thresholds values of 10 cm and 20 cm were tested. When a larger threshold value is used, the intersection angle between two planes becomes large and therefore, two planes are wrongly extracted as one single plane. It should be noted that a single threshold value of 10 cm was used in this study for the processing of point cloud data sets for all buildings. However, it was found that using a single threshold value for all buildings does not always produce satisfactory results. For example, in the case of the processing of the point cloud

data for Building #20 in the first study area, the use of 20-cm threshold value produced better results than that of the results achieved using the 10-cm threshold value. This was due to the type of the roof material used for Building #20.

- For the gable roofs, if a large distance threshold between the plane and data points is selected then, at the intersection of the neighboring planes, the plane identified for a roof surface may incorrectly include points from the intersecting other plane. Therefore, this problem can be solved by properly selecting the distance threshold limit.

The conclusions reached from the processings of the point cloud data sets in the second study area are as follows:

- It was found that due to roof materials of some of the buildings a false plane was extracted parallel to the main one. However, in the study areas used the number of buildings with such rooftops were limited, and therefore the use of single threshold value of 10 cm for all buildings did not quite affect the results in the present case.
- It was found that the results are affected by tall trees that partially occlude the roofs of the buildings that are in particular located on slopes. It was observed that parts of the tall trees neighboring to buildings were identified as the extension of the roof planes or as voids on the roofs. However, this problem can be somehow solved by excluding all points that belong to trees by means of an appropriate classification of the non-ground points. In the plane extraction stage, a parameter value regarding the minimum number of points that a plane must contain is needed to be defined to avoid the extraction of small planes. It should be noted that this parameter value must be defined based on the density of the point cloud dataset as the point density affects the results.
- It was found that those buildings that share a single roof generate a single plane and this matter affects the results. The separation of the roof planes in such conditions can be done either using additional data sets, such as high resolution imagery or by visiting the site.
- The developed script in the MATLAB programming environment for the extraction of the roof planes from a point cloud dataset that belongs to a building is quite efficient. For example, it takes about 161 seconds for the extraction of the planes from the point cloud data set of building #01 which consists of 1994 points. The

processing times compued for several buildings using a computer with the specifications “Intel(R) Xenon(R) CPU E5-1620 V2 3.70 GHz, RAM: 32 GB” are given in Table 6.

Table 6. The processing time for RANSAC results.

Building	Study area	Number of planes	Distance treshold (cm)	Processing time (s)
#01	1	1994	10	161.3005
#16	1	3435	10	322.312
#38	2	5051	10	354.653
#30	2	4063	15	293.218
#02	2	2162	15	212.466
#17	1	2391	15	284.308

- By assessing the extracted results and visually comparing them with both theoriginal point cloud data and the Google Earth imagery it can be concluded that, to a great extent, a close aggreement is evident between the extracted planes and the original roof planes. Thus, this indicates that the approach used in this study is quite robust in the extraction of building roof planes from airborne LiDAR data. For buildings #1 and #8, the comparison of the raw data and the extracted planes are shown in Figure 5.1.The information about the extracted planes such as the area, perimeter, normal of the plane, etc would be quite useful for the generatiion of 3D building models.
- Based on the results achieved in this study, the extracted planes is useful for setting roof solar system on rooftop of the building.

