HOMEWORK 2

PREPARED BY

NAME SURNAME	: ONAT BİNGÖL
STUDENT NUMBER	**:** 010180617
DUE DATE	**:** 02/05/2023

COURSE NAME **:** 3D MODELING WITH REMOTE SENSING DATA

LECTURER **:** ASSOC. PROF. DR. ESRA ERTEN

1) Display all PCD

In the first stage of our homework, we opened the data given to us with the CloudCompare application and colored it according to height in order to increase its clarity at first glance.

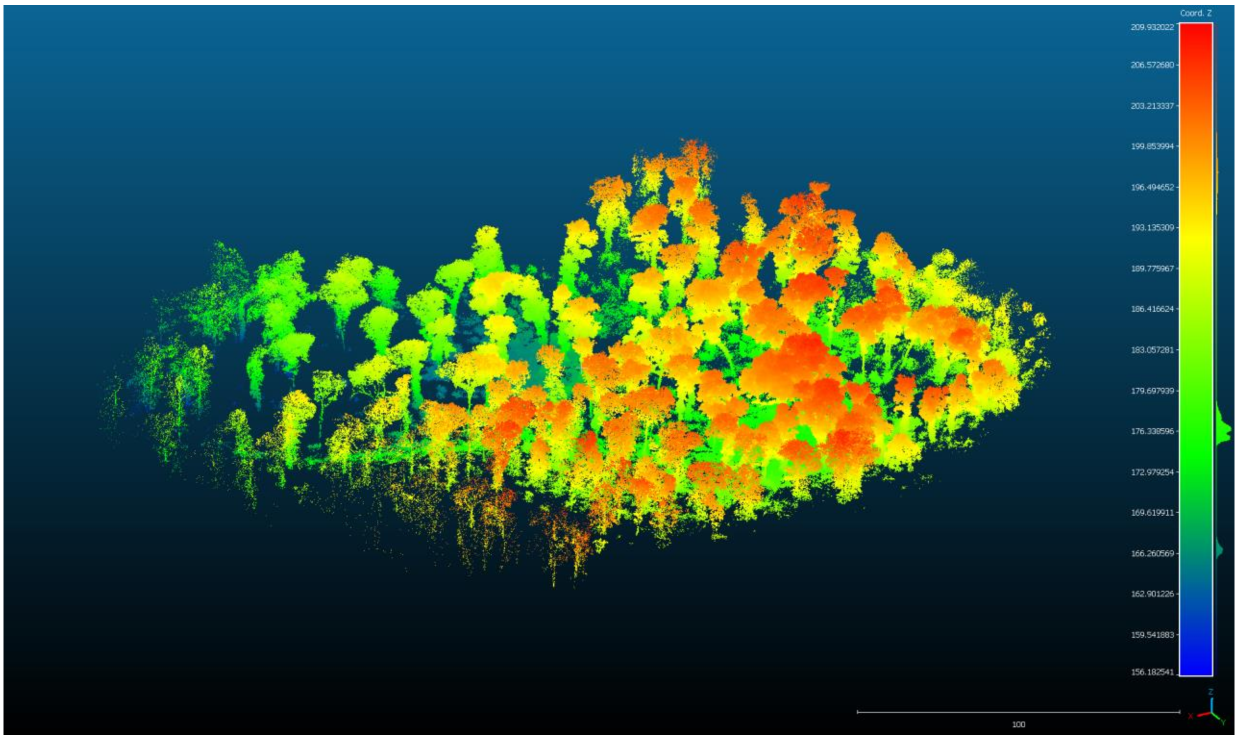


Image 1 All Point Clouds Color by Height

2) Tree trunk individualization: select and display the extracted tree

In order to find our own tree in the data we transferred to CloudCompare, we transferred the coordinates ((X) 407020.280436, (Y) 4566967.013238, (Z) 168.02525) of our point from the coordinate list uploaded to the class files to the application and first cleaned the tree in this coordinate from large and unnecessary data by rough cutting. We then extracted our own tree from this rough cut. Finally, in this step, we colored the tree according to height to increase its clarity.

