LiDAR Data Tree Segmentation and Classification

Manoneet Gawali
231030035
manoneetag23@iitk.ac.in
Department of Civil Engineering

1. Objective

Become familiar with TLS data processing in forests and learn the process of individual tree segmentation. Once the Trees are segmented label it and test for accuracy against the actual data.

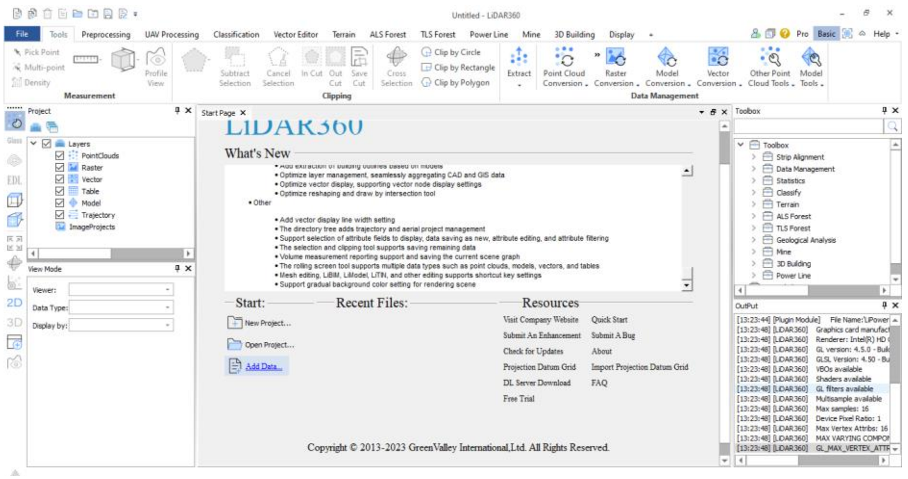
2. Introduction

Terrestrial Laser Scanning (TLS) is a specific application of LiDAR that involves the collection of point cloud data from the ground, allowing for precise and detailed characterization of the forest environment. In this analysis, we combine TLS data from various forest plots and employ machine learning techniques to classify individual trees based on their point cloud data. Specifically, a Random Forest classifier is used to differentiate between trees by analyzing their spatial coordinates (X, Y, Z). The model's performance is evaluated through confusion matrices and classification reports to identify misclassified points, which are further analyzed to understand the model's weaknesses and improve accuracy. Misclassified data points are investigated, where the predicted label represents the actual tree type, allowing for refined tree segmentation and better model training

3. Methodology

A. Denormalization in LiDAR 360

a. Install LiDAR360 and enter the license code to activate the software.



- Add extraction of building outlines based on models
- Optimize layer management, seamlessly aggregating CAD and GIS data
- Optimize vector display, supporting vector node display settings
- Optimize reshaping and draw by intersection tool
- Other
  - Add vector display line width setting
  - The directory tree adds trajectory and aerial project management
  - Support selection of attribute fields to display, data saving as new, attribute editing, and attribute filtering
  - The selection and clipping tool supports saving remaining data
  - Volume measurement reporting support and saving the current scene graph
  - The rolling screen tool supports multiple data types such as point clouds, models, vectors, and tables
  - Mesh editing, LiBIM, LiModel, LiTIN, and other editing supports shortcut key settings
  - Support gradual background color setting for rendering scene

Copyright © 2013-2023 GreenValley International,Ltd. All Rights Reserved.

```
[13:23:44] [Plugin Module]    File Name:'LiPower
[13:23:48] [LiDAR360]    Graphics card manufact
[13:23:48] [LiDAR360]    Renderer: Intel(R) HD
[13:23:48] [LiDAR360]    GL version: 4.5.0 - Buil
[13:23:48] [LiDAR360]    GLSL Version: 4.50 - Bu
[13:23:48] [LiDAR360]    VBOs available
[13:23:48] [LiDAR360]    Shaders available
[13:23:48] [LiDAR360]    GL filters available
[13:23:48] [LiDAR360]    Multisample available
[13:23:48] [LiDAR360]    Max samples: 16
[13:23:48] [LiDAR360]    Device Pixel Ratio: 1
[13:23:48] [LiDAR360]    Max Vertex Attribs: 16
[13:23:48] [LiDAR360]    MAX VARYING COMPON
[13:23:48] [LiDAR360]    GL_MAX_VERTEX_ATTR
```

Figure 1 LiDAR 360 with add data highlighted

b. Open the raw data file via File > Add Data.

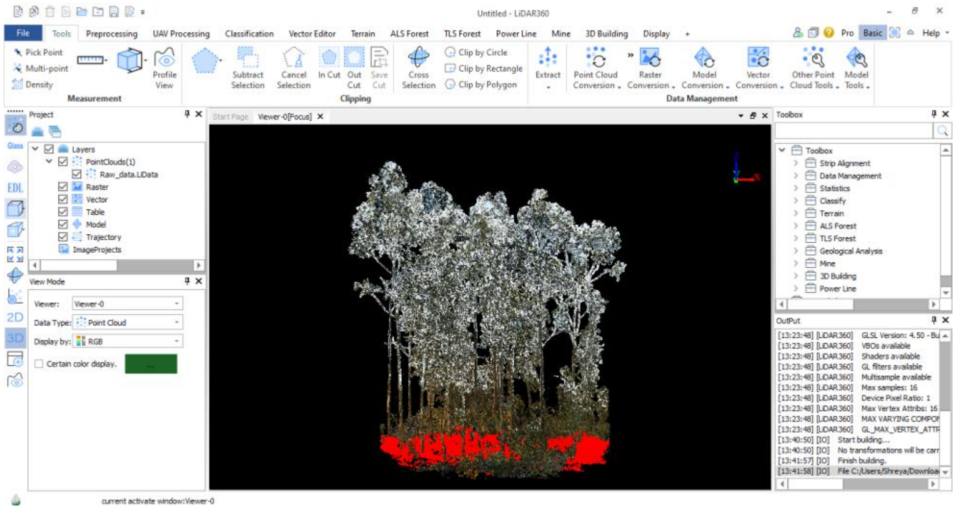
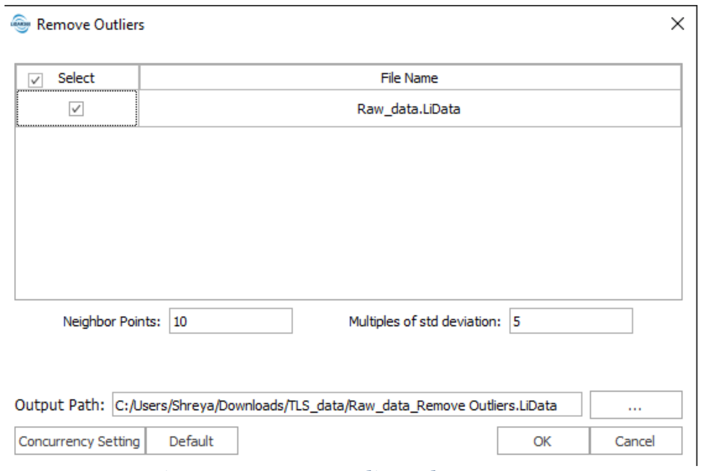

```
[13:23:48] [LiDAR360]   GLSL Version: 4.50 - Bu
[13:23:48] [LiDAR360]   VBOs available
[13:23:48] [LiDAR360]   Shaders available
[13:23:48] [LiDAR360]   GL filters available
[13:23:48] [LiDAR360]   Multisample available
[13:23:48] [LiDAR360]   Max samples: 16
[13:23:48] [LiDAR360]   Device Pixel Ratio: 1
[13:23:48] [LiDAR360]   Max Vertex Attribs: 16
[13:23:48] [LiDAR360]   MAX VARYING COMPON
[13:23:48] [LiDAR360]   GL_MAX_VERTEX_ATTR
[13:40:50] [IO]   Start building...
[13:40:50] [IO]   No transformations will be carr
[13:41:57] [IO]   Finish building.
[13:41:58] [IO]   File C:/Users/Shreya/Downloa
```

Figure 2 Data view in Lidar 360

c. Perform Individual Tree Segmentation (ITS) by following these steps:
The whole pipeline is:
a Removing outliers, if any.
The outliers are removed by using remove outlier toolbox as shown below. Go to Data Management> Remove Outliers. Double click on it. You can set the parameters accordingly. For this exercise, keep default parameters. Click OK.


Remove Outliers

✓ Select	File Name
✓	Raw_data.LiData

Neighbor Points: 10 Multiples of std deviation: 5

Output Path: C:/Users/Shreya/Downloads/TLS_data/Raw_data_Remove Outliers.LiData ...

