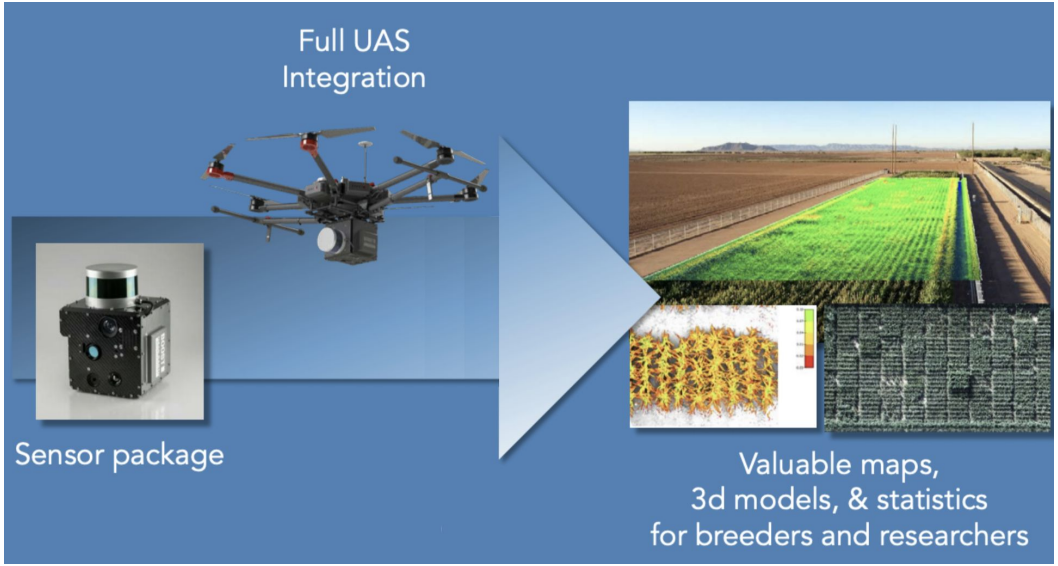
Carnegie Mellon University

Visual Inspection for Aircraft & Power Lines

Chang Gao & Anshuman Majumdar
Master of Science in Computer Vision, CMU

Summary



Motivation

Automate asset inspection with sensor data

Problem

Power line surveys: (1) Map and georeference vegetation around power lines (2) Power line damage assessment