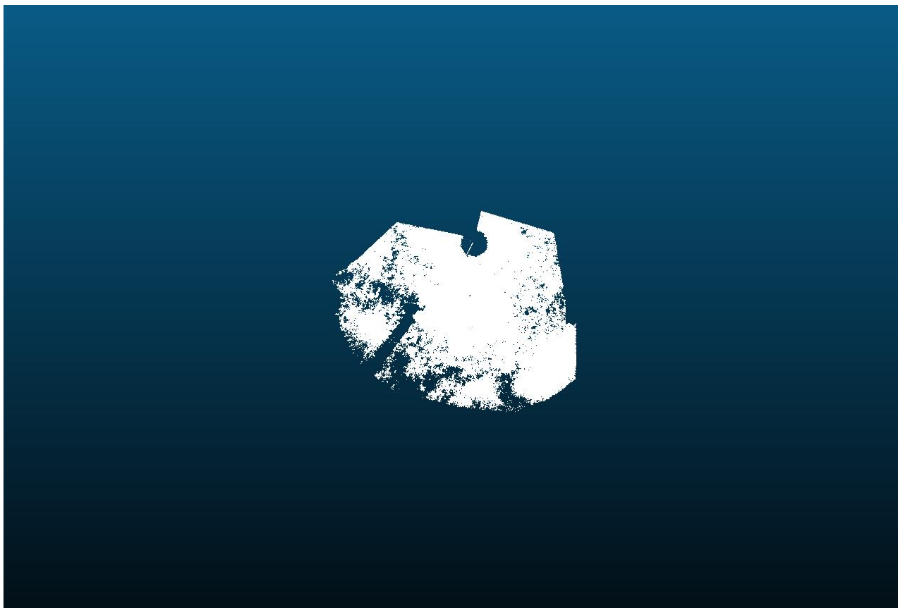


Image 2 Roughly Retrieved Area of Interest From All Point Cloud Data.

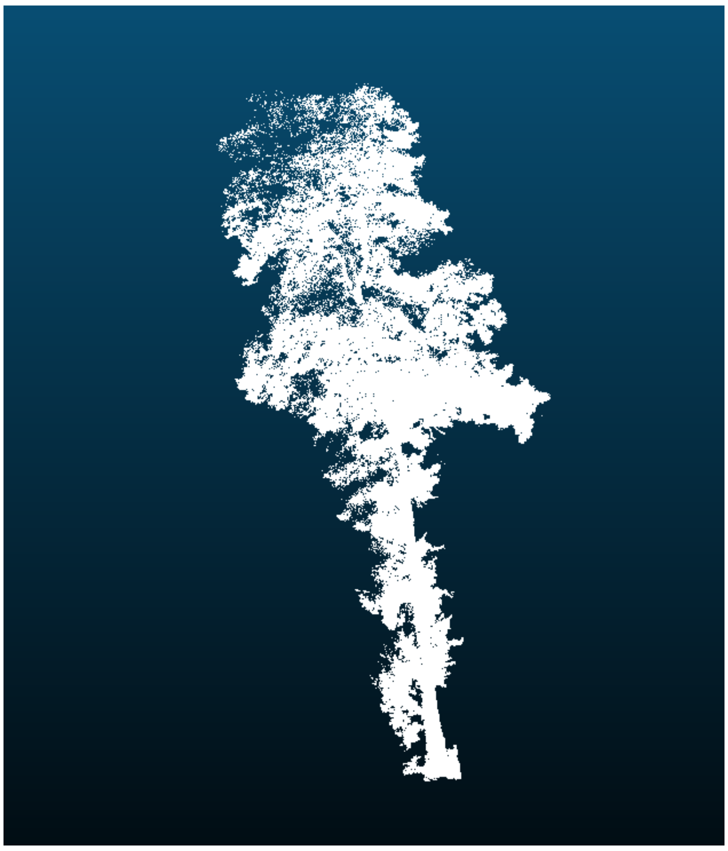


Image 3 The Related Tree from the Entire Point Cloud, Cleared of other Unnecessary Points.

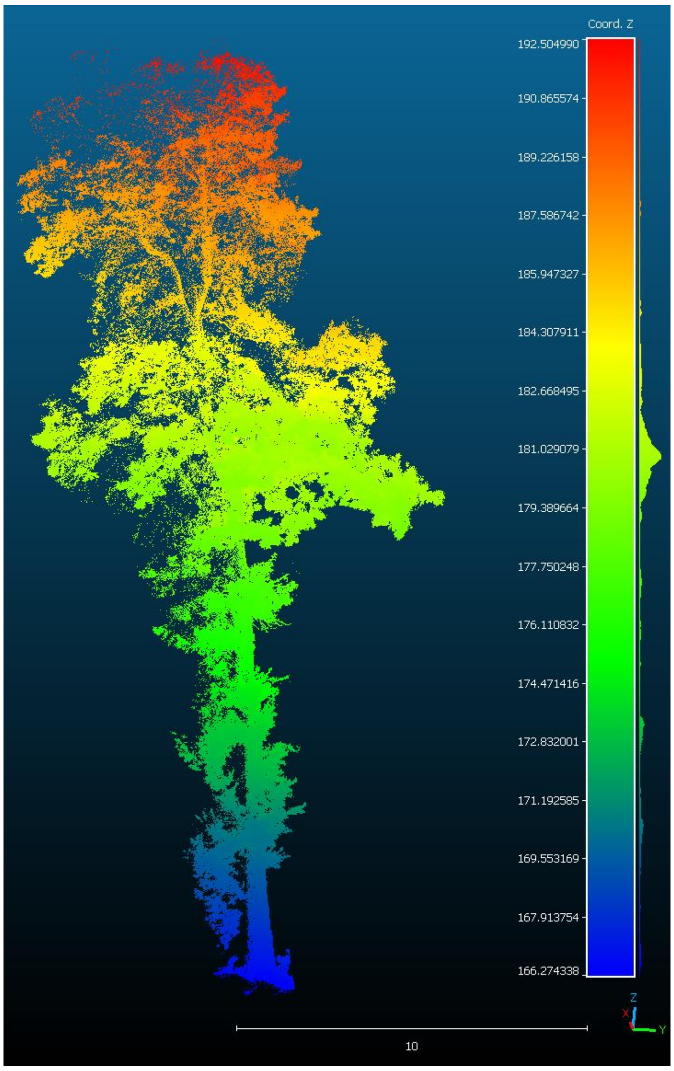
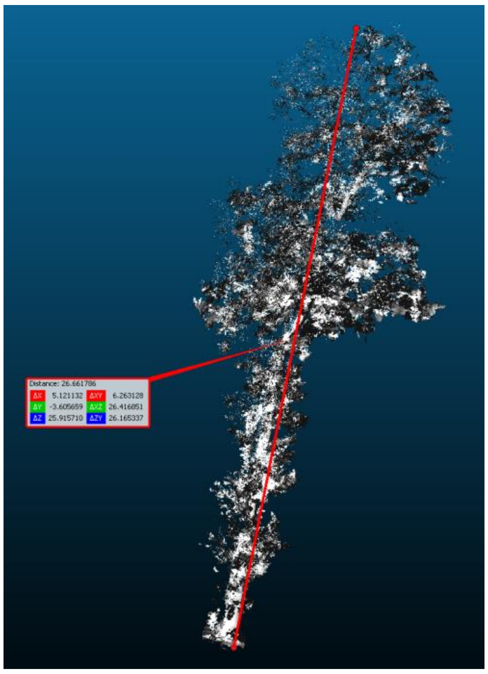


