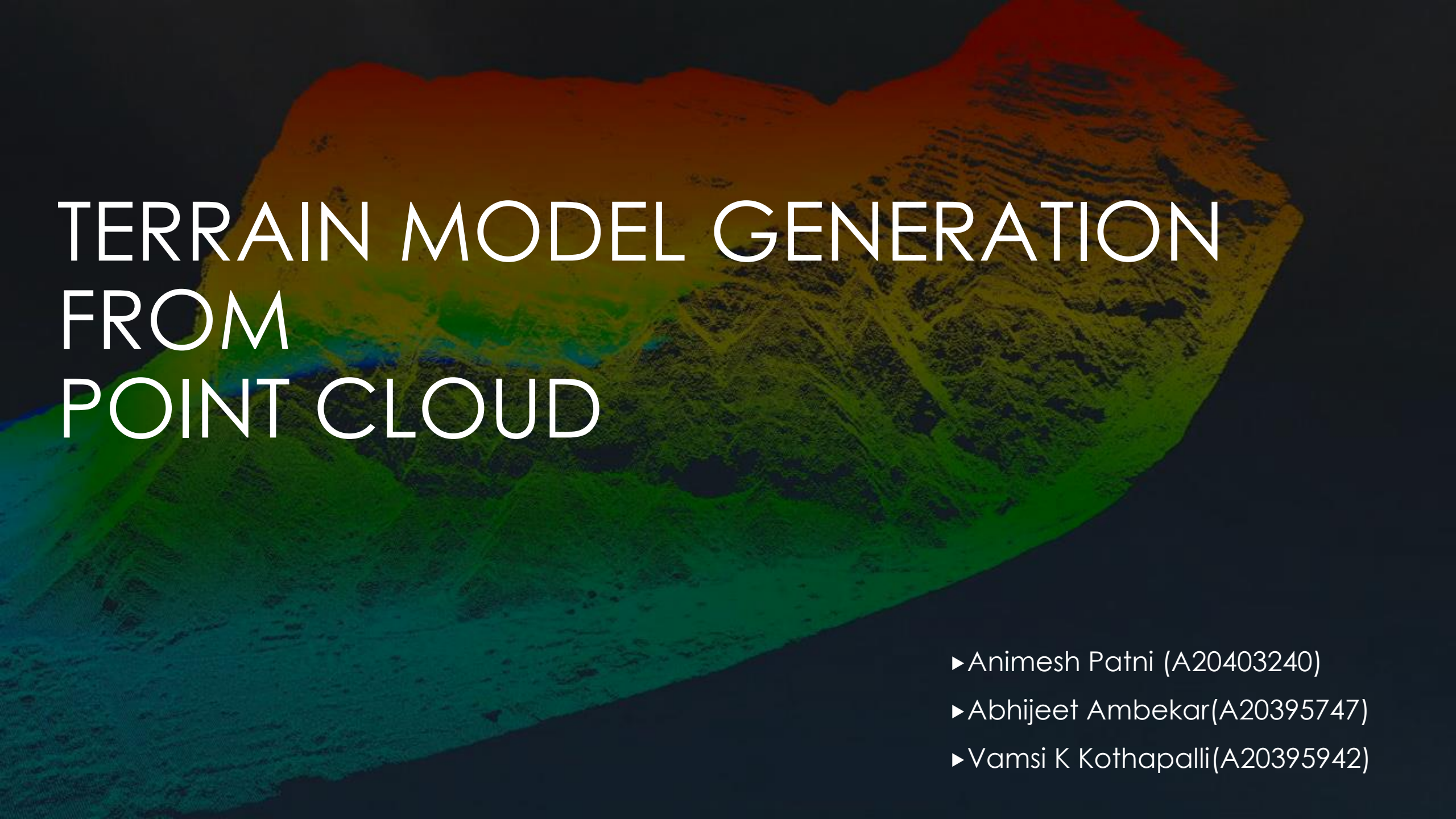
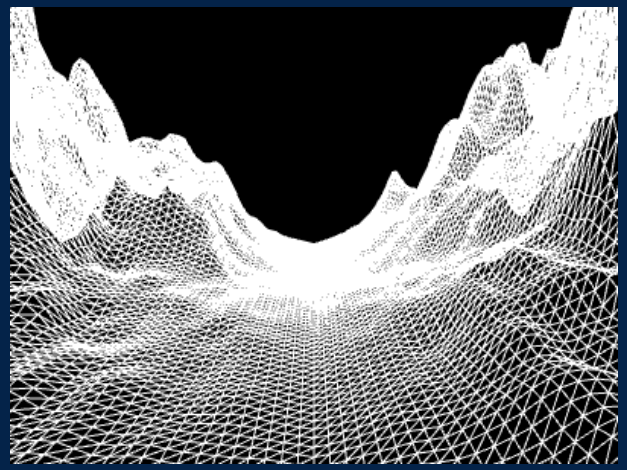


TERRAIN MODEL GENERATION FROM POINT CLOUD

- Animesh Patni (A20403240)
- Abhijeet Ambekar(A20395747)
- Vamsi K Kothapalli(A20395942)

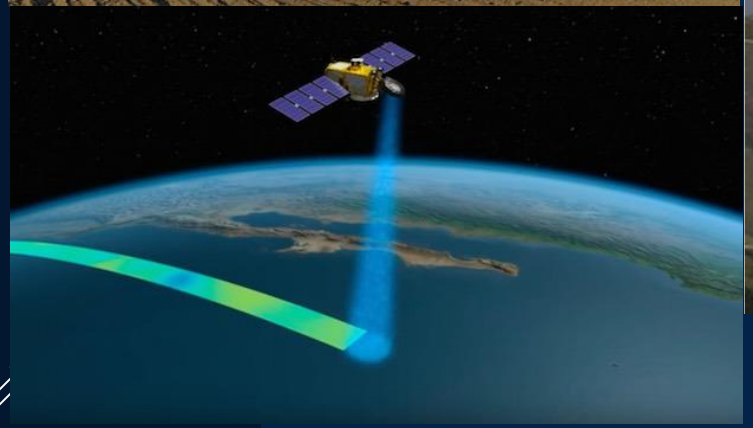
CONTENT

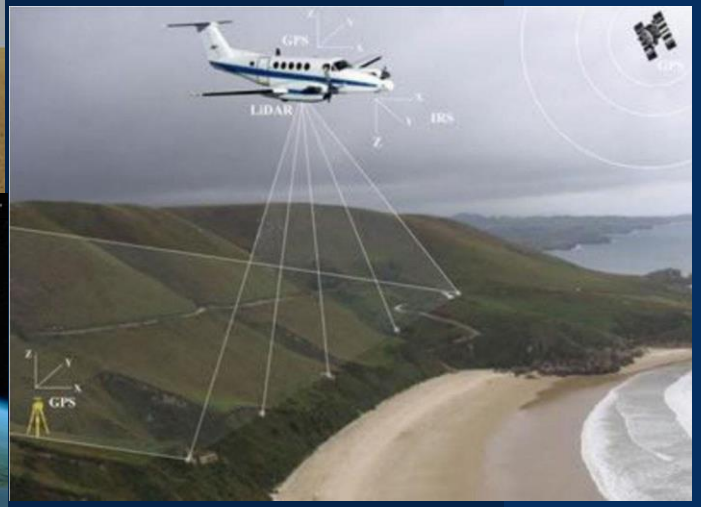
INTRODUCTION



- Point Cloud.
  - Set of data points in space
  - Various file format standards are available to store these points
  - Used for DTM, DSM and other spatial analysis
- Digital terrain model (DTM).
  - A visual representation of geographical spatial features.
  - Digital Elevation Model is a 3D representation of the terrain elevations.
  - Digital Terrain Model is a DEM in which data has been further enhanced with break lines or filtering out the noise.
  - Outliers are noise and Inliers are points that are needed