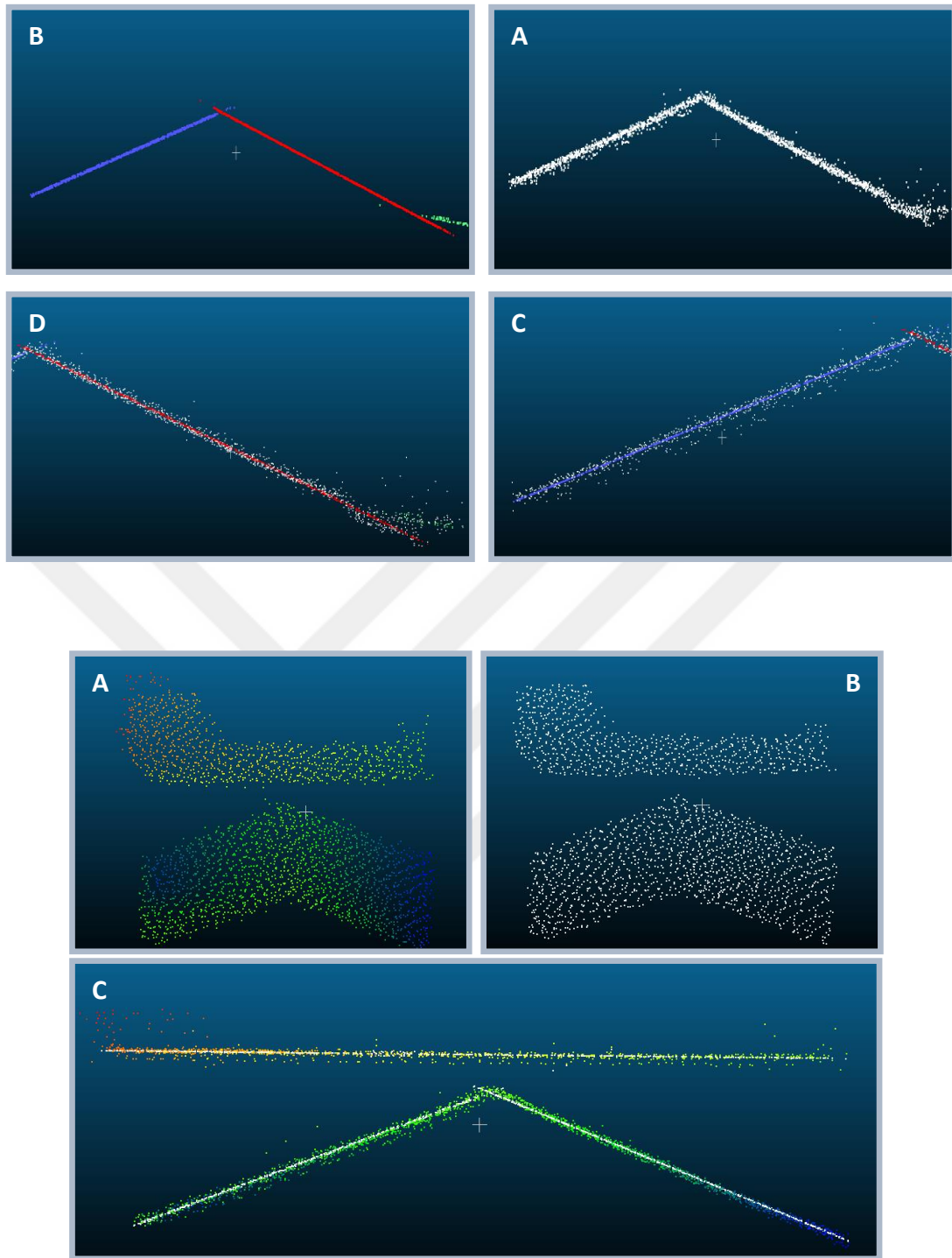


Figure 5.1. The comparison of the raw data (A) and RANSAC result (B) for buildings #1 and #8

5.2. Recommendations

Based on the results obtained and the observations made in this study, the followings are recommended for future studies:

1. In this research study, the developed RANSAC approach was tested on the point cloud data sets one building at a time. It is recommended that the efficiency of the approach can be increased by processing all the buildings at the same time.
2. The RANSAC algorithm selects three random points to cross the plane iteratively. It is recommended that an approach is developed that selects the points intuitively according the data features.
3. It is recommended the approach used in this study should be used in disaster monitoring and in damage detection.