Image 4 Colored Version of the Related Tree, Cleared of Unnecessary Points, According to Height.

3) Calculate the tree trunk height

Used the length measurement tool on the CloudCompare application, the height of the tree whose coordinates were given was measured.



```
Distance: 26.661786
ΔX   5.121132   ΔXY   6.263128
ΔY  -3.605659   ΔXZ  26.416851
ΔZ  25.915710   ΔZY  26.165337
```

Image 5 Measuring Tree Height Using Tool in Cloudcompare App.

After these steps, several different operations were performed to find the cylinder closest to the shape of the tree trunk to be made. The first step was to create the normal of the tree.

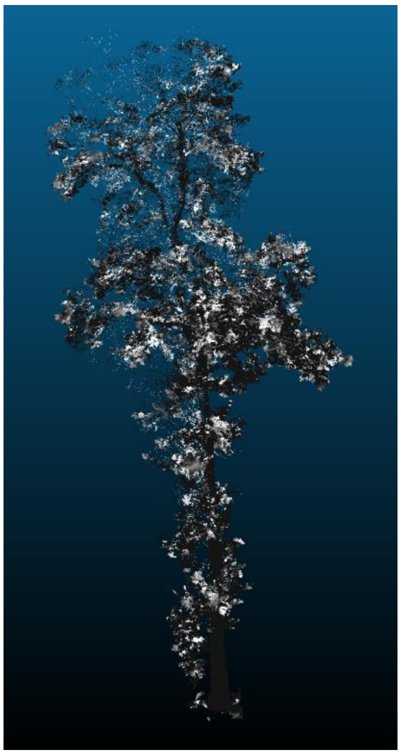


Image 6 Calculated Version of the Normal of the Tree.

After we created the tree normal, we did the poisson Recon Rendering of the tree, but a few adjustments were made via the SF Display parameter to make the tree look like cotton candy more understandable when it was first made.

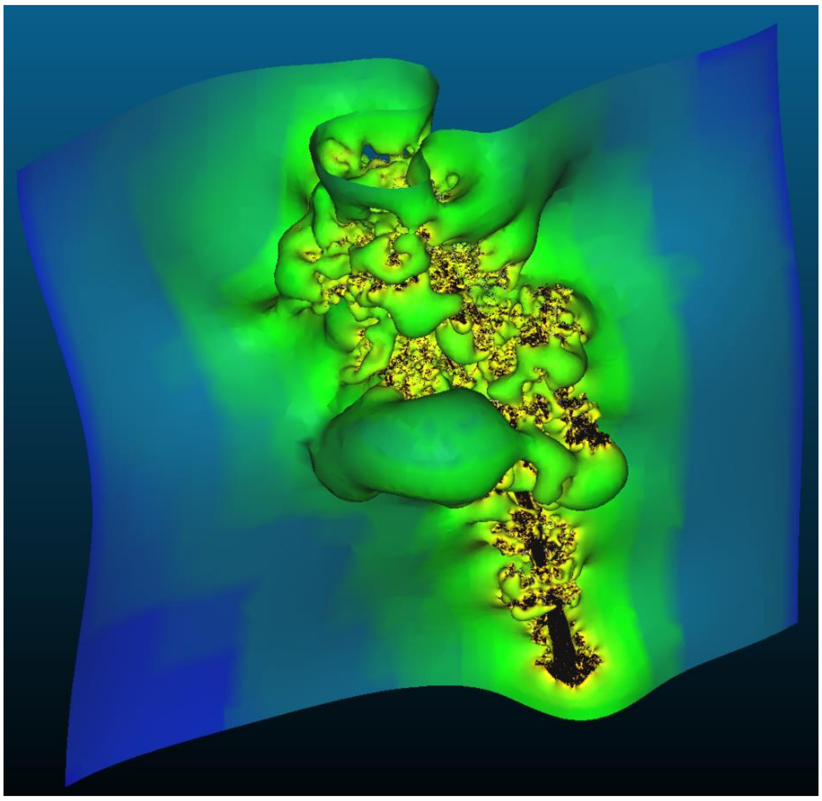
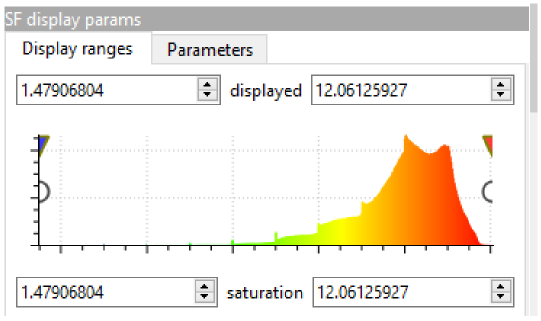