DATA ACQUISITION





- Satellite Mapping
- Stereo Vision
- Surveying and mapping drones
- LiDAR
- Physically touching the model
- Structured Light

DTM USE-CASES

- To build canals and design water flow for mass movement of water bodies
- It can also be used to study the natural disaster prone regions for flash floods
- In building Metro, Rail systems and even roads
- It can also be helpful to improve Navigation by building better maps
- Used in Flight simulation technology

POINT CLOUD PROCESSING

- Development Environment
  - Python
  - PCL
  - PPTK
- Coordinate Transformation
  - LLA to ECEF

COORDINATE TRANSFORMATIONS: LLA → ECEF

- Calculate ellipsoid flattening *f*.
- $f = \frac{a-b}{a}$
- Calculate eccentricity e.
- $e = \sqrt{f * (2 - f)}$
- Calculate the distance *N* from the surface to z- axis along ellipsoid normal as a function of Φ (geodetic latitude):
- $N(\Phi) = \frac{a}{\sqrt{1-e^2 sin^2(\Phi)}}$

- The ECEF coordinates *X*, *Y* and *Z* can be calculated from:
- $X = \left(h + N(\Phi)\right) \cos(\lambda) \cos(\Phi)$
- $Y = \left(h + N(\Phi)\right) \cos(\lambda) \sin(\Phi)$
- $Z = \left(h + (1 - e^2)N(\Phi)\right) \sin(\lambda)$

METHODOLOGY

VISUALIZING POINT CLOUD

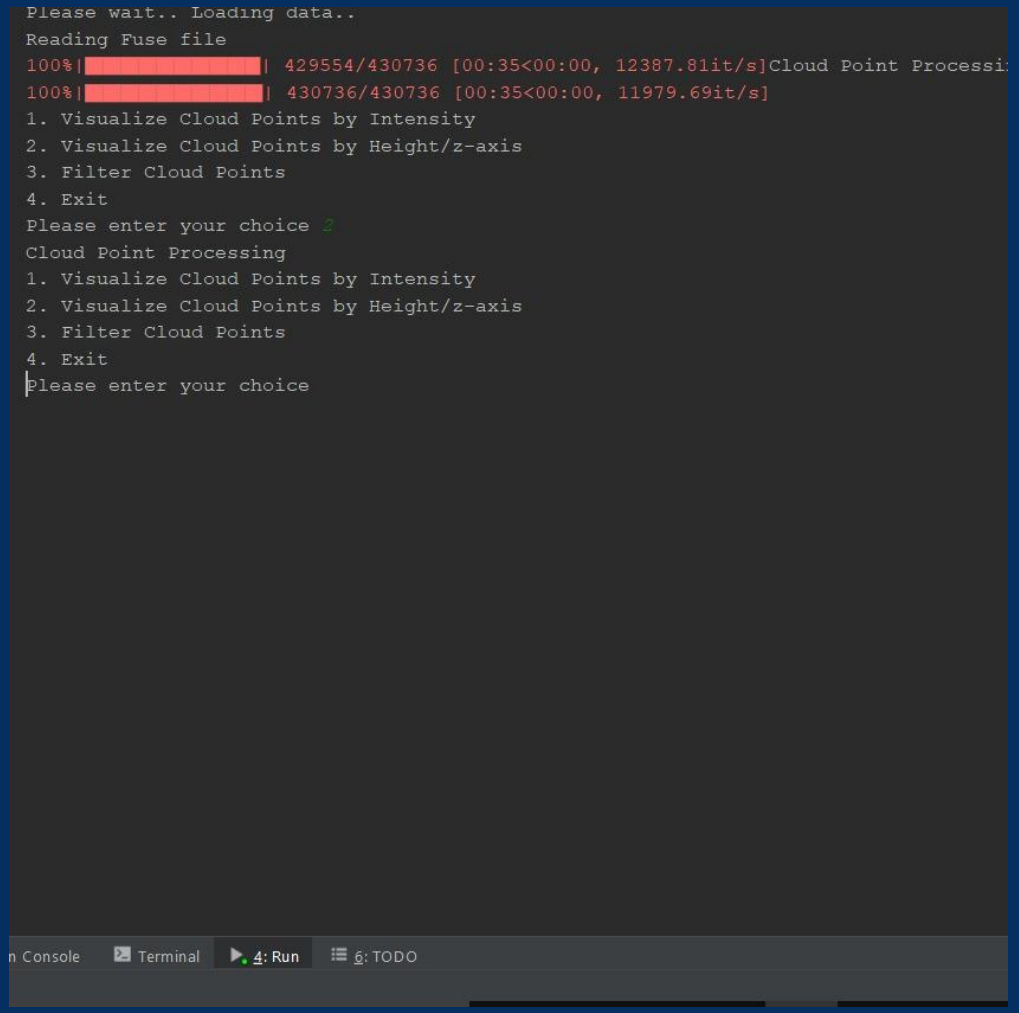
Read	**Read input file** • Latitude, longitude, altitude, intensity
Intensity Model	**Generate Terrain model by Intensity Data** • Convert <Lat, Long> to ECEF coordinates <X,Y,Z> • Visualize the cloud points <x,y,z,intensity>
Elevation Model	**Generate Terrain model by Elevation Data** • Convert <Lat, Long> to ECEF coordinates <X,Y,Z> • Visualize the cloud points <X,Y,Z> • Color the point clouds by the altitude data