Concurrency Setting | Default | OK | Cancel

Figure 3 remove outlier tab

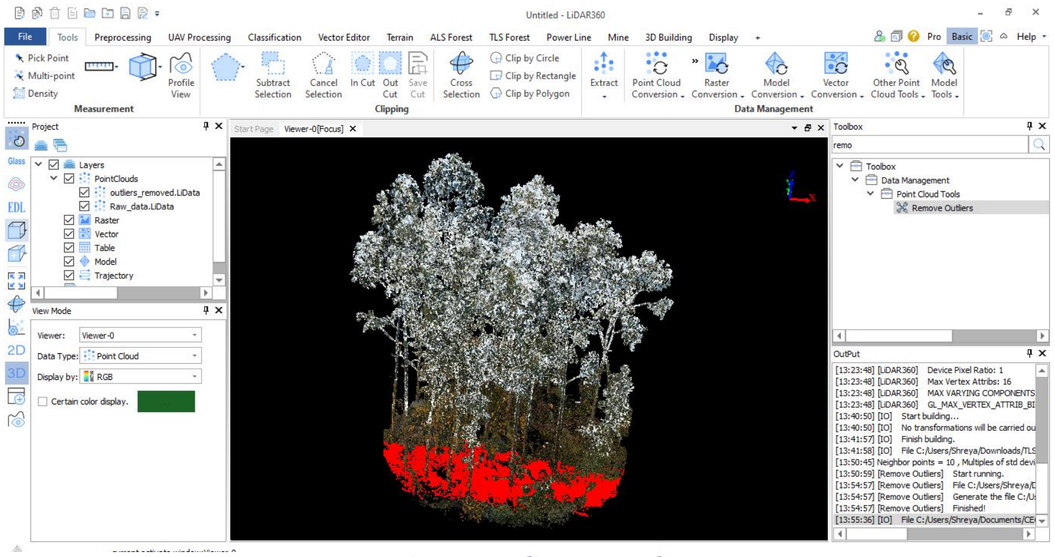
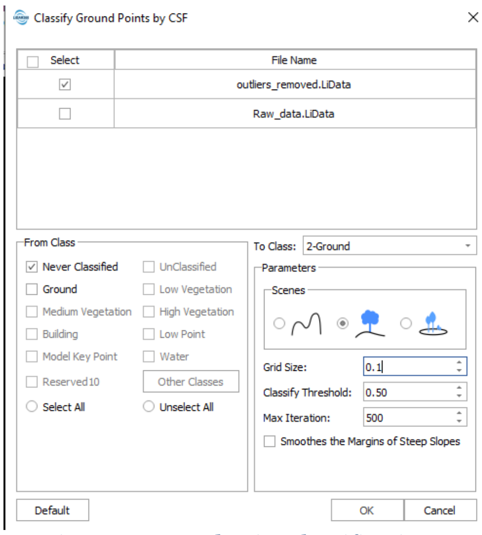

```
[13:23:48] [LiDAR360]   Device Pixel Ratio: 1
[13:23:48] [LiDAR360]   Max Vertex Attribs: 16
[13:23:48] [LiDAR360]   MAX VARYING COMPONENTS
[13:23:48] [LiDAR360]   GL_MAX_VERTEX_ATTRIB_BI
[13:40:50] [IO]   Start building...
[13:40:50] [IO]   No transformations will be carried ou
[13:41:57] [IO]   Finish building.
[13:41:58] [IO]   File C:/Users/Shreya/Downloads/TLS
[13:50:45] Neighbor points = 10 , Multiples of std devi
[13:50:59] [Remove Outliers]   Start running.
[13:54:57] [Remove Outliers]   File C:/Users/Shreya/D
[13:54:57] [Remove Outliers]   Generate the file C:/U
[13:54:57] [Remove Outliers]   Finished!
[13:55:36] [IO]   File C:/Users/Shreya/Documents/CE
```

Figure 4 outlier removed

b Classify ground points and non-ground points using CSF filter. Then go tot Classify> Classify Ground Points by CSF. Double click on it. Choose the terrain type as Gentle Terrain and set grid size equals to 0.1. Keep all other values as default. Click OK.


Classify Ground Points by CSF

Select	File Name
☑	outliers_removed.LiData
☐	Raw_data.LiData

From Class: ☑ Never Classified

To Class: 2-Ground

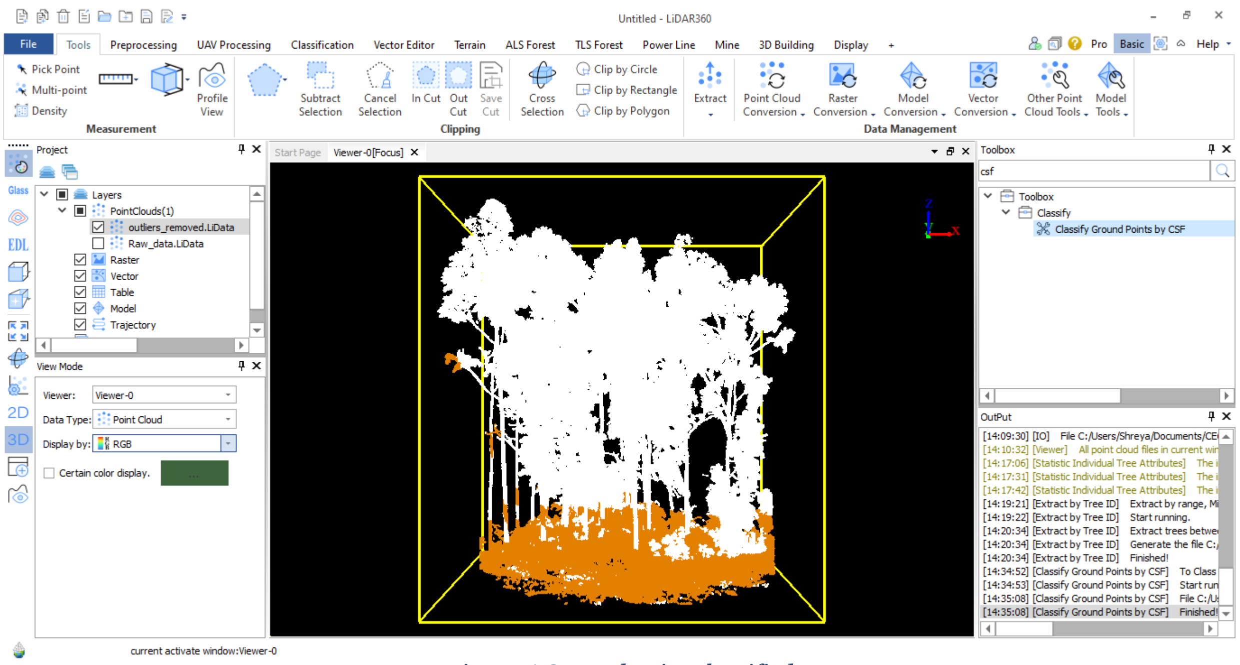
Grid Size: 0.1

Classify Threshold: 0.50

Max Iteration: 500

☐ Smoothes the Margins of Steep Slopes

Figure 5 Ground point classification

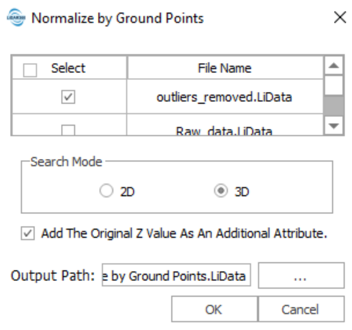

Untitled - LiDAR360

```
[14:09:30] [IO]    File C:/Users/Shreya/Documents/CE
[14:10:32] [Viewer]    All point cloud files in current win
[14:17:06] [Statistic Individual Tree Attributes]    The i
[14:17:31] [Statistic Individual Tree Attributes]    The i
[14:17:42] [Statistic Individual Tree Attributes]    The i
[14:19:21] [Extract by Tree ID]    Extract by range, Mi
[14:19:22] [Extract by Tree ID]    Start running.
[14:20:34] [Extract by Tree ID]    Extract trees betwe
[14:20:34] [Extract by Tree ID]    Generate the file C:/
[14:20:34] [Extract by Tree ID]    Finished!
[14:34:52] [Classify Ground Points by CSF]    To Class
[14:34:53] [Classify Ground Points by CSF]    Start run
[14:35:08] [Classify Ground Points by CSF]    File C:/U
[14:35:08] [Classify Ground Points by CSF]    Finished!
```

current activate window:Viewer-0

Figure 6 Ground point classified

c. Normalize the point cloud using ground. Next, go to Data Management> Normalize by Ground Points. Double click on it. Choose search mode as 3D and tick the option 'Add the Original Z value as an Additional Attribute'



Normalize by Ground Points

Select	File Name
✓	outliers_removed.LiData
☐	Raw_data.LiData

Search Mode: ○ 2D [✓ 3D]

☑ Add The Original Z Value As An Additional Attribute.