Image 7 Poisson Recon Rendering of the Tree.



SF display params

Display ranges | Parameters

1.47906804 displayed 12.06125927

1.47906804 saturation 12.06125927

Image 8 SF Display Parameters Before Setting.

We have increased the lowest value via the SF Display parameter and it has been set to a value where the leaves, branches and trunk colors of the tree will become evident.

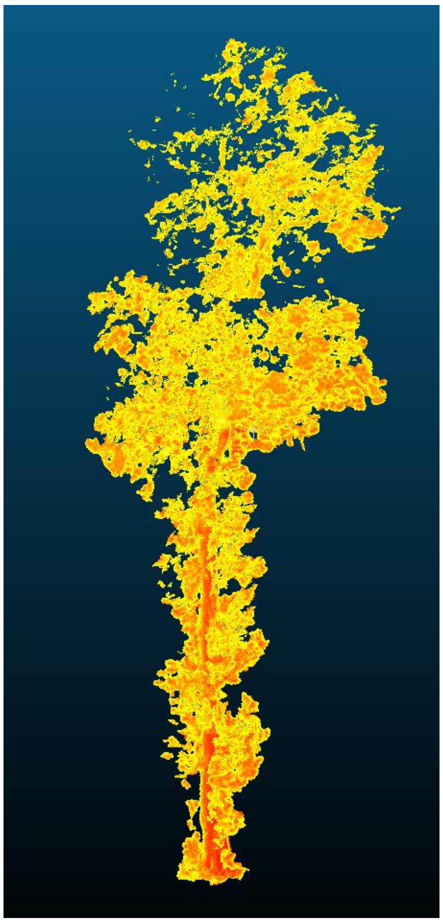
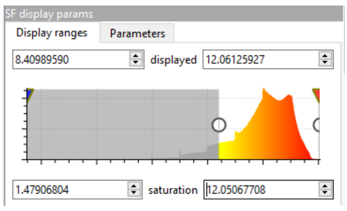


Image 9 Poisson Recon Rendering of the Tree After SF. Setting.



SF display params

Display ranges | Parameters

8.40989590 displayed 12.06125927

1.47906804 saturation 12.05067708

Image 10 SF Display Parameters After Setting.

After the adjustments were made, the branches and leaves that made it difficult to see, understand and measure around the tree trunk were removed.

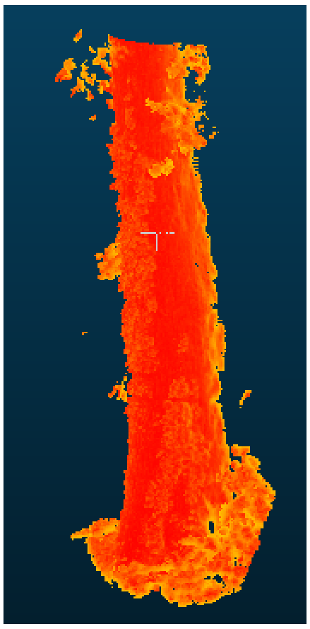


Image 11 The Entire Image of the Tree is Cleared and Only the Trunk is Left.

The tree trunk was shortened according to the desired value of 1.30<x<1.40 meters.

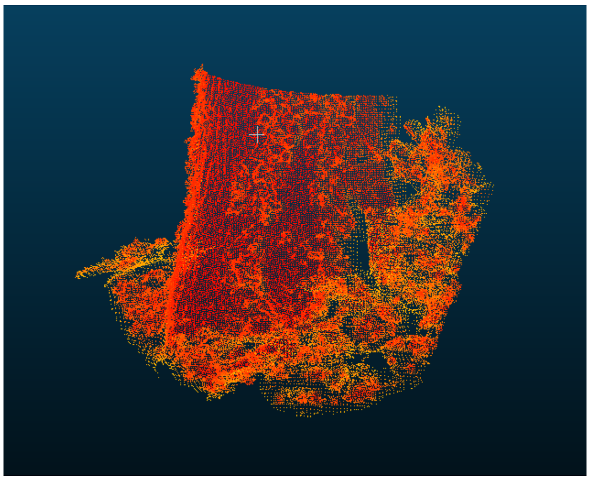


Image 12 The shortened version of the body length according to the desired length.