Solution

Offline 3D semantic segmentation with PoinNet++ or Superpoint Graphs

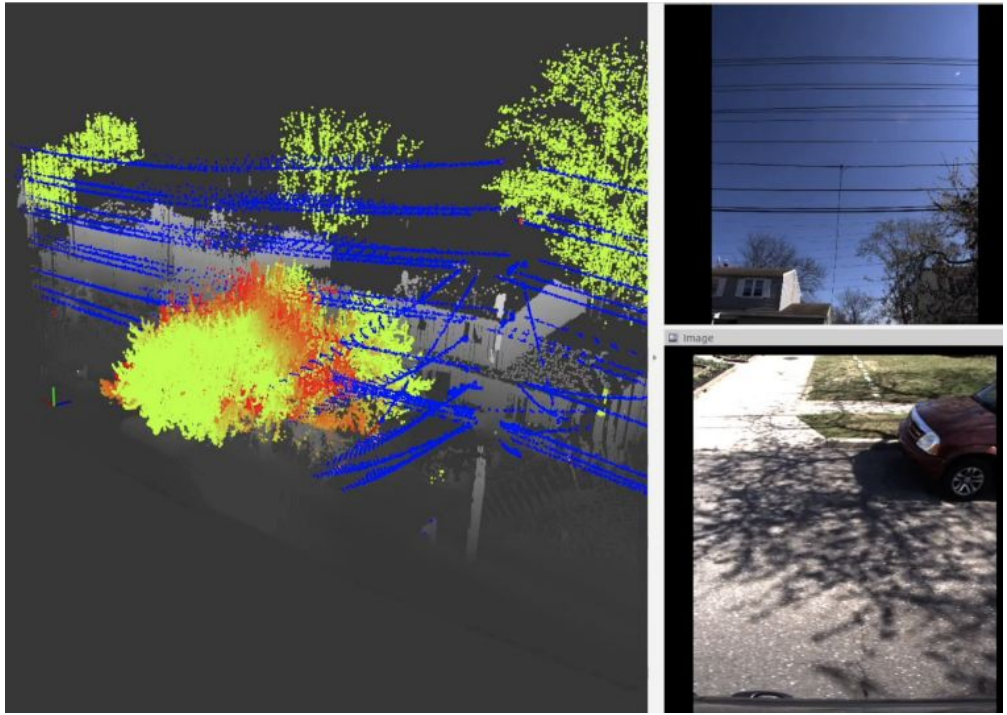
NEAR EARTH AUTONOMY

Carnegie Mellon University

Past (Apr 19): Overview

1. Dataset
2. Anshuman: Superpoint Graphs
3. Chang: PointNet++

Past (Apr 19): Dataset



Calibrated & Processed

3D point cloud for the whole scanned scene

- Two labeled scene and one unlabeled scene
- Classes are power lines, trees, high-voltage power, and ground & houses

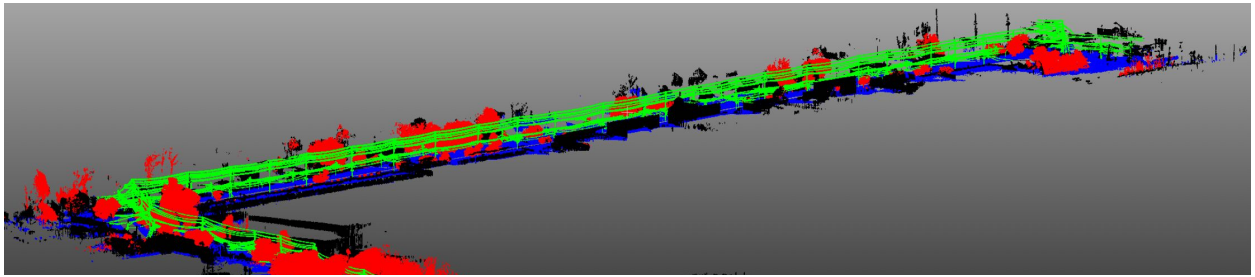
Uncalibrated

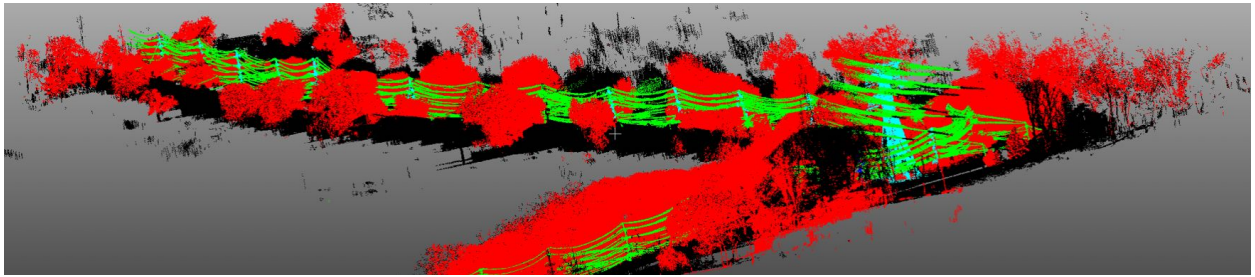
Images from two cameras

- Currently not used

Past (Apr 19): Dataset

Two Labeled & Merged 3D Point Clouds ("circuit1" & "circuit3")