REFERENCES

- [1] F. Rottensteiner, “Automatic Generation of High-Quality Building Models from Lidar Data,” *IEEE Comput. Graph. Appl.*, vol. 23, no. December, pp. 42–50, **2003**.
- [2] F. Tarsha-Kurdi, T. Landes, and P. Grussenmeyer, “Hough-Transform and Extended Ransac Algorithms for Automatic Detection of 3D Building Roof Planes From Lidar Data,” *ISPRS Work. Laser Scanning 2007 SilviLaser 2007*, vol. XXXVI, no. 1, pp. 407–412, **2007**.
- [3] H. G. Maas and G. Vosselman, “Two algorithms for extracting building models from raw laser altimetry data,” *ISPRS J. Photogramm. Remote Sens.*, vol. 54, no. 2–3, pp. 153–163, **1999**.
- [4] A. Nguyen and B. Le, “3D point cloud segmentation: A survey,” *IEEE Conf. Robot. Autom. Mechatronics, RAM - Proc.*, no. November, pp. 225–230, **2013**.
- [5] M. C. De Andrade, “An Interactive Algorithm for Image Smoothing and Segmentation,” *Electron. Lett. Comput. Vis. Image Anal.*, vol. 4, no. July, pp. 32–48, **2004**.
- [6] M. a Fischler and R. C. Bolles, “Paradigm for Model,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, **1981**.
- [7] R. Raguram, J. M. Frahm, and M. Pollefeys, “A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5303 LNCS, no. PART 2, pp. 500–513, **2008**.
- [8] S. Canaz, F. Karsli, A. Guneroglu, M. Dihkan, R. Sensing, and L. Detection, “LiDAR VERİLERİ KULLANILARAK GÖL SINIRLARININ OTOMATİK OLARAK BELİRLENMESİ,” pp. 287–293, **2015**.
- [9] F. Karsli and N. Pfeifer, “Ransac Algorithm for Automatic Extraction of the Ground Surface,” *IV. Uzaktan algılama ve Coğrafi Bilgi Sistemleri sempozyumu (UZAL-CBS 2012)*, 16-19 Ekim, Zonguldak, **2014**

- [10] R. Çömert and U. Avdan, "Yersel Lazer Tarayıcı Verilerden Basit Geometrik Yüzeylerin Otomatik Olarak Çıkarılması," *5.Uzaktan algılama-Cbs sempozyumu (UZAL-CBS 2014)*, 14-17 Ekim, İstanbul, **2014**
- [11] J. Wang and J. Shan, "Segmentation of LiDAR point clouds for building extraction," *Am. Soc. Photogramm. Remote Sens. ...*, pp. 9–13, **2009**.
- [12] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *ACM SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 71–78, **1992**.
- [13] F. Rottensteiner and C. Briese, "Automatic Generation of Building Models From Lidar Data and the Integration of Aerial Images," *Isprs*, vol. XXXIV, pp. 174–180, **2003**.
- [14] K. H. Kim and J. Shan, "Building roof modeling from airborne laser scanning data based on level set approach," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 4, pp. 484–497, **2011**.
- [15] N. Demir, "Lidar Verisinden Çatı Düzlemlerinin Otomatik Çıkarılması," *6. Uzaktan algılama-Cbs sempozyumu (UZAL-CBS 2016)*, 5-7 Ekim, Adana, **2016**
- [16] P. Axelsson, "Processing of laser scanner data—algorithms and applications," **1999**.
- [17] P. Rashidi and H. Rastiveis, "Ground filtering LiDAR data based on multi-scale analysis of height difference threshold," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, vol. 42, no. 4W4, pp. 225–229, **2017**.
- [18] "LASTools: software for rapid LiDAR processing."
<http://www.cs.unc.edu/~isenburg/lastools/guidelines-2..> (**Jun, 2018**)
- [19] "LASTools lasground." <https://rapidlasso.com/lastools/lasground/>. (**Jun, 2018**).
- [20] "LASTools las2las." <https://rapidlasso.com/lastools/las2las/>. (**Jun, 2018**).
- [21] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *ACM SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 71–78, **1992**.