The most suitable cylindrical shape was adopted for the outer wall of the tree trunk, which was shortened in accordance with the given dimensions.

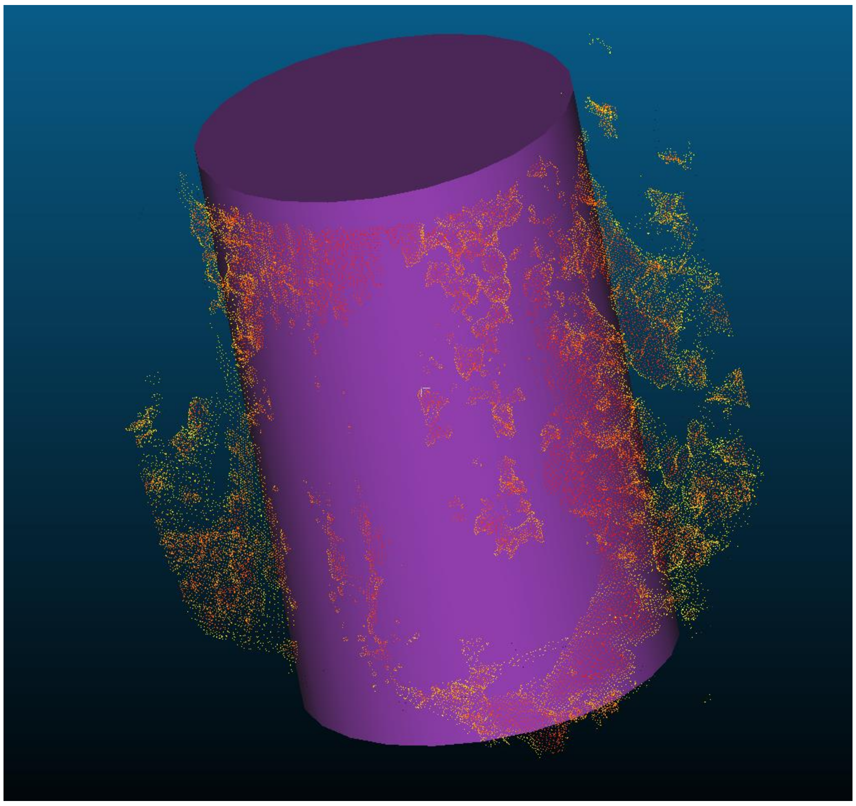


Image 13 The Most Suitable Cylinder Placed on the Shortened Body.

☑ Cylinder (r=0.412282/h=1.293182)

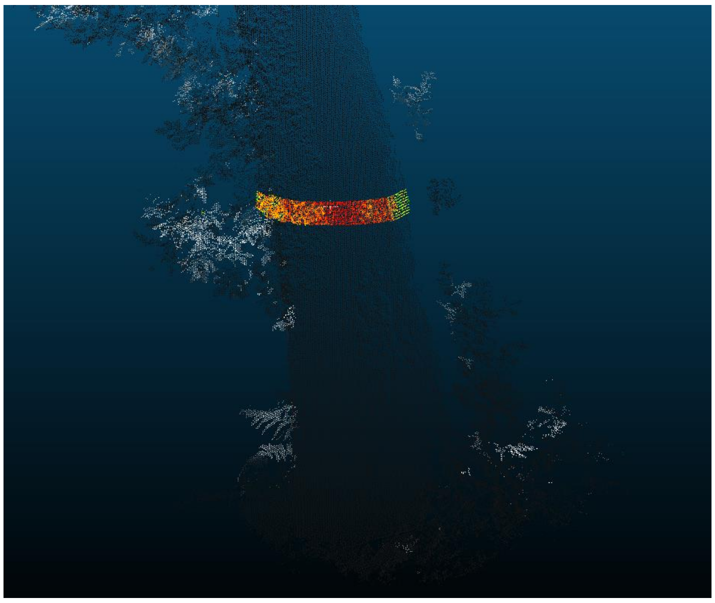


Image 14 A Section of the Body Shown.

4) Calculate the BDH at 1.35 m from the terrain using circle fitting (e.g. points where 1.30 > ztree > 1.40)

The code was written in Python by adding the center point of the coordinates in the line of the appropriate cylinder drawn, and this code was run and the drawing was made on the circle plot with the calculations made on the cylinder.