FILTERING POINTS

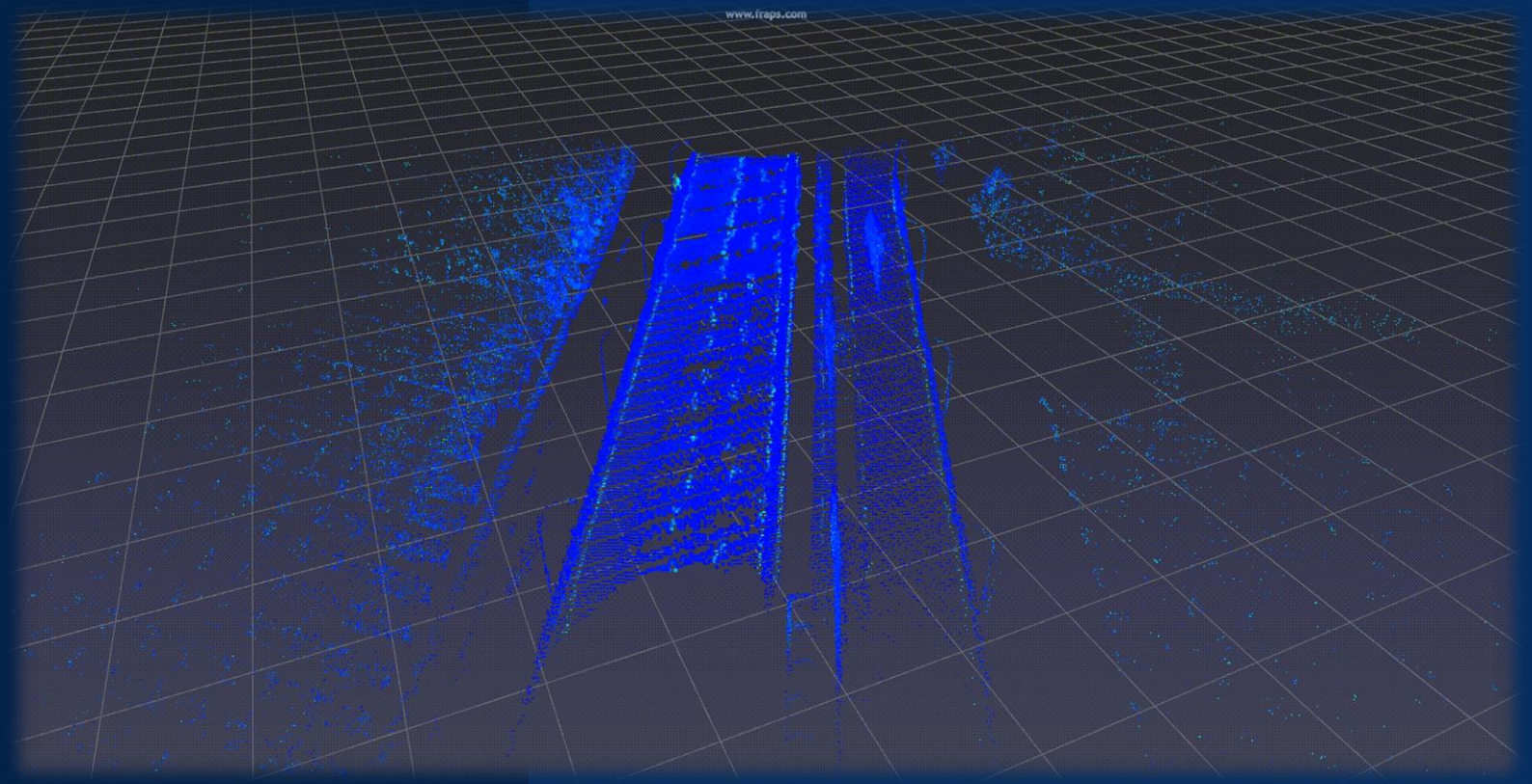
- Filtering Noise
  - Read the input file
  - Create point cloud by PCL library
  - Use Statistical Outlier Filter
  - Set the number of neighboring points for mean analysis (K)
  - Set standard deviation threshold
- Outliers
  - Point with mean distance > (mean * std_dev)
- Inliers
  - Point with mean distance < (mean * std_dev)
  - Generate Terrain Model

HOW TO RUN

- Python main.py
  - Cloud Point Processing
  - 1. Visualize Cloud Points by Intensity
  - 2. Visualize Cloud Points by Height/z-Axis
  - 3. Filter Cloud Points
  - 4. Exit
  - Please enter your choice