Green = Power lines

Red = Trees

Black = Houses

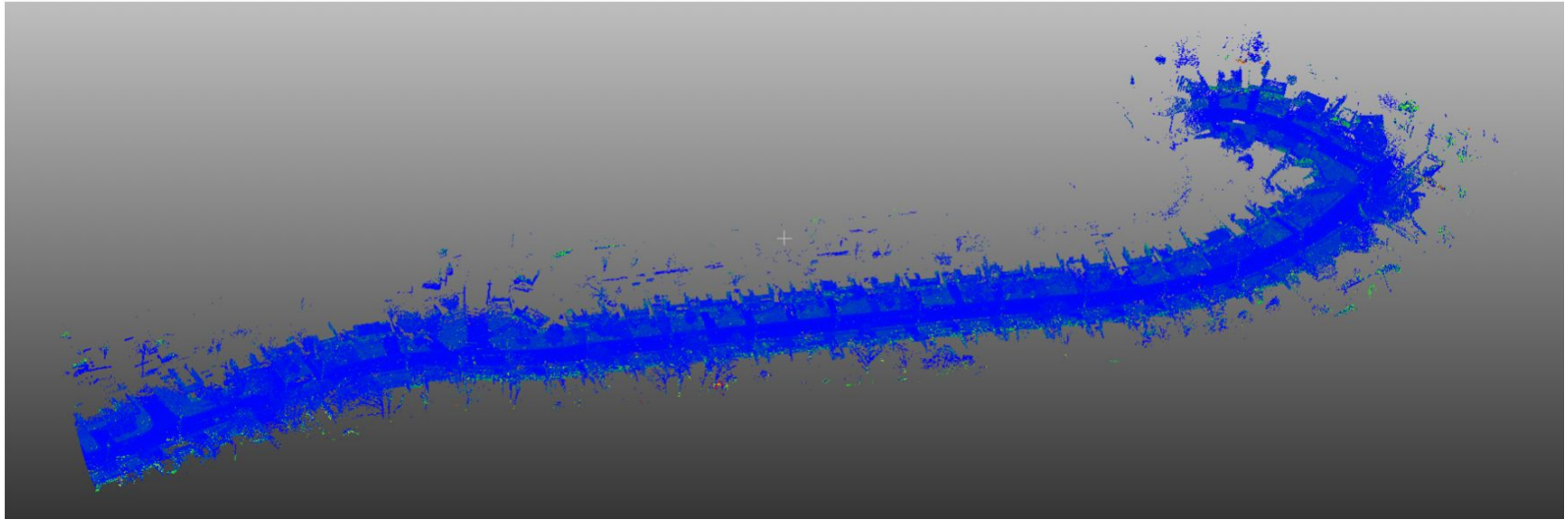
Cyan = High-voltage tower

Blue & **Black** = Ground

Carnegie Mellon University

Past (Apr 19): Dataset

One Unlabeled but Merged 3D Point Cloud (“circuit2”)



Carnegie Mellon University

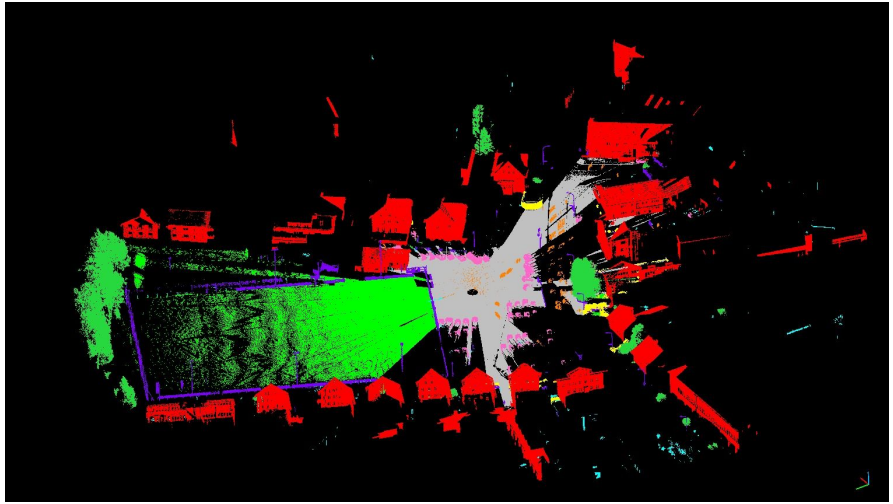
Past (Apr 19): Dataset

Question:

Why two approaches?

Answer:

1. PointNet++ is easier to re-implement
2. Superpoint Graphs claim higher accuracy on a similar dataset (Semantic3D) although hard to adapt



Carnegie Mellon University

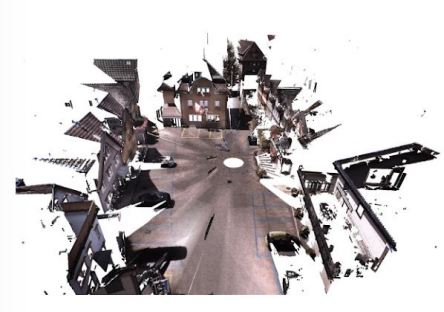
Past (Apr 19): Superpoint Graphs

Anshuman:

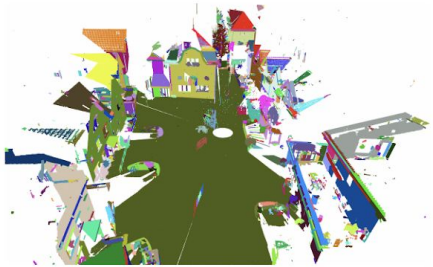
1. Model Introduction
2. Processing Pipeline
3. Data Augmentation
4. Quantitative Results
5. Qualitative Results
6. Potential Problems