Output Path: e by Ground Points.LiData

Figure 8 normalized options

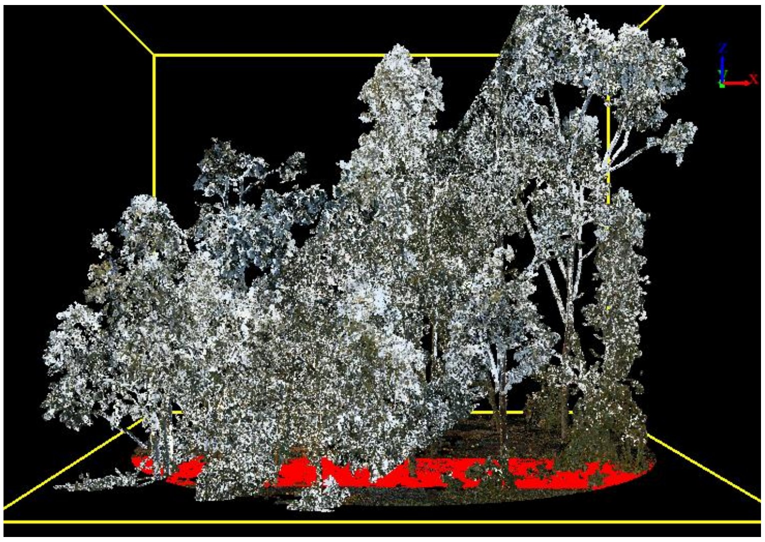
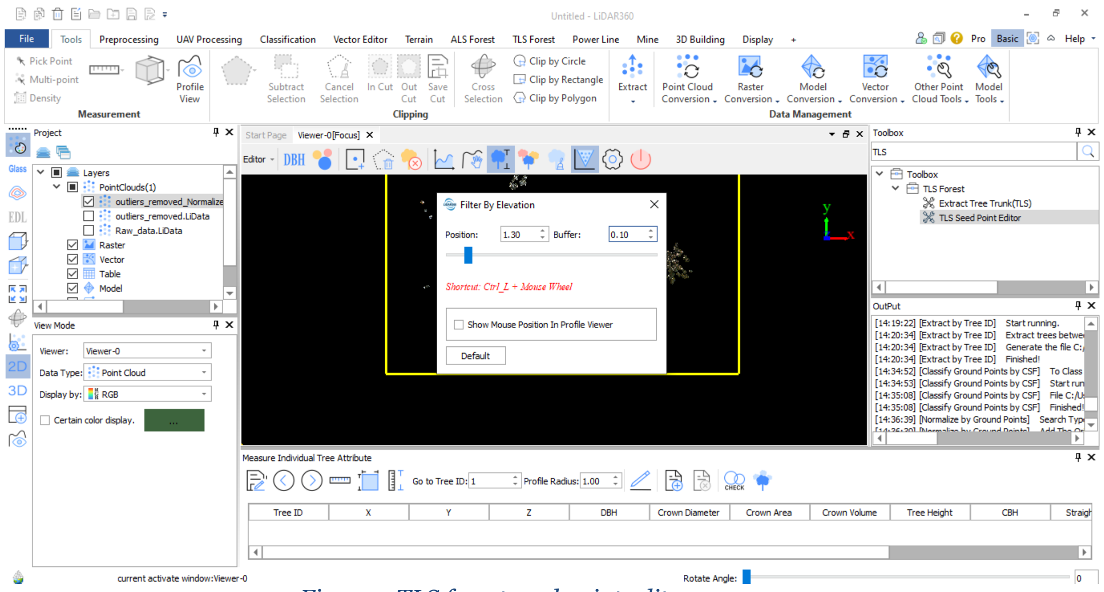


Figure 7 Normalized image

d. Click on 'TLS Forest' option from the toolbar. Click 'TLS Seed Point Editor'. Click on Editor> Start Edit. Select the normalized data file and click OK. 'Filter by elevation' window will open. Set buffer value to 0.1. Close the window.



Filter By Elevation

Position: 1.30 Buffer: 0.10

Shortcut: Ctrl_L + Mouse Wheel

☐ Show Mouse Position In Profile Viewer

Default

```
[14:19:22] [Extract by Tree ID]    Start running.
[14:20:34] [Extract by Tree ID]    Extract trees betwe
[14:20:34] [Extract by Tree ID]    Generate the file C:/
[14:20:34] [Extract by Tree ID]    Finished!
[14:34:52] [Classify Ground Points by CSF]    To Class
[14:34:53] [Classify Ground Points by CSF]    Start run
[14:35:08] [Classify Ground Points by CSF]    File C:/U
[14:35:08] [Classify Ground Points by CSF]    Finished!
[14:36:39] [Normalize by Ground Points]    Search Typ
```

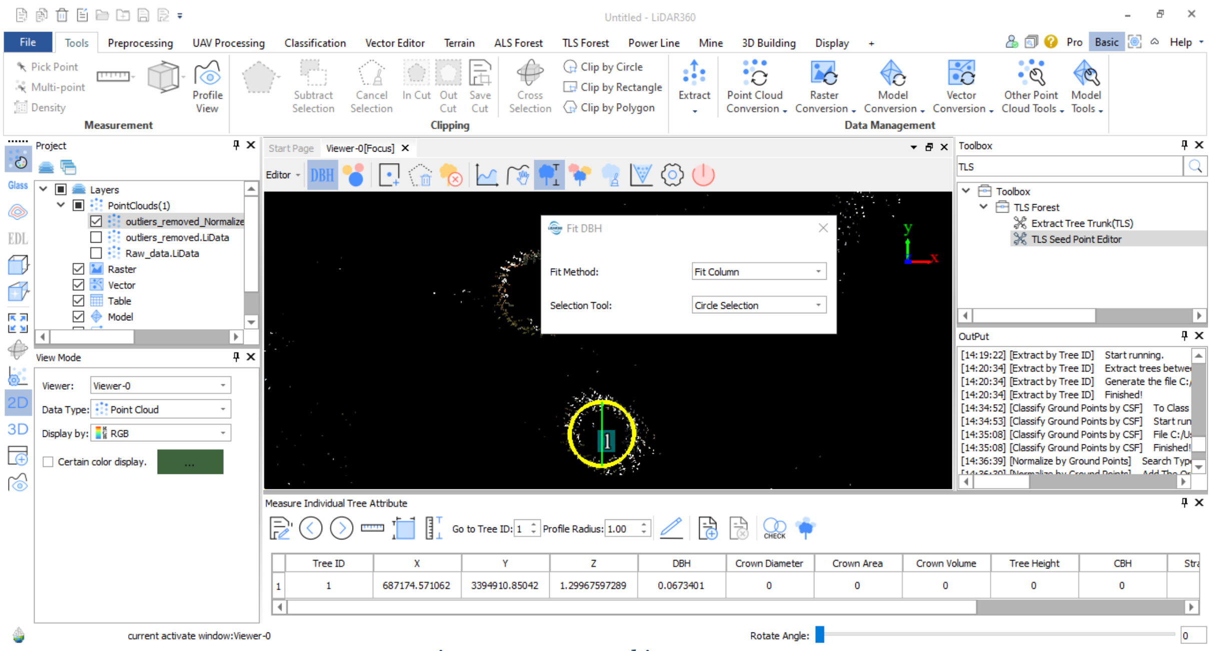
Measure Individual Tree Attribute

Go to Tree ID: 1 Profile Radius: 1.00

Tree ID	X	Y	Z	DBH	Crown Diameter	Crown Area	Crown Volume	Tree Height	CBH	Straigh

Figure 9 TLS forest seed point editor

e. Now you have to mark the seed points. Zoom upto the level so that trunk's circular shape can be detected. Click from TLS seed point editor toolbar 'Fit DBH' and fit a circle with the help of mouse so that all trunk points comes under this circle. Do same for all the trees.