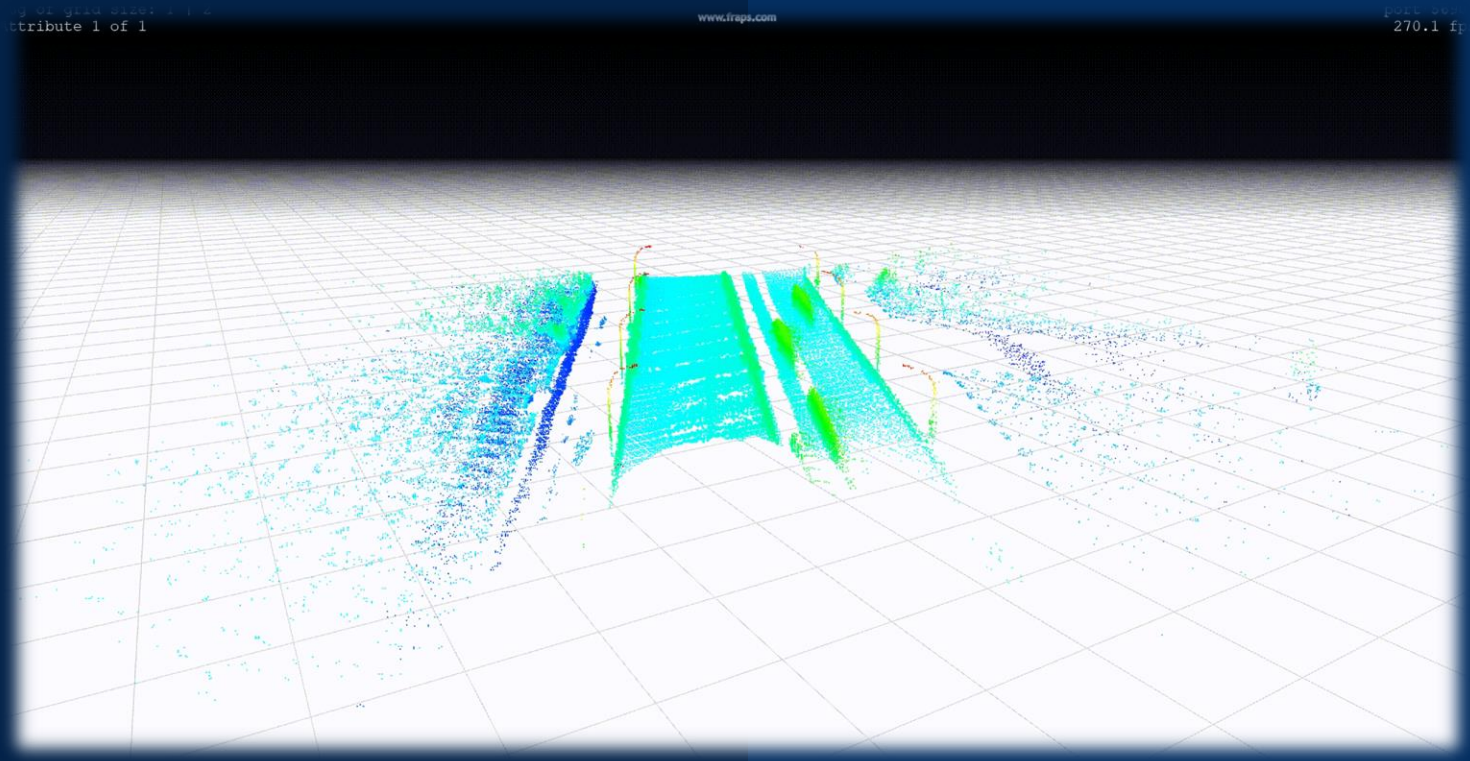


OUTPUT



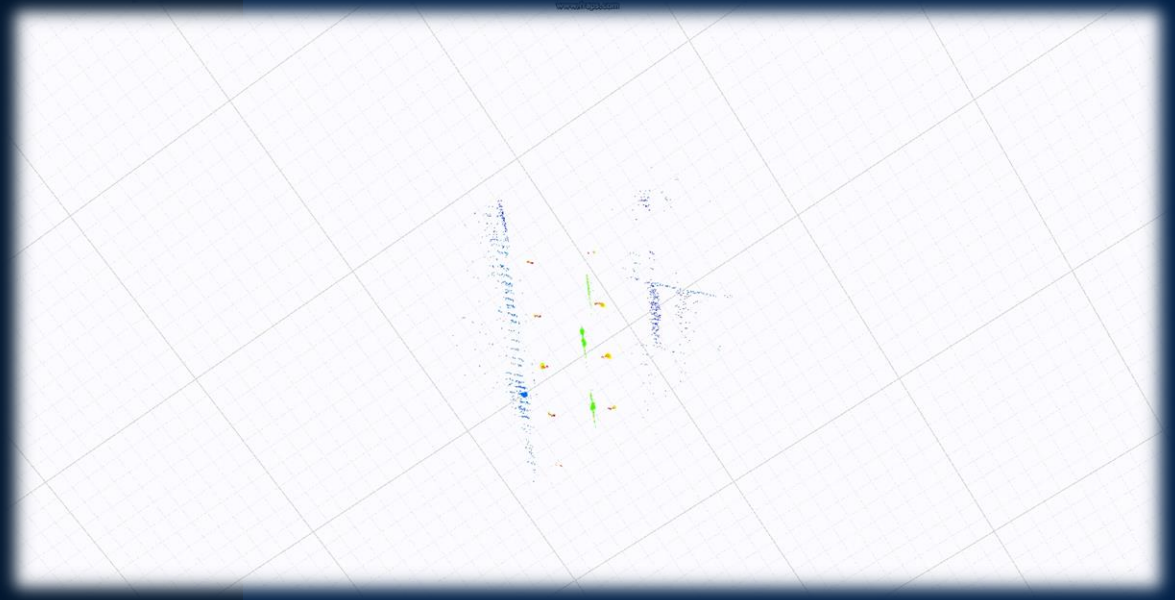
TERRAIN MODEL BY INTENSITY

OUTPUT

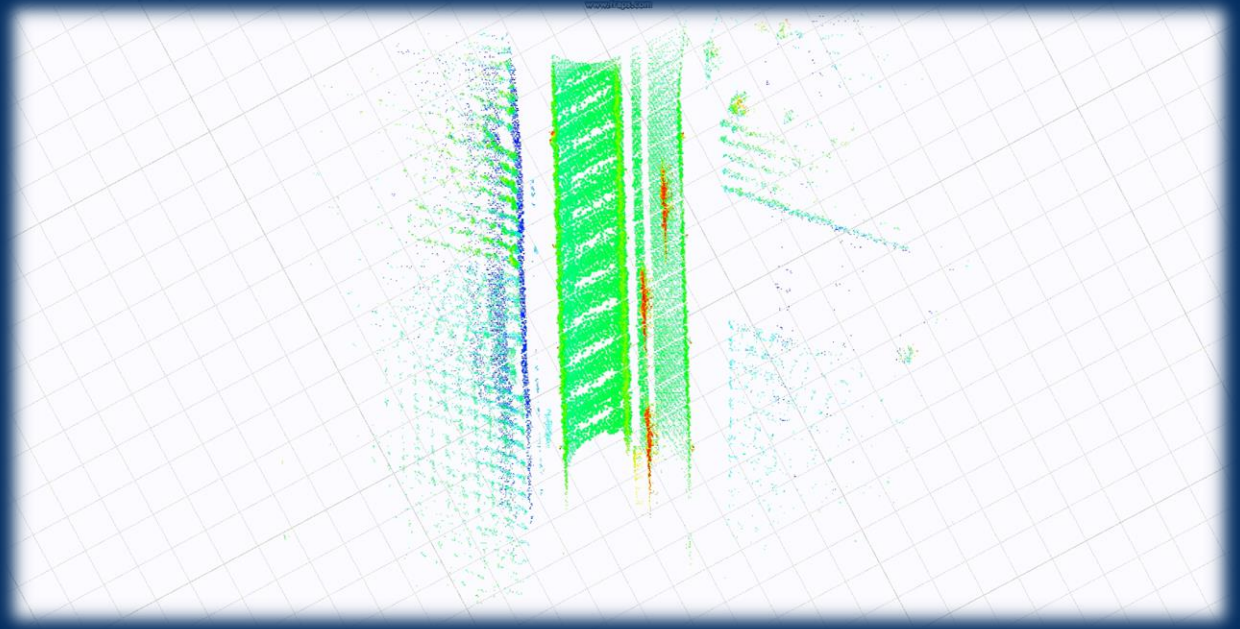


Terrain Model by Elevation Visualization

OUTPUT

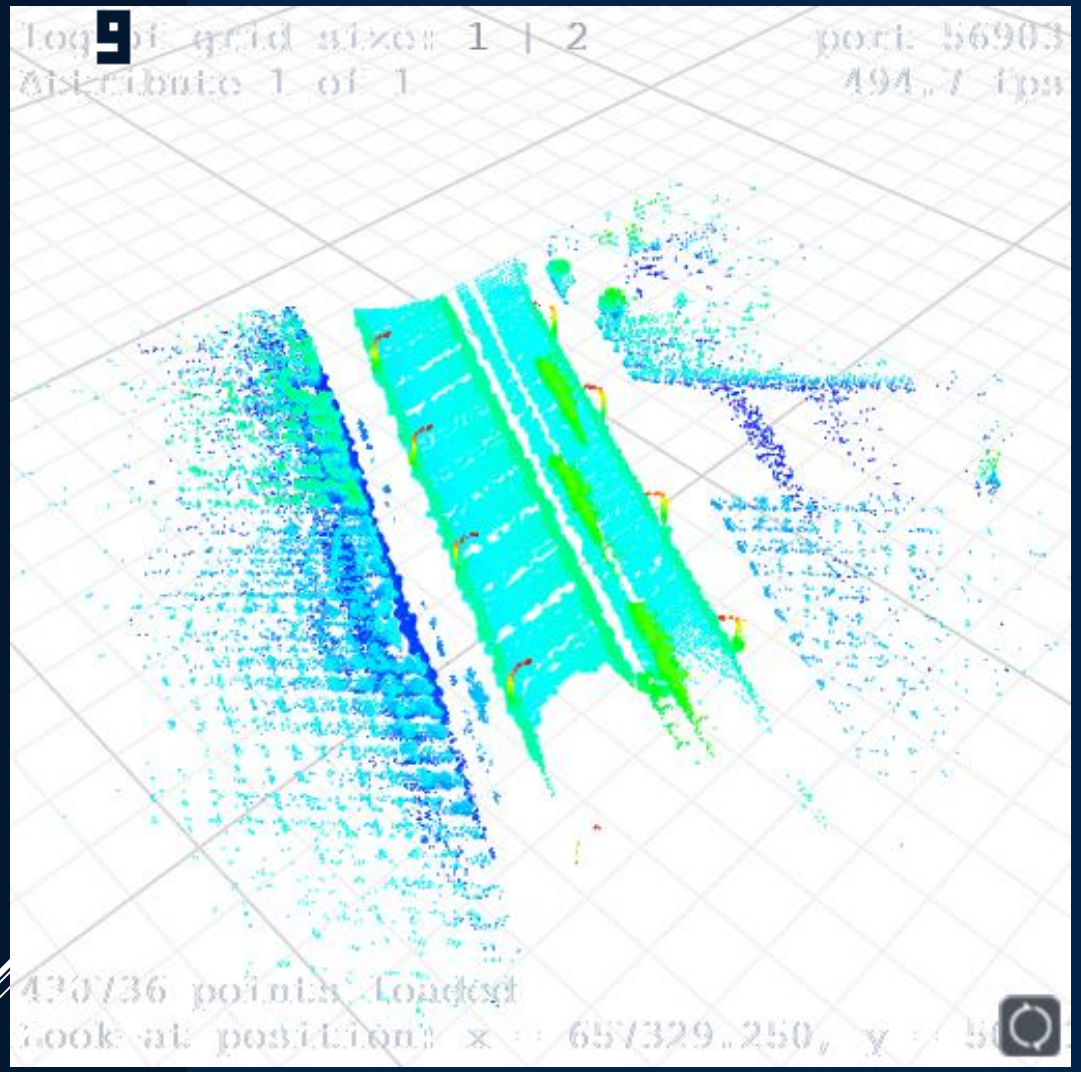