- [22] E. G. Parmehr, M. Amati, and C. S. Fraser, "Mapping Urban Tree Canopy Cover Using Fused Airborne Lidar and Satellite Imagery Data," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. III-7, no. July, pp. 181–186, **2016**.
- [23] "LAStools lasclassify." <https://rapidlasso.com/lastools/lasclassify/>. (Jun, **2018**).
- [24] The American Society for Photogrammetry & Remote Sensing, "LAS 1.4 Draft Specification," *ASPRS, Am. Soc. Photogramm. Remote Sens.*, no. August, pp. 1–18, **2011**.
- [25] D. Girardeau-Montaut, "CloudCompare Software version 2.6.1. user manual," p. 181, **2015**.
- [26] D. Girardeau-Montaut, M. Roux, R. Marc, and G. Thibault, "Change detection on points cloud data acquired with a ground laser scanner," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 36, no. 3, p. W19, **2005**.
- [27] D. Hetherington, *Topographic Laser Ranging and Scanning: Principles and Processing*, vol. 31, no. 12. **2010**.
- [28] Strang, Gilbert; Borre, Kai, *Linear Algebra, Geodesy, and GPS*, SIAM, pp. 22–23, ISBN 9780961408862. (**1997**)
- [29] Anton, Howard, *Elementary Linear Algebra* (7th ed.), John Wiley & Sons, ISBN 0-471-58742-7. (**1994**)
- [30] C. H. Lin, J. Y. Chen, P. L. Su, and C. H. Chen, "Eigen-feature analysis of weighted covariance matrices for LiDAR point cloud classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 94, pp. 70–79, **2014**.

CURRICULUM VITAE

Credentials

Name, Surname : Khalil Karbalai Mohammadzadeh

Place of Birth : Bonab, Iran

Marital Status : Single

E-mail : enkhalil2010@gmail.com

Address : Geomatics Engineering Department of Hacettepe University
Beytepe – ANKARA

Education

High School : Energy Nour High School, Iran

BSc. : Geomatic Engineering, Islamic Azad University, Iran

MSc. :

PhD. :

Foreign Languages

English, Persian

Work Experiences

Areas of Experiences

Projects and Budgets

Publications

Oral and Poster Presentations



**HACETTEPE UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING
THESIS/DISSERTATION ORIGINALITY REPORT**

**HACETTEPE UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING
TO THE DEPARTMENT OF GEOMATICS ENGINEERING**

Date: 26/06/2018

Thesis Title / Topic: AUTOMATIC ROOF PLANE EXTRACTION FROM LIDAR DATA USING RANSAC ALGORITHM

According to the originality report obtained by myself/my thesis advisor by using the *Turnitin* plagiarism detection software and by applying the filtering options stated below on 26/06/2018 for the total of 67 pages including the a) Title Page, b) Introduction, c) Main Chapters, d) Conclusion sections of my thesis entitled as above, the similarity index of my thesis is 7 %.

Filtering options applied:

1. Bibliography/Works Cited excluded
2. Quotes excluded /included
3. Match size up to 5 words excluded

I declare that I have carefully read Hacettepe University Graduate School of Science and Engineering Guidelines for Obtaining and Using Thesis Originality Reports; that according to the maximum similarity index values specified in the Guidelines, my thesis does not include any form of plagiarism; that in any future detection of possible infringement of the regulations I accept all legal responsibility; and that all the information I have provided is correct to the best of my knowledge.

I respectfully submit this for approval.

26/06/2018

Name Surname: KHALIL KARBALAI MOHAMMADZADEH

Student No: N13221205

Department: GEOMATICS ENGINEERING

Program: GEOMATICS

Status: ☒ Masters ☐ Ph.D. ☐ Integrated Ph.D.

ADVISOR APPROVAL

APPROVED.

Prof. Dr. Mustafa TÜRKER