Fit DBH
Fit Method: Fit Column
Selection Tool: Circle Selection

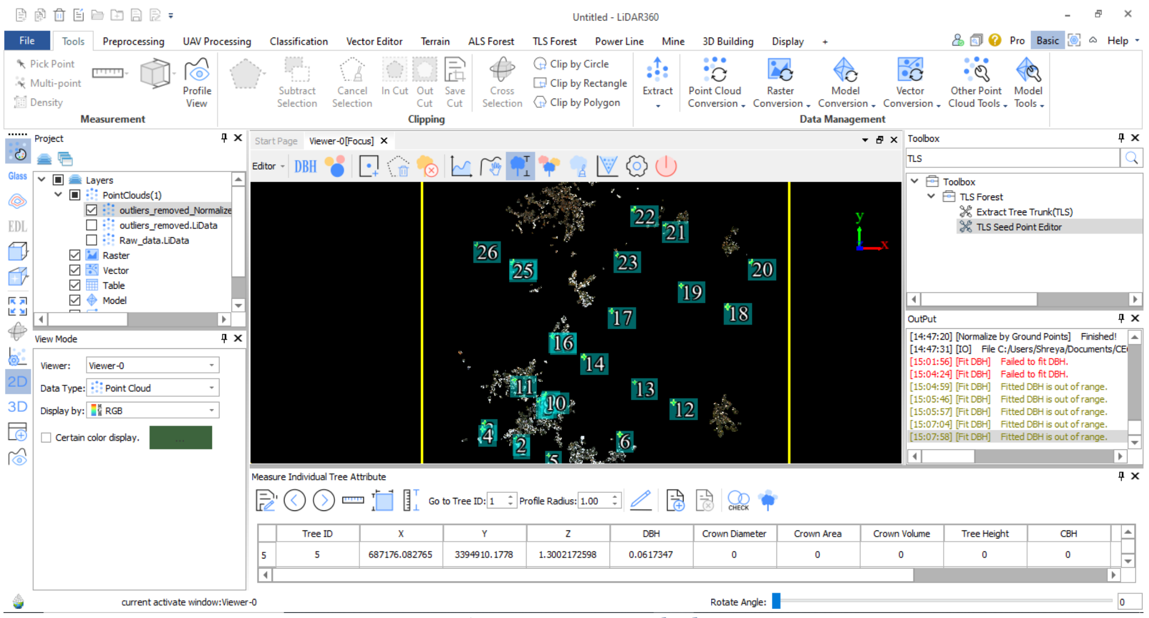
```
[14:19:22] [Extract by Tree ID]   Start running.
[14:20:34] [Extract by Tree ID]   Extract trees betwe
[14:20:34] [Extract by Tree ID]   Generate the file C:/
[14:20:34] [Extract by Tree ID]   Finished!
[14:34:52] [Classify Ground Points by CSF]   To Class
[14:34:53] [Classify Ground Points by CSF]   Start run
[14:35:08] [Classify Ground Points by CSF]   File C:/U
[14:35:08] [Classify Ground Points by CSF]   Finished!
[14:36:39] [Normalize by Ground Points]   Search Typ
```

Measure Individual Tree Attribute
Go to Tree ID: 1 Profile Radius: 1.00

	Tree ID	X	Y	Z	DBH	Crown Diameter	Crown Area	Crown Volume	Tree Height	CBH	Stra
1	1	687174.571062	3394910.85042	1.29967597289	0.0673401	0	0	0	0	0	

current activate window:Viewer-0
Rotate Angle: 0

Figure 10 Tree marking


```
[14:47:20] [Normalize by Ground Points]   Finished!
[14:47:31] [IO]   File C:/Users/Shreya/Documents/CE
[15:01:56] [Fit DBH]   Failed to fit DBH.
[15:04:24] [Fit DBH]   Failed to fit DBH.
[15:04:59] [Fit DBH]   Fitted DBH is out of range.
[15:05:46] [Fit DBH]   Fitted DBH is out of range.
[15:05:57] [Fit DBH]   Fitted DBH is out of range.
[15:07:04] [Fit DBH]   Fitted DBH is out of range.
[15:07:58] [Fit DBH]   Fitted DBH is out of range.
```

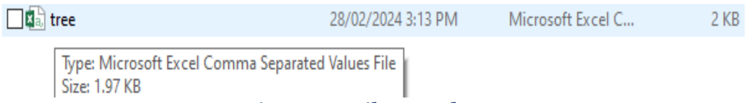
Measure Individual Tree Attribute
Go to Tree ID: 1 Profile Radius: 1.00

	Tree ID	X	Y	Z	DBH	Crown Diameter	Crown Area	Crown Volume	Tree Height	CBH
5	5	687176.082765	3394910.1778	1.3002172598	0.0617347	0	0	0	0	0

current activate window:Viewer-0
Rotate Angle: 0

Figure 11Trees marked

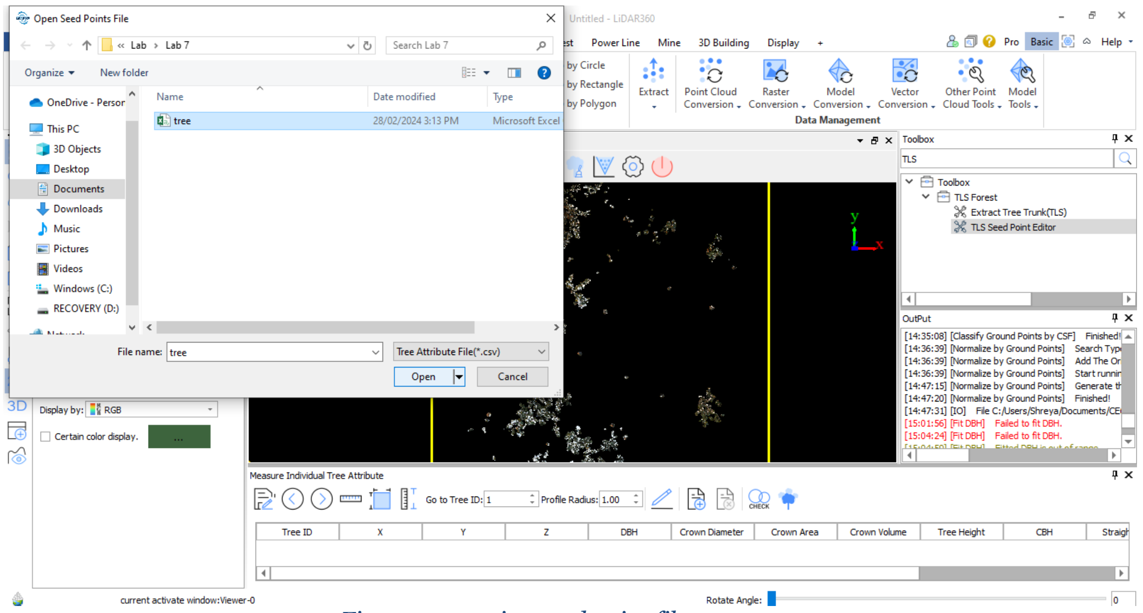
f. Now go to Editor> End Edit. It will ask you to save the seed points file. Click Yes and save the seed points file.


tree 28/02/2024 3:13 PM Microsoft Excel C... 2 KB

Type: Microsoft Excel Comma Separated Values File
Size: 1.97 KB

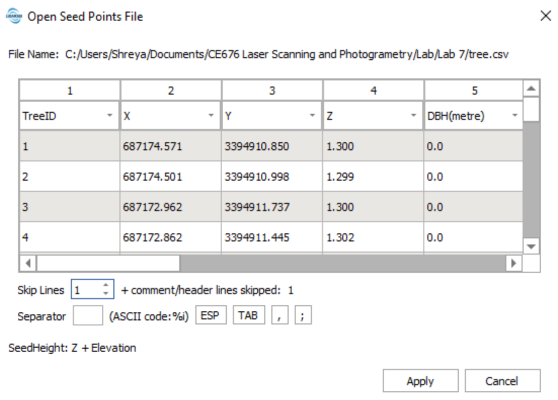
Figure 12 File saved

g. Go to Editor> Start Edit> Select the normalized data> Close the 'Filter by elevation' window. Click Editor> Open Seed Points File. Chose the seed points file. Change the view type to .csv and open the file. Set Skip lines equal to 1. Click Apply.



```
[14:35:08] [Classify Ground Points by CSF]    Finished!
[14:36:39] [Normalize by Ground Points]    Search Typ
[14:36:39] [Normalize by Ground Points]    Add The Or
[14:36:39] [Normalize by Ground Points]    Start runnin
[14:47:15] [Normalize by Ground Points]    Generate th
[14:47:20] [Normalize by Ground Points]    Finished!
[14:47:31] [IO]    File C:/Users/Shreya/Documents/CE
[15:01:56] [Fit DBH]    Failed to fit DBH.
[15:04:24] [Fit DBH]    Failed to fit DBH.
```

Figure 13 opening seed point file



File Name: C:/Users/Shreya/Documents/CE676 Laser Scanning and Photogrammetry/Lab/Lab 7/tree.csv