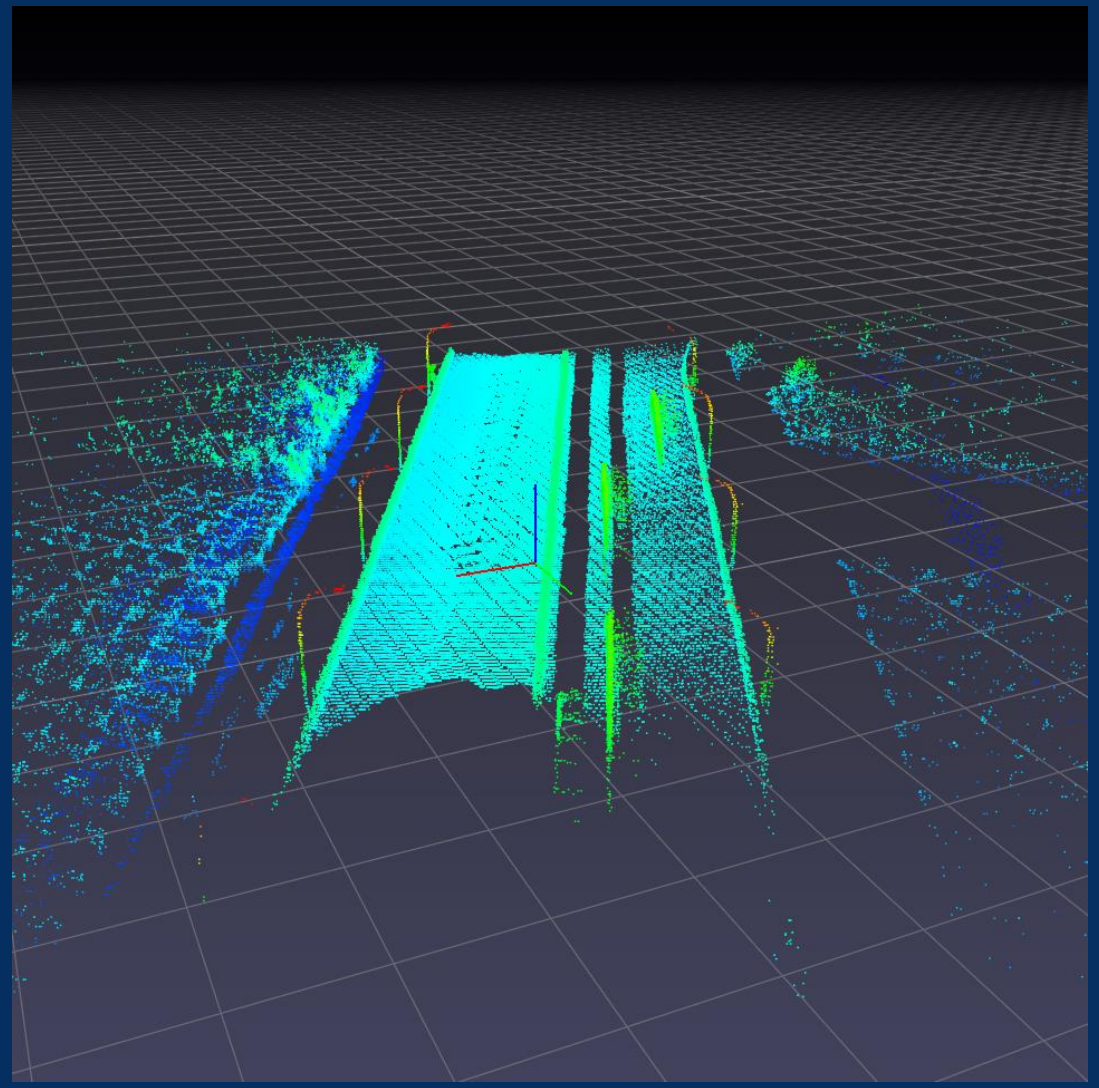
OUTLIERS - Visualization



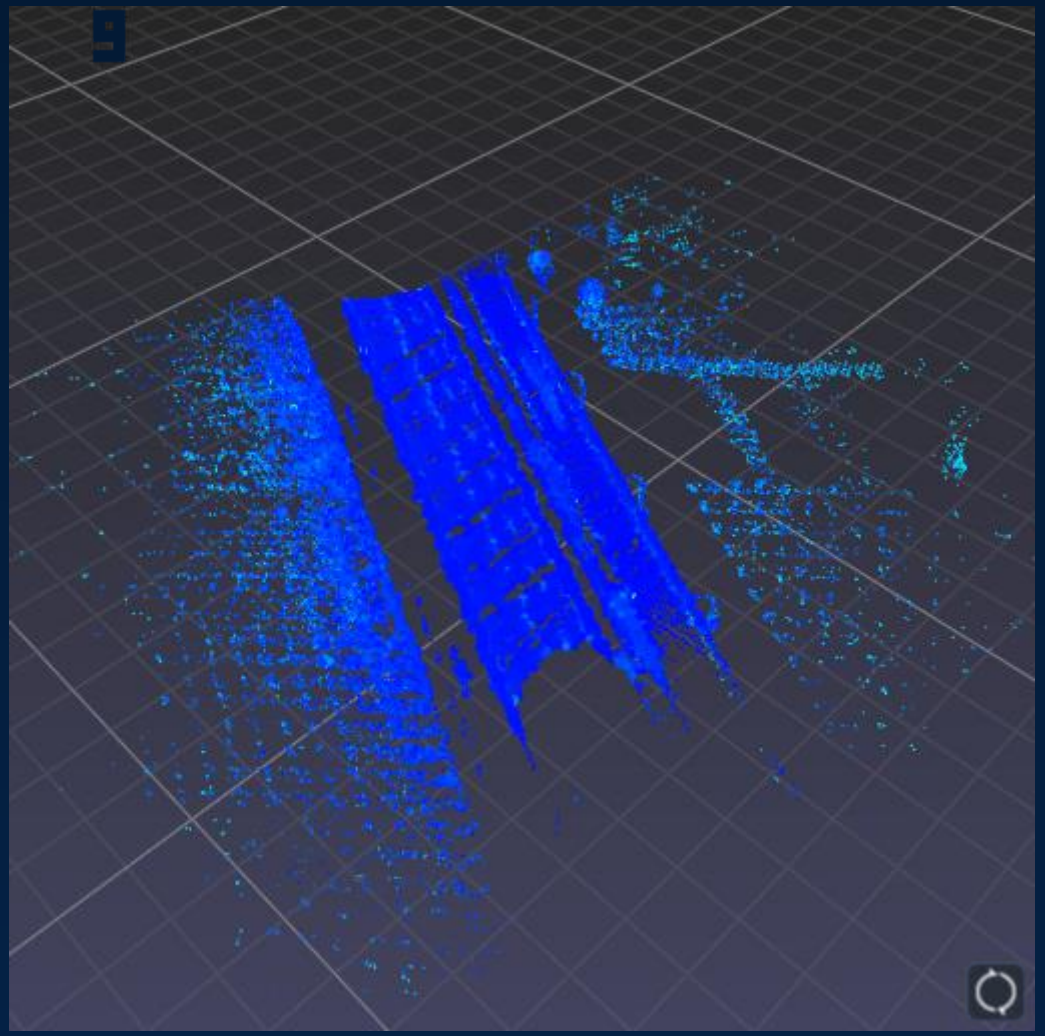
INLIERS - Visualization

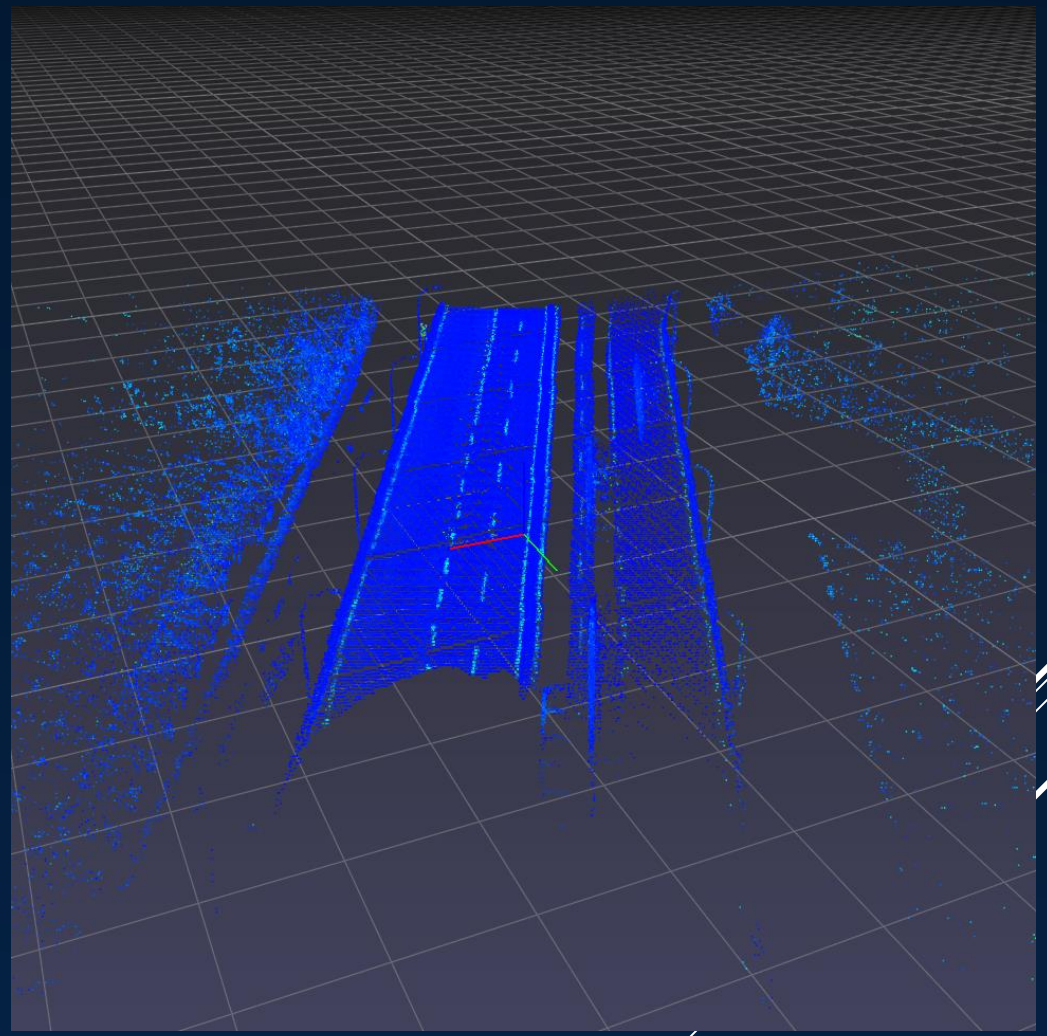
TERRAIN MODEL BY ELEVATION





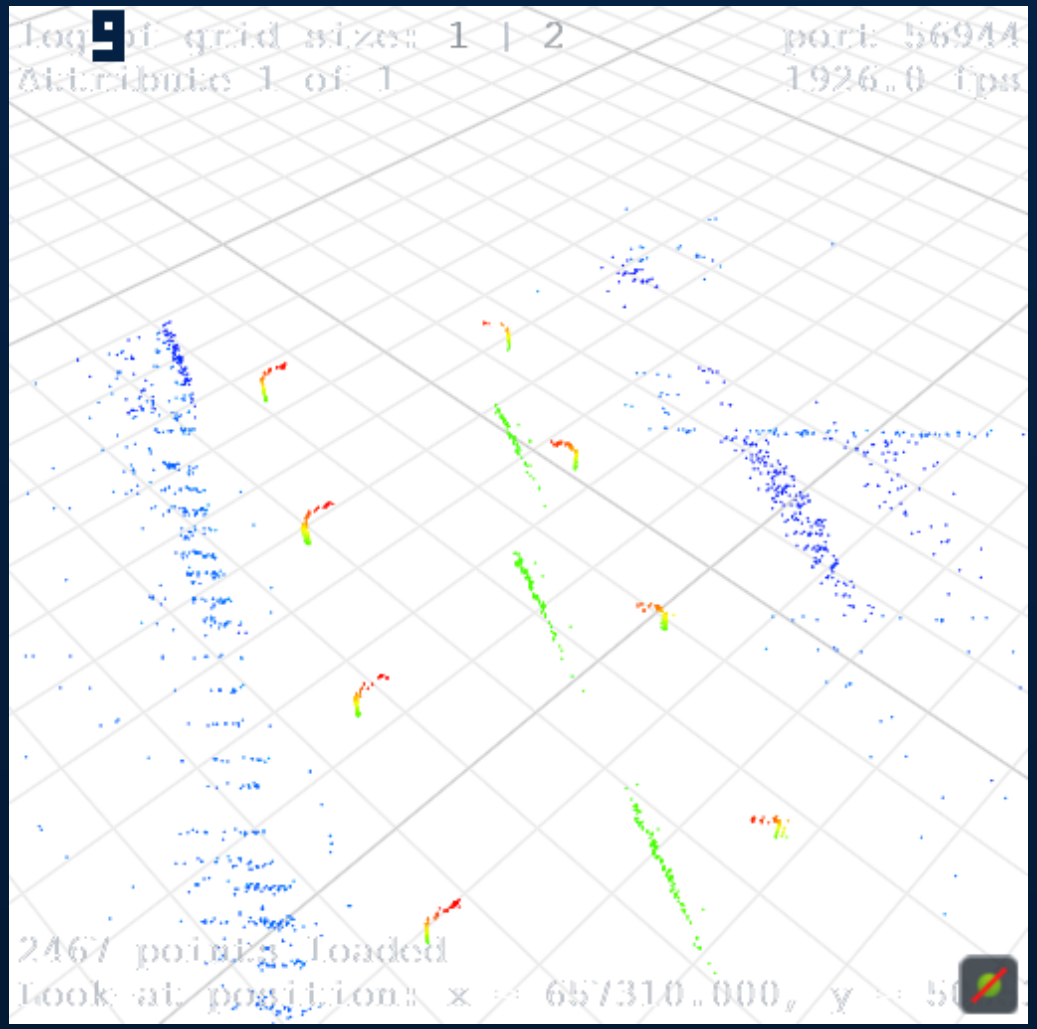