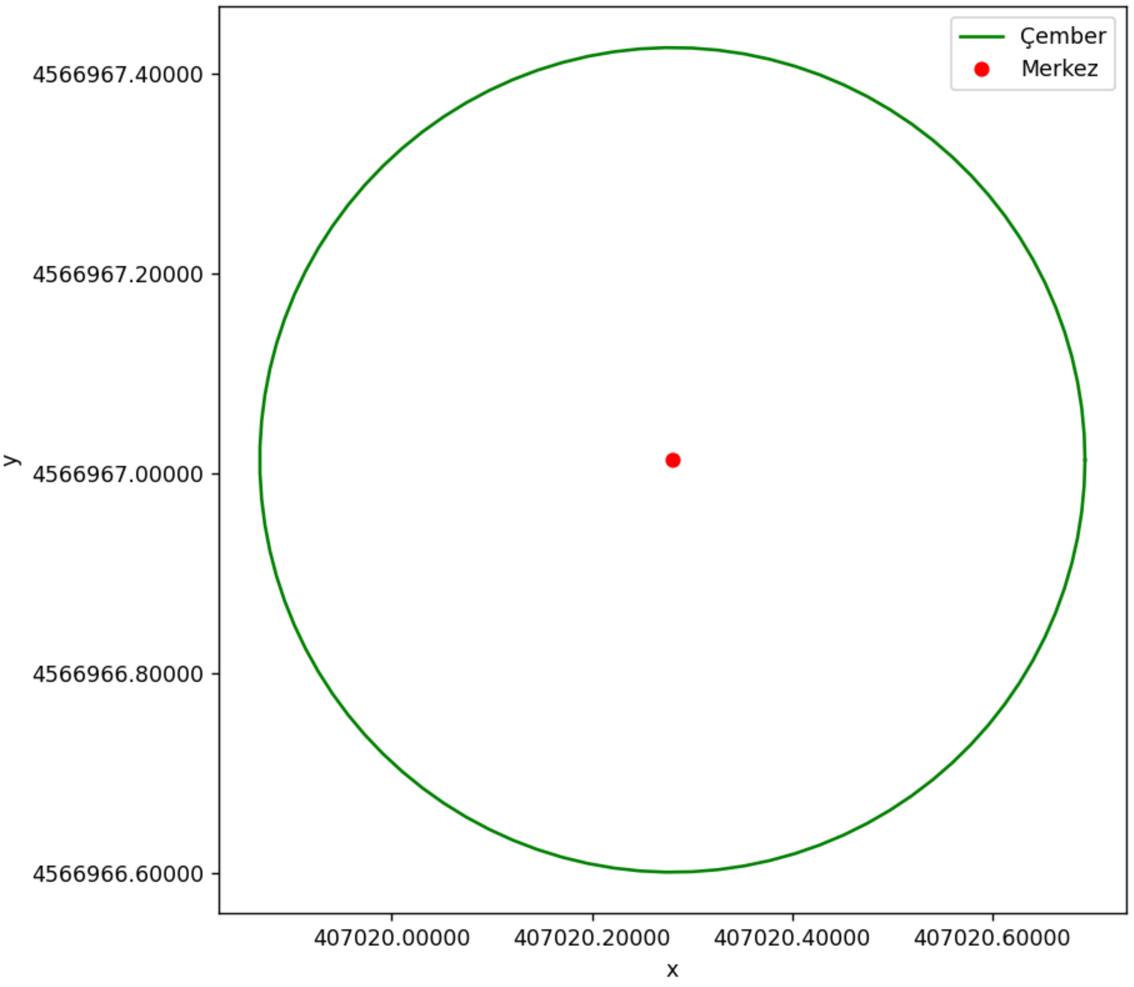


Image 15 The Closest Size Circle to the Tree Trunk is Drawn with Python.

5) Calculate the BDH at 1.35 m from the terrain using ellipsoid fitting (e.g. points where 1.30 > ztree > 1.40)

The code was written in Python by adding the center point of the coordinates in the line of the appropriate cylinder drawn, and this code was run and the drawing was made on the ellipse plot with the calculations made on the cylinder.