1	2	3	4	5
TreeID	X	Y	Z	DBH(metre)
1	687174.571	3394910.850	1.300	0.0
2	687174.501	3394910.998	1.299	0.0
3	687172.962	3394911.737	1.300	0.0
4	687172.862	3394911.445	1.302	0.0

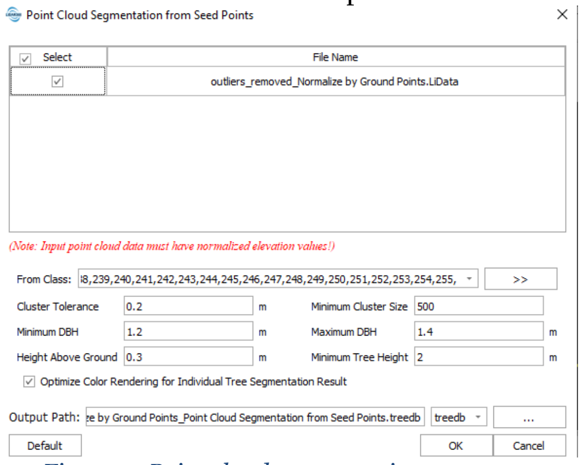
Skip Lines 1 + comment/header lines skipped: 1

Separator (ASCII code:%i) ESP TAB , ;

SeedHeight: Z + Elevation

Figure 14 open seed file editor

h. Click Point Cloud Segmentation from Seed Points. Select the normalized data and set parameters as default. Click OK. Exit the TLS seed point Editor.



Select	File Name
✓	outliers_removed_Normalize by Ground Points.LiData

(Note: Input point cloud data must have normalized elevation values!)

From Class:]8,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,

Cluster Tolerance	0.2	m	Minimum Cluster Size	500	
Minimum DBH	1.2	m	Maximum DBH	1.4	m
Height Above Ground	0.3	m	Minimum Tree Height	2	m

✓ Optimize Color Rendering for Individual Tree Segmentation Result

Output Path: e by Ground Points_Point Cloud Segmentation from Seed Points.treedb treedb

Figure 15 Point cloud segmentation

i. You can see the results by changing Display by to 'Tree ID'. Each tree has its own Tree ID but you can see that some of the crowns or ground points, etc may be mixed due to the limitation of algorithm and the parameters we chose.

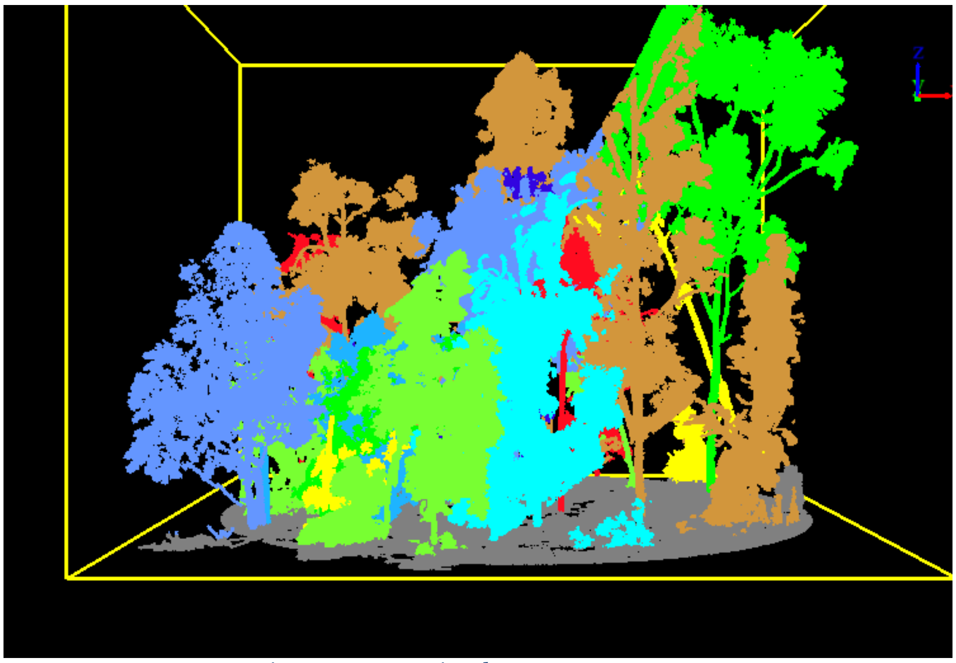
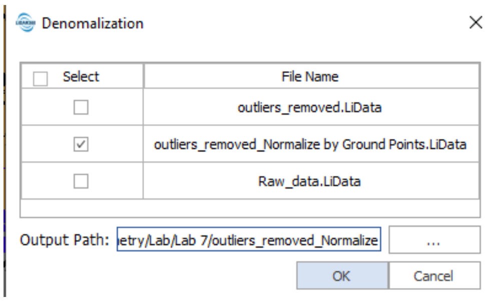


Figure 16 Trees view by tree ID

j. Denormalize the point cloud by clicking on TLS Forest> TLS Workflow Tools> Denormalization. Select the normalized data and click OK.


Denomalization

Select	File Name
☐	outliers_removed.LiData
✓	outliers_removed_Normalize by Ground Points.LiData
☐	Raw_data.LiData

Output Path: etry/Lab/Lab 7/outliers_removed_Normalize

Figure 17 Denormalization toolbox

k. Add the Denormalized data.

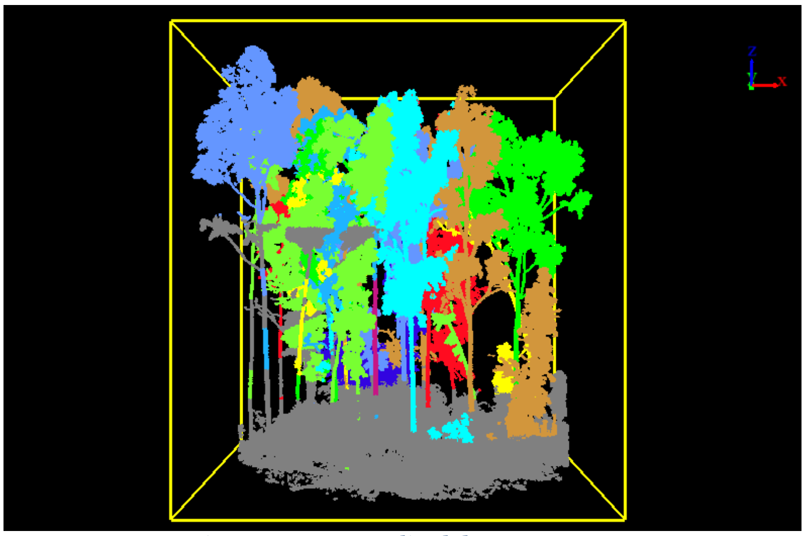
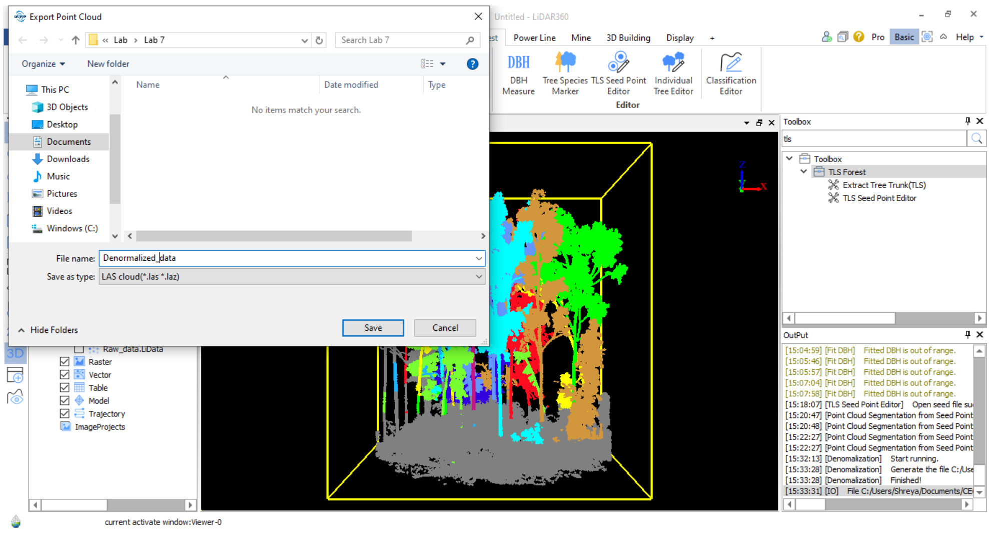


Figure 18 Denormalized data

l. Export the data by right clicking on the Denormalized data and click on Export. Give output path and click Save.