TERRAIN MODEL BY INTENSITY

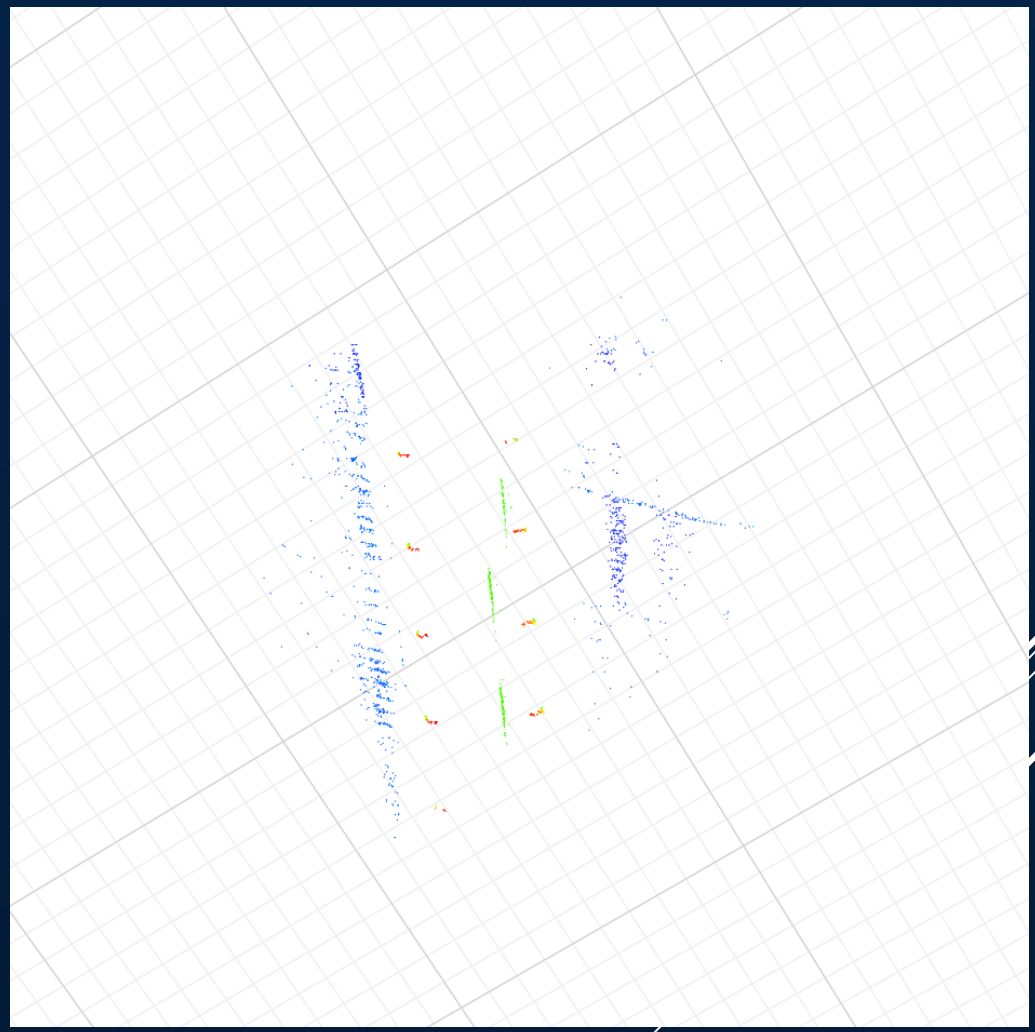




Intensity Visualization

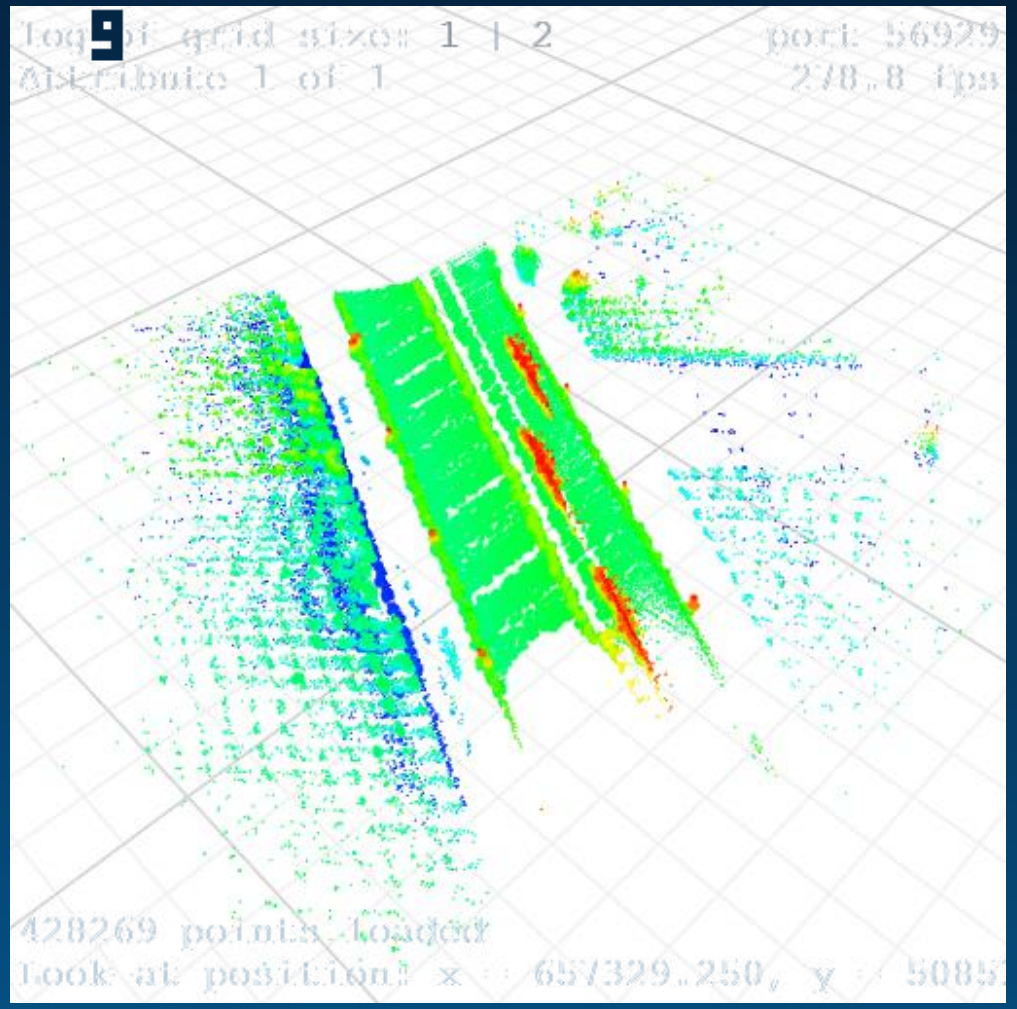
TERRAIN MODEL - OUTLIERS

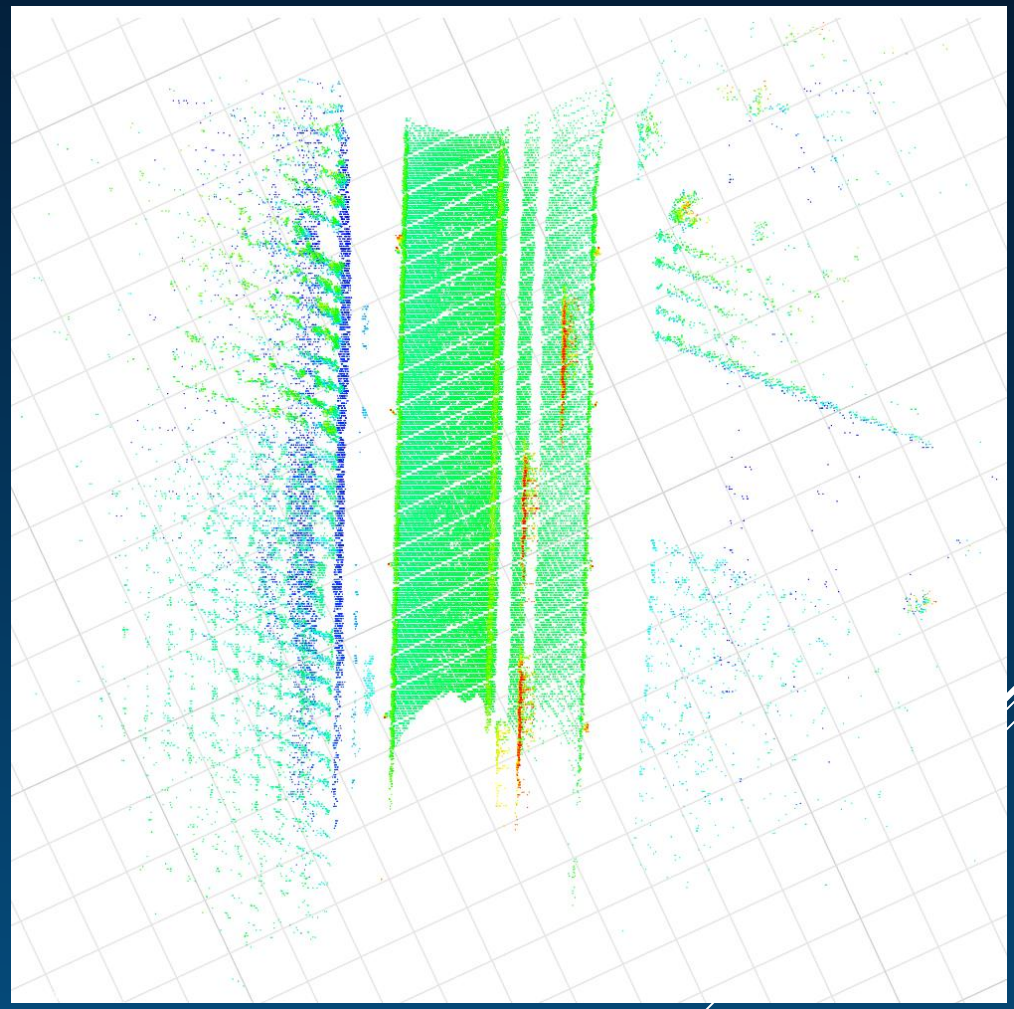




Elevation Map

TERRAIN MODEL FOR INLIERS





Final Elevation Map

WHAT DID WE LEARN?

- DTM and Point Cloud
- How to process and handle the raw data
- Visualize the 3d model in various ways

REFERENCES

- https://www.pointtopointsurvey.com/service/digital-terrain-modeling/
- https://en.wikipedia.org/wiki/Digital_elevation_model
- https://www.sanborn.com/dems-dtms/
- https://gisgeography.com/dem-dsm-dtm-differences/
- https://ieeexplore.ieee.org/document/7009913
- https://www.researchgate.net/publication/228754113_From_point_cloud_to_surface_The_modeling_and_visualization_problem

THANK YOU