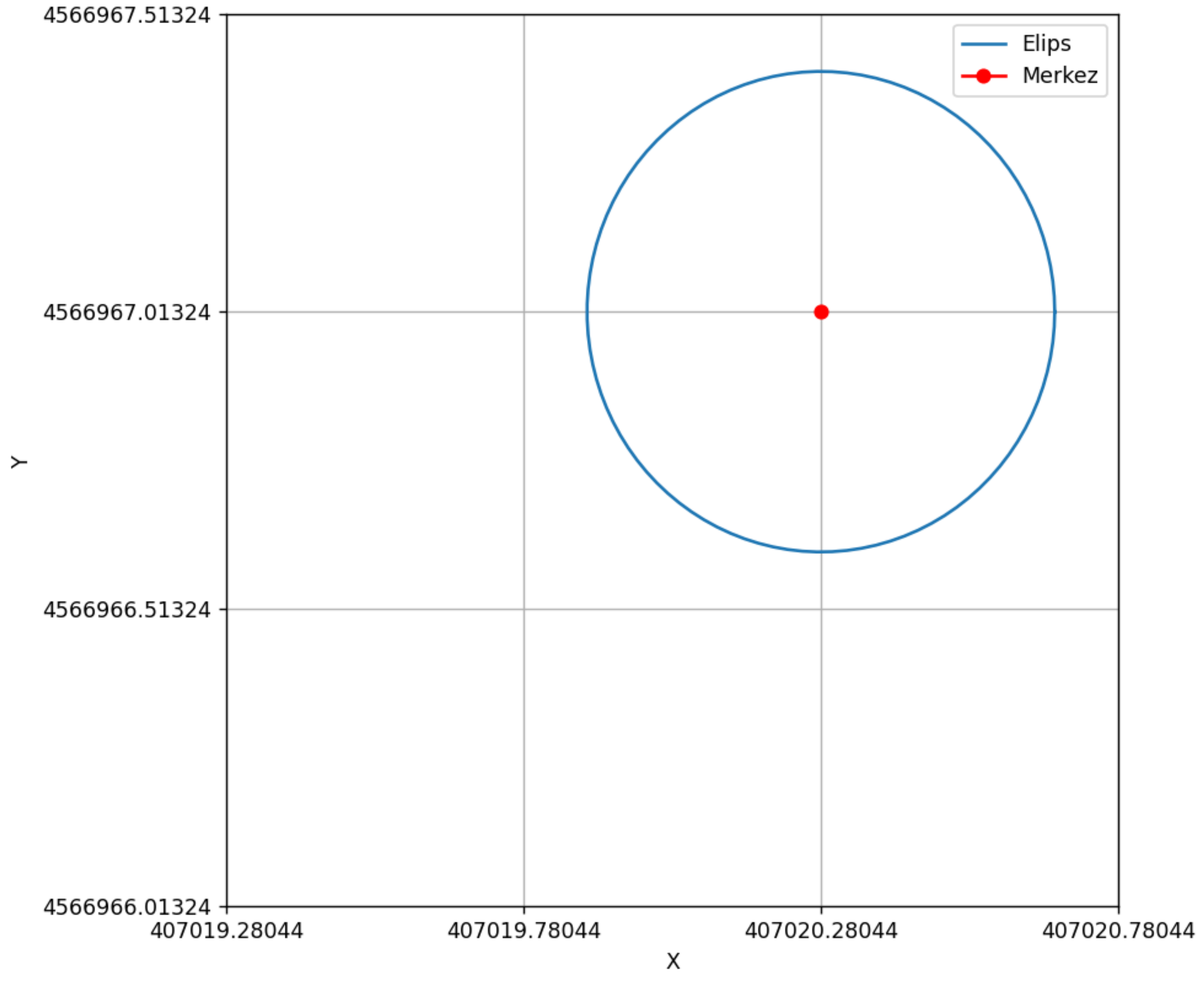


Image 16 The Closest Size Ellipse to the Tree Trunk is Drawn with Python.

6) Visualize both BDH calculations on a single plot.

The code was written in Python by adding the center point of the coordinates in the line of the appropriate cylinder drawn, and by running this code, the calculations on the cylinder were drawn on a single plot of circle and ellipse.