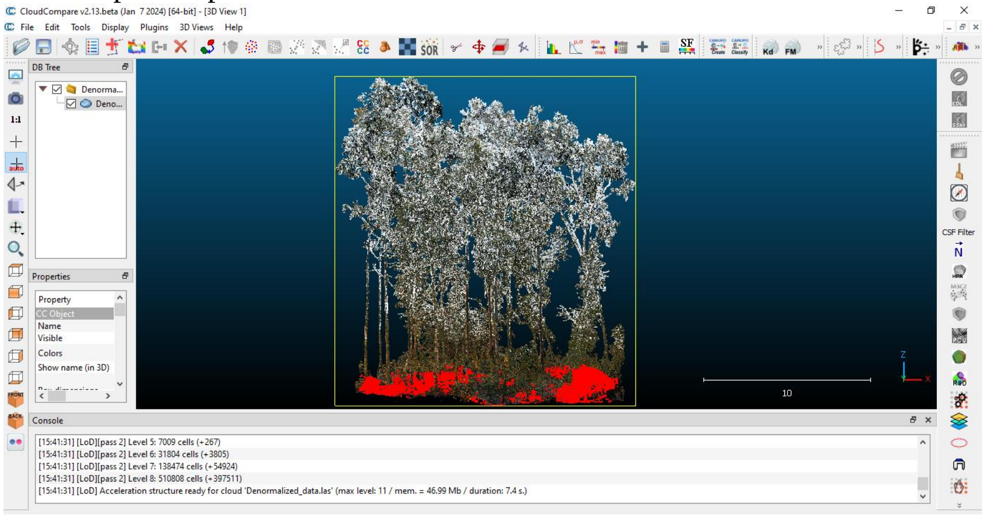
File name: Denormalized_data
Save as type: LAS cloud(*.las *.laz)

OutPut
```
[15:04:59] [Fit DBH]    Fitted DBH is out of range.
[15:05:46] [Fit DBH]    Fitted DBH is out of range.
[15:05:57] [Fit DBH]    Fitted DBH is out of range.
[15:07:04] [Fit DBH]    Fitted DBH is out of range.
[15:07:58] [Fit DBH]    Fitted DBH is out of range.
[15:18:07] [TLS Seed Point Editor]    Open seed file su
[15:20:47] [Point Cloud Segmentation from Seed Point
[15:20:48] [Point Cloud Segmentation from Seed Point
[15:22:27] [Point Cloud Segmentation from Seed Point
[15:22:27] [Point Cloud Segmentation from Seed Point
[15:32:13] [Denomalization]    Start running.
[15:33:28] [Denomalization]    Generate the file C:/Use
[15:33:28] [Denomalization]    Finished!
[15:33:31] [IO]    File C:/Users/Shreya/Documents/CE
```

Figure 19 Saving the denormalized data

B. Tree Segmentation in Cloud Compare

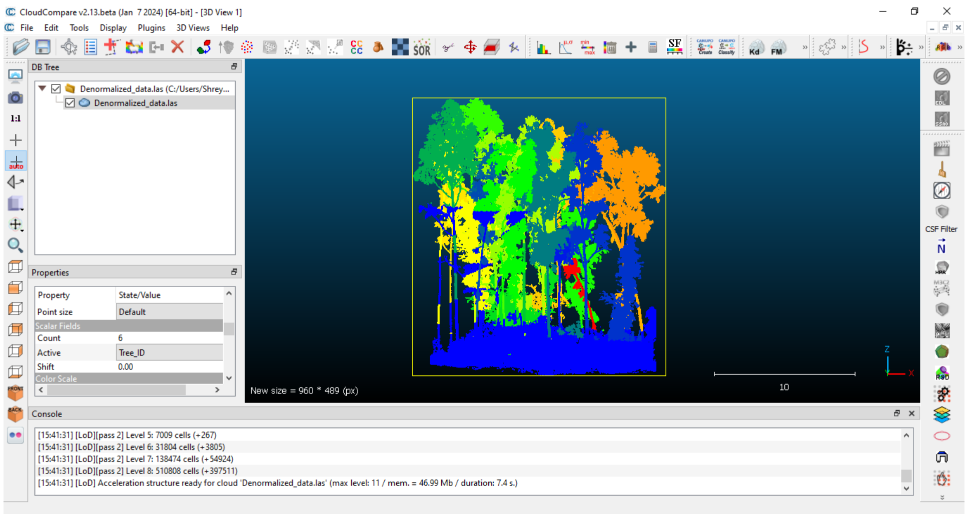
a. Install CloudCompare. Open the Denormalized classified data saved as .las or .laz file.



Console
```
[15:41:31] [LoD][pass 2] Level 5: 7009 cells (+267)
[15:41:31] [LoD][pass 2] Level 6: 31804 cells (+3805)
[15:41:31] [LoD][pass 2] Level 7: 138474 cells (+54924)
[15:41:31] [LoD][pass 2] Level 8: 510808 cells (+397511)
[15:41:31] [LoD] Acceleration structure ready for cloud 'Denormalized_data.las' (max level: 11 / mem. = 46.99 Mb / duration: 7.4 s.)
```

Figure 20 Denormalized file in cloud compare

b. Chose Active Scalar Field as Tree ID.


CloudCompare v2.13.beta (Jan 7 2024) [64-bit] - [3D View 1]
File
Edit
Tools
Display
Plugins
3D Views
Help
DB Tree
Denormalized_data.las (C:/Users/Shrey...
Denormalized_data.las
Properties
CSF Filter
10

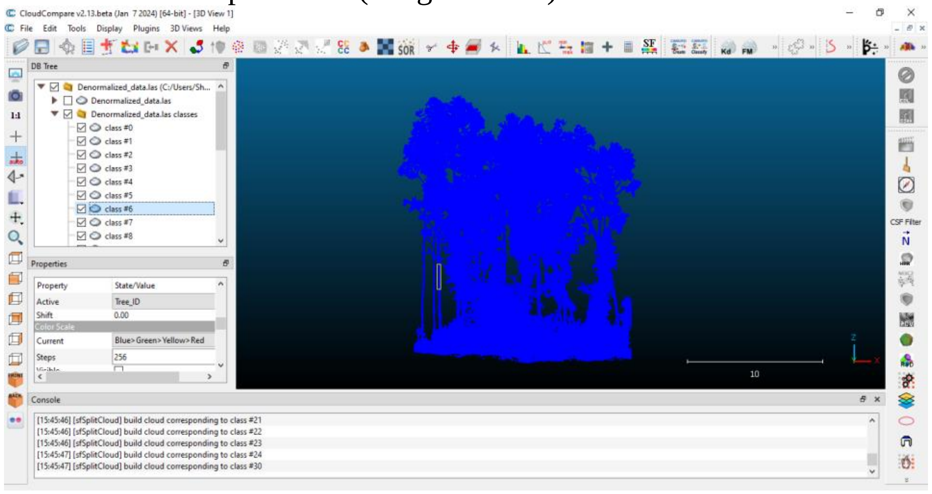
Property	State/Value
Point size	Default
Scalar Fields	
Count	6
Active	Tree_ID
Shift	0.00
Color Scale	

New size = 960 * 489 (px)

Console
```
[15:41:31] [LoD][pass 2] Level 5: 7009 cells (+267)
[15:41:31] [LoD][pass 2] Level 6: 31804 cells (+3805)
[15:41:31] [LoD][pass 2] Level 7: 138474 cells (+54924)
[15:41:31] [LoD][pass 2] Level 8: 510808 cells (+397511)
[15:41:31] [LoD] Acceleration structure ready for cloud 'Denormalized_data.las' (max level: 11 / mem. = 46.99 Mb / duration: 7.4 s.)
```

Figure 21 view by tree ID

c. Go to Edit> Scalar Fields> Split cloud (integer values).



```
[15:45:46] [sfSplitCloud] build cloud corresponding to class #21
[15:45:46] [sfSplitCloud] build cloud corresponding to class #22
[15:45:46] [sfSplitCloud] build cloud corresponding to class #23
[15:45:47] [sfSplitCloud] build cloud corresponding to class #24
[15:45:47] [sfSplitCloud] build cloud corresponding to class #30
```

Figure 22 split point cloud

d. Now multiple point clouds will be generated corresponding to each tree ID.
e. Show only one point cloud at a time and segment one tree without any other tree crown or ground or any other feature present.
f. To do this, Chose segment tool.

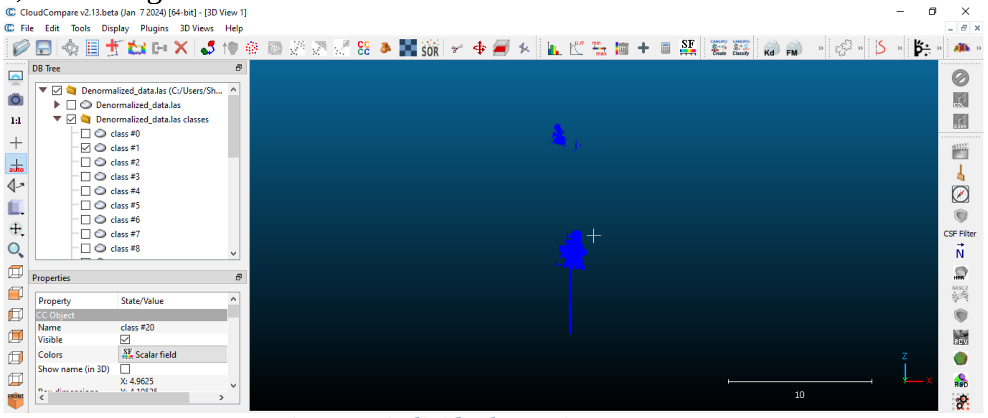
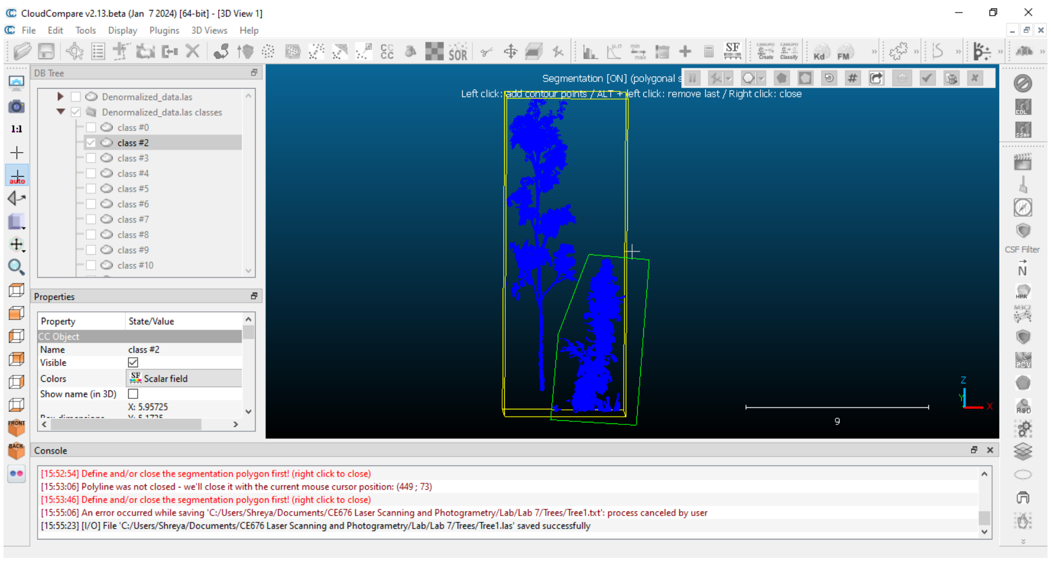


Figure 23 individual tree view

g. Form a polygon that is not a part of the tree and segment out it (by pressing O in keyboard). Pause and resume the segmentation with the help of space bar. Rotate the point cloud to have a complete view and segment it accordingly. Once all the unwanted features are removed, press Enter.



```
[15:52:54] Define and/or close the segmentation polygon first! (right click to close)
[15:53:06] Polyline was not closed - we'll close it with the current mouse cursor position: (449 ; 73)
[15:53:46] Define and/or close the segmentation polygon first! (right click to close)
[15:55:06] An error occurred while saving 'C:/Users/Shreya/Documents/CE676 Laser Scanning and Photogrametry/Lab/Lab 7/Trees/Tree1.txt': process canceled by user
[15:55:23] [I/O] File 'C:/Users/Shreya/Documents/CE676 Laser Scanning and Photogrametry/Lab/Lab 7/Trees/Tree1.las' saved successfully
```

Figure 24 Segmenting

h. Repeat step 8 for all other trees.
i. Once it is done for all trees, save corresponding files.

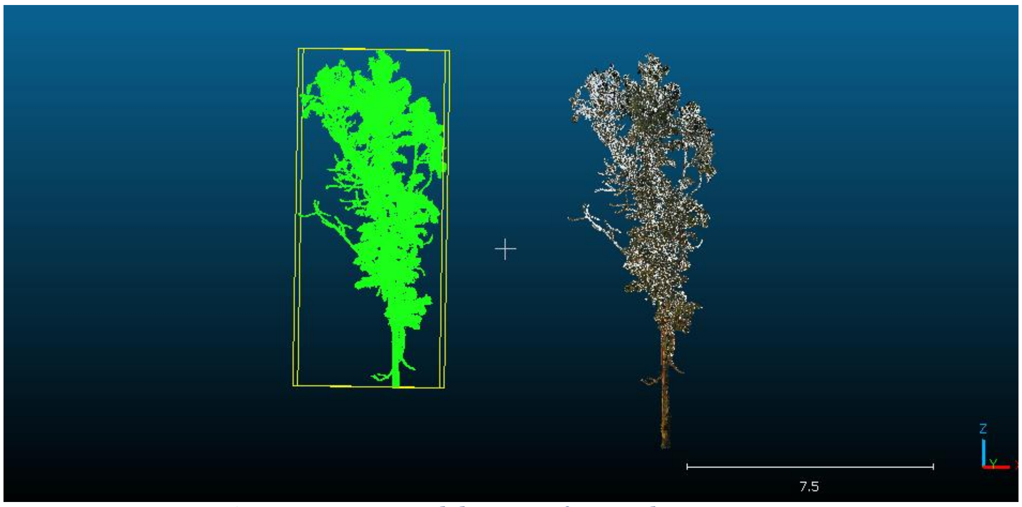


Figure 25 Segmented data vs reference data

C. Labelling of Data

In this analysis, the data is labeled by associating each point cloud dataset with a specific tree type or category based on its source. For instance, datasets from different forest plots are assigned distinct labels to identify the type of tree they represent. Each file of point cloud data is tagged with a label corresponding to its tree type, such as 'Tree_22' for data from the plot labeled as 22, 'Tree_23' for plot 23, and so on. This is done with Python code.

Steps to Accomplish:

1. Preprocess the Data: Combine all the datasets into one and label them according to the tree (CSV file) they belong to.
2. Feature Engineering: Use the coordinates (x, y, z) as features.
3. Model Training: Train a clustering model (e.g., k-means) or a supervised classifier (e.g., Random Forest) to classify points based on their features.
4. Model Testing: Use the trained model to predict the tree labels for a new set of points.
5. Evaluation: Compare the predicted labels with the true labels, highlight misclassified points, and create a confusion matrix.

D. Training the model

The model is developed using a Random Forest classifier, which is well-suited for handling complex datasets like those from Terrestrial Laser Scanning. Initially, the test dataset, having multiple labeled point cloud datasets, is divided into training and testing subsets. The Random Forest classifier is trained on the training data, which includes features such as the spatial coordinates (X, Y, Z) of each point and their corresponding labels. This training phase allows the model to learn the relationships between these features and the tree types. Once the model is trained, it is evaluated using the test data, which consists of previously unseen point cloud data. Performance metrics such as confusion matrices and classification reports are generated to quantify the model's effectiveness. These metrics reveal how well the model classifies each tree type and highlight any misclassifications. The evaluation process ensures that the model not only performs well on the training data but also generalizes effectively to new, real-world data.

4. Discussion

a. What is CSF filter? How the results will be affected when we change the grid size in CSF filter. The method is based on the simulation of a simple physical process. Imagine a piece of cloth is placed above a terrain, and then this cloth drops because of gravity. Assuming that the cloth is soft enough to stick to the surface, the final shape of the cloth is the DSM (digital surface model). However, if the terrain is firstly turned upside down and the cloth is defined with rigidness, then the final shape of the cloth is the DTM. To simulate this physical process, we employ a technique that is called cloth simulation. Based on this technique,the cloth simulation filtering (CSF) algorithm works to extract ground points from LiDAR points. First, the original point cloud is turned upside down, and then a cloth drops to the inverted surface from above. By analyzing the interactions between the nodes of the cloth and the corresponding LiDAR points, the final shape of the cloth can be determined and used as a base to classify the original points into ground and non-ground parts.

Cloth resolution refers to the grid size (the unit is same as the unit of point clouds) of cloth which is used to cover the terrain. The bigger cloth resolution you have set; the coarser DTM you will get.

b. What do you mean by normalization of data? Why we chose ‘Add the Original Z value as an Additional Attribute’ during the normalization process?