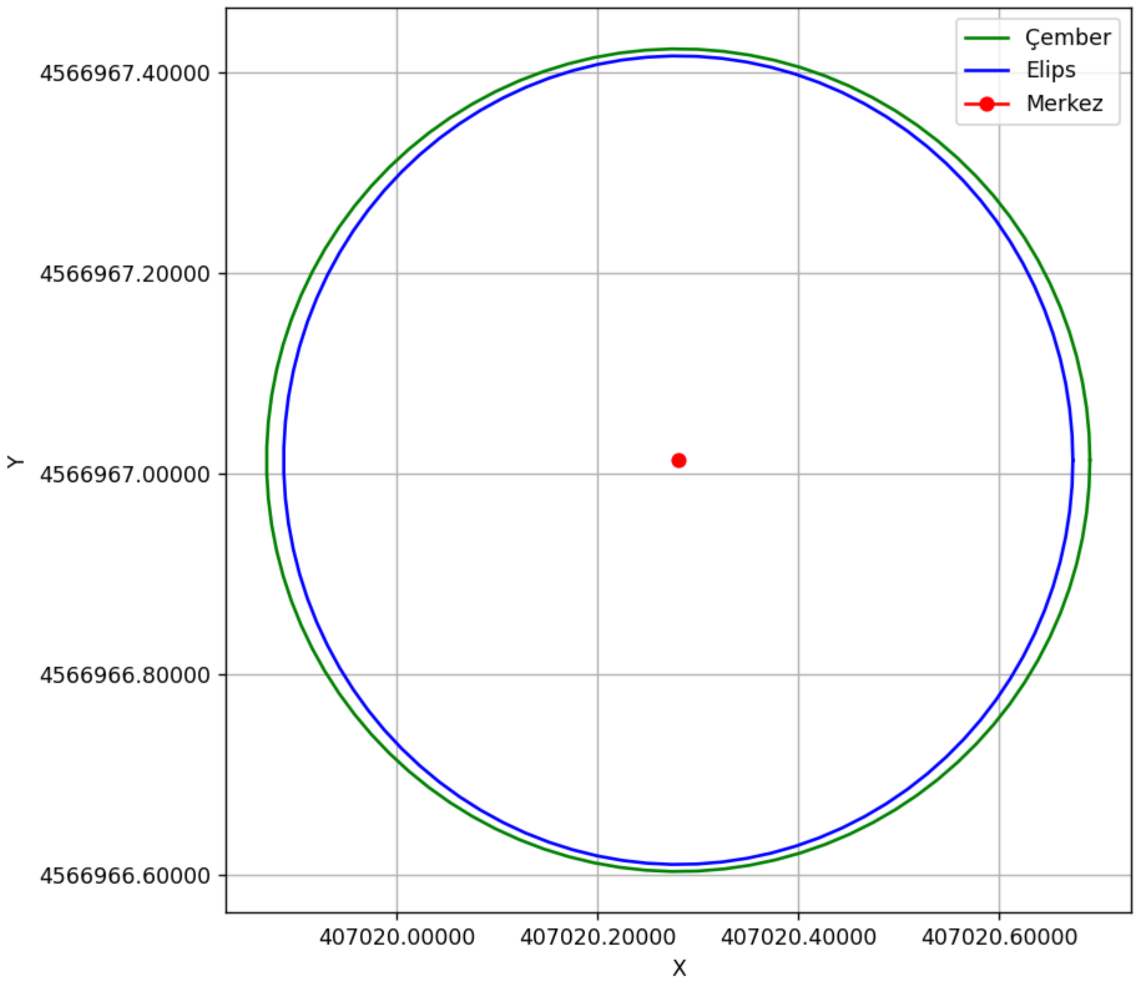


Image 17 The Closest Sized Circle and Ellipse to the Tree Trunk Were Drawn Together with Python.

7) Discussing the results and representing your project (Background/Results/Conclusions).

In conclusion, the Cloud Compare application was used to identify and isolate a specific tree within a forest area with coordinates (X) 407020.280436, (Y) 4566967.013238, and (Z) 168.022525. After clearing the surrounding trees and exposing the trunk, measurements were taken to determine the trunk height, Breast Height Diameter (BHD), cylinder radius (r), and cylinder height (h) of the tree. The cylinder radius was found to be 0.412282 m, the cylinder height was 1.293182 m, and the tree length was 26.661786 m.

With these measurements, the biomass of the tree can be estimated using the BHD method, applying an appropriate allometric equation specific to the tree species and region. This information will help in understanding the tree's contribution to carbon sequestration and overall forest management practices. By successfully completing this assignment, you have shown that it is possible to examine individual trees in detail and estimate their biomass, which is crucial for sustainable forest management, environmental monitoring, and climate change research. This project also demonstrated the use of 3D modeling software.

ATTACHMENT

Circle

```
import numpy as np

import matplotlib.pyplot as plt

x_merkez, y_merkez = 407020.280436, 4566967.013238

yaricap = 0.412282

acilar = np.linspace(0, 2 * np.pi, 100

x = x_merkez + yaricap * np.cos(acilar)

y = y_merkez + yaricap * np.sin(acilar)

plt.figure(figsize=(8, 8))

plt.plot(x, y, label="Çember", color="green")

plt.plot(x_merkez, y_merkez, 'ro', label="Merkez")

plt.gca().set_aspect('equal', adjustable='box')

plt.xlabel("x")

plt.ylabel("y")

plt.legend()

plt.gca().xaxis.set_major_formatter(plt.FormatStrFormatter('%.5f'))

plt.gca().yaxis.set_major_formatter(plt.FormatStrFormatter('%.5f'))

plt.show()
```

Ellipse

```
import numpy as np

import matplotlib.pyplot as plt

center_x, center_y = 407020.280436, 4566967.013238

ellipse_a, ellipse_b = 0.393177, 0.4039785

theta = np.linspace(0, 2 * np.pi, 100)

ellipse_x = center_x + ellipse_a * np.cos(theta)

ellipse_y = center_y + ellipse_b * np.sin(theta)

fig, ax = plt.subplots()

ax.plot(ellipse_x, ellipse_y, label="Elips")

ax.plot(center_x, center_y, marker="o", color="red", label="Merkez")

ax.set_aspect("equal")

ax.set_xticks(np.arange(center_x - 1, center_x + 1, 0.5))

ax.set_yticks(np.arange(center_y - 1, center_y + 1, 0.5))

ax.xaxis.set_major_formatter(plt.FormatStrFormatter('%.5f'))

ax.yaxis.set_major_formatter(plt.FormatStrFormatter('%.5f'))

ax.set_xlabel("X")

ax.set_ylabel("Y")

ax.grid(True)

ax.legend()

plt.show()
```

Ellipse and Circle

```
import numpy as np

import matplotlib.pyplot as plt

center_x, center_y = 407020.280436, 4566967.013238

circle_radius = 0.41

ellipse_a, ellipse_b = 0.393, 0.403

theta = np.linspace(0, 2 * np.pi, 100)

circle_x = center_x + circle_radius * np.cos(theta)

circle_y = center_y + circle_radius * np.sin(theta)

ellipse_x = center_x + ellipse_a * np.cos(theta)

ellipse_y = center_y + ellipse_b * np.sin(theta)

fig, ax = plt.subplots()

ax.plot(circle_x, circle_y, label="Çember", color="green")

ax.plot(ellipse_x, ellipse_y, label="Elips", color="blue")

ax.plot(center_x, center_y, marker="o", color="red", label="Merkez")

ax.set_aspect("equal")

ax.xaxis.set_major_formatter(plt.FormatStrFormatter('%.5f'))

ax.yaxis.set_major_formatter(plt.FormatStrFormatter('%.5f'))

ax.set_xlabel("X")

ax.set_ylabel("Y")

ax.grid(True)
ax.legend()

plt.show()
```