Normalization of the data was done to remove the ground points and only keep the above ground points in the file. Here Z value was used as an additional attribute in order to preserve the original data of the image, to normalize the data w.r.t. original elevation value. This statistical normalization ensures that different features are on a common scale

c. We have cut the data at 1.3 m from ground for seed point selection. Why we chose this height? Which morphological parameter is calculated at that height?
From the normalized data, the data above 1.3 m is removed as the foliage will be removed and the trunk of the tree will be visible for further selection. 1.3 meters from the ground is often considered a representative height for capturing the vegetation canopy. In many forested areas, this height corresponds to the lower branches and foliage of the trees. Focusing on this height allows for the characterization of the vegetation structure and density.

d. What is the significance of Precision and Recall? What other parameters can be calculated from confusion matrix.
Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positives. Precision is a measure of the accuracy of the positive predictions made by the model.
Recall: Recall is the ratio of correctly predicted positive observations to the total actual positives. Recall is a measure of the model's ability to capture all the positive instances in the dataset.

For accuracy analysis, following values are calculated:
Precision: Precision measures the accuracy of positive predictions. It indicates the proportion of correctly identified positive instances out of all instances predicted as positive. A high precision value implies that when the model predicts a positive class, it is likely to be correct.

$$Precision = \frac{TP}{TP+FP}$$

Recall: Recall, also known as sensitivity or true positive rate, measures the ability of a model to capture all relevant instances of a class. It represents the proportion of actual positive instances that were correctly identified by the model. A high recall value indicates that the model is effective at identifying most of the positive instances.

$$Recall = \frac{TP}{TP+FN}$$

Overall accuracy: Overall Accuracy provides a general measure of accuracy across all classes. It represents the proportion of correctly predicted instances (both true positives and true negatives) out of all instances. While overall accuracy is a useful metric, it may not capture class imbalances or the specific performance of the model on individual classes.

$$\textit{Overall Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

Kappa value: Kappa Value assesses the agreement between the model's predictions and the actual labels while accounting for the agreement that could occur by chance. A higher kappa value suggests a better agreement beyond what would be expected due to random chance, providing a more robust measure of performance.

$$Cohen's\ kappa = \frac{N\sum_{i=1}^{m} CM_{ii} - \sum_{i=1}^{m} Ci_{corr}Ci_{pred}}{N^2 - \sum_{i=1}^{m} Ci_{corr}Ci_{pred}}$$

5. Result

The following are the values obtained for accuracy:

Method	**Values**
Precision	0.9995
Recall	0.9929
Overall Accuracy	0.9999
Kappa Value	1

Overall Accuracy & Kappa:

- Accuracy: 1.00 (100%) - This indicates that your model correctly classified all the instances in the dataset.
- Cohen's Kappa: 1.00 - This perfect score suggests a high level of agreement between the predicted and true labels, beyond what would be expected by chance.

Classification Report:

- Precision, Recall, and F1-Score for each tree class are very high (close to 1.00), indicating that the model performs exceptionally well across all classes.
- The high precision and recall values imply that the model rarely misclassifies instances, and when it does, the errors are minima

Precision

- 0.999537 - This represents the overall precision of the model, accounting for the number of instances in each class. It reflects that the model is very precise overall.

Weighted Average:

- Precision, Recall, and F1-Score are identical to the macro average in this case, indicating that the performance metrics are consistent and not heavily influenced by the class distribution.

Confusion Matrix:

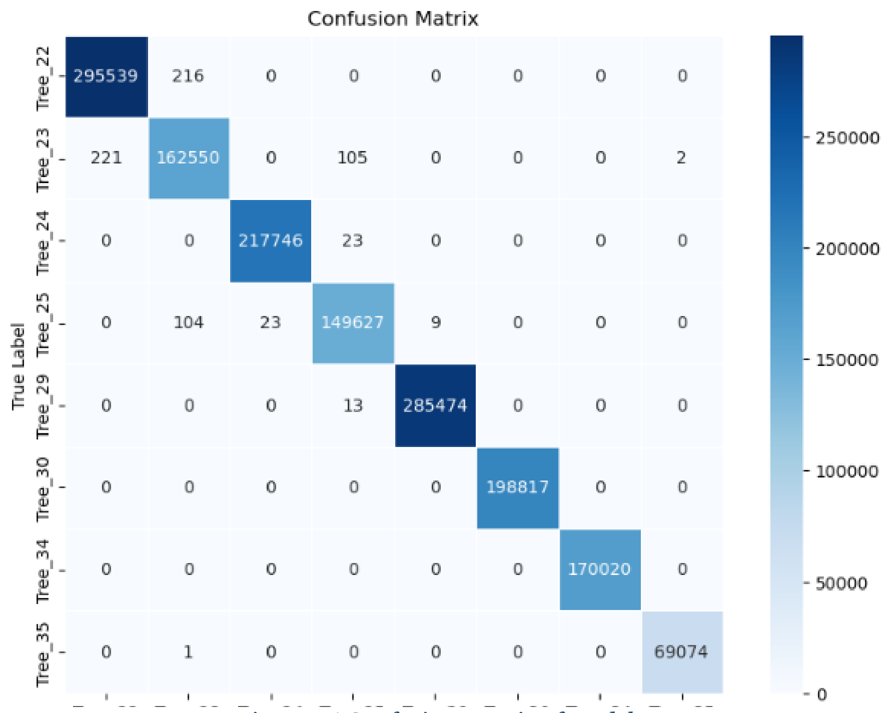


Figure 26 Confusion Matrix of model

Result for the
classification of segmented trees based on trained model:

1) The total percentage of Misclassified Lidar points in segmentation are: 0.019011

2) Label wise misclassification is as follows:

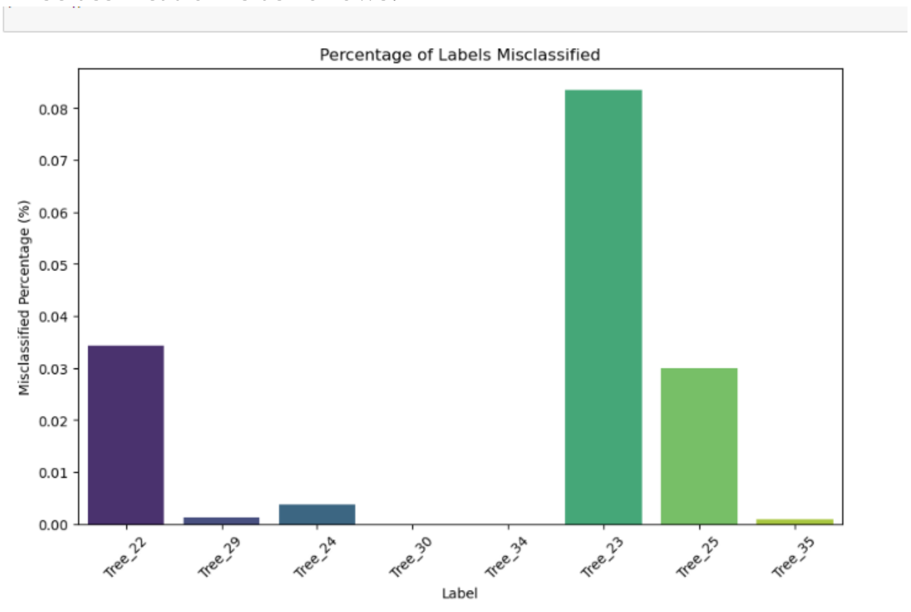


Figure 27 Graph for classification of segmented trees

6. Conclusion

In conclusion, this lab report has provided a hands-on experience with TLS data processing in forests, emphasizing the crucial step of individual tree segmentation. By exploring the methodologies and techniques involved in processing point cloud data, as well as the rationale behind specific parameter calculations at 1.3 meters from the ground, we have gained valuable insights into the application of TLS technology in forest monitoring. Understanding these processes is essential for harnessing the full potential of TLS data to inform sustainable forestry practices and contribute to our broader understanding of forest ecosystems. The comprehensive analysis reveals that the Random Forest classification model, developed through meticulous feature engineering and rigorous data preprocessing, achieved exceptional performance with an accuracy and Cohen's Kappa of 1.00. The precision, recall, and F1-score values, all approaching 1.00, underscore the model's high precision and effectiveness in classifying instances correctly across all classes. The detailed process of feature selection, data transformation, and model training has contributed to a robust, reliable, and accurate classification system, demonstrating the effectiveness of the Random Forest approach and the thoroughness of the data preparation.

7. References

[1] M. George Vosselman, Airborne and Laser Terrestrial Scanning.

[2] A. W. K.-H. Thiel, "PERFORMANCE CAPABILITIES OF LASER SCANNERS," *International archives of Photogrammatry, Remote sensing and spatial information science,* vol. 36.