Past (Apr 19): Superpoint Graphs

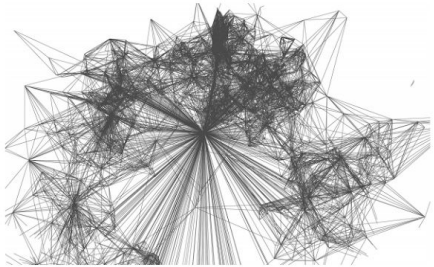
1. **Model Introduction**
   - **Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs**
   - Loic Landrieu et al, University of Paris-Est, CVPR 2018



3D Input



Superpoints



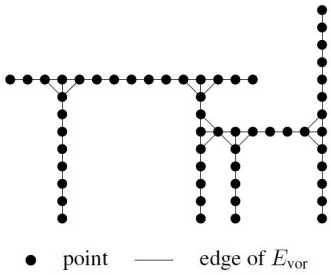
SPG

Segmentation

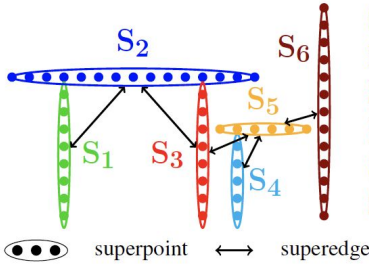
Carnegie Mellon University

Past (Apr 19): Superpoint Graphs

- **Step 1: Points to Superpoints**
  - Form superpoints using unsupervised criteria
  - linearity, planarity, scattering, verticality, elevation
- **Step 2: Connect Superpoints to graph**
  - Connect superpoints using unsupervised criteria
  - offset, length/surface/volume ratio, point count ratio etc.



(a) Input point cloud

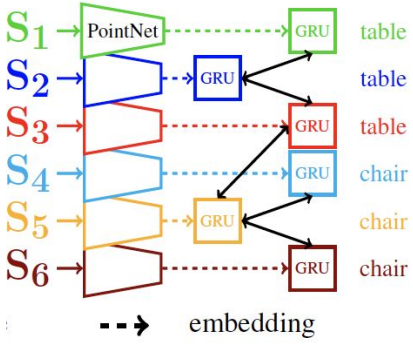


(b) Superpoint graph

Carnegie Mellon University

Past (Apr 19): Superpoint Graphs

- **Step 3: Extract information with PointNet**
  - Use PointNet to compute descriptor for every superpoint
  - Re-sample to a fixed number of points
- **Step 4: Propagate information with GRU**
  - Hidden state: initialized by output from PointNet
  - Input: weighted sum of hidden states of linked superpoints
  - Iterate several time steps

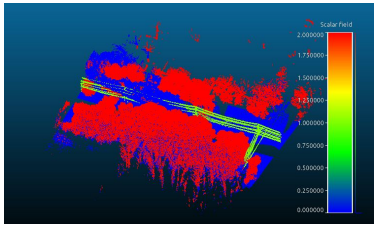


(c) Network architecture

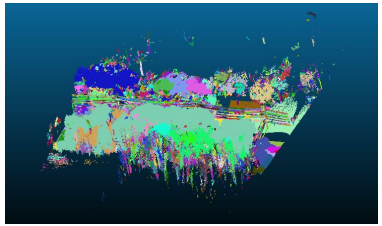
Carnegie Mellon University

Past (Apr 19): Superpoint Graphs

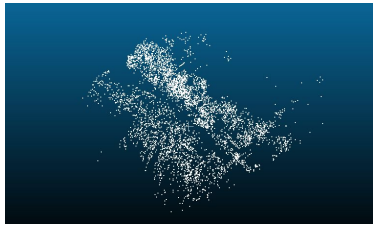
2. Processing Pipeline



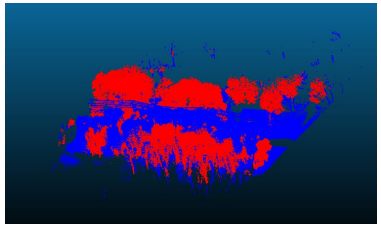
Input Point Cloud



Geometric Partition



Superpoint Graph



Output Point Cloud

Carnegie Mellon University

Past (Apr 19): Superpoint Graphs

3. Data Augmentation

1. Subsampling of superpoints to have 128 points
2. Random rotation of points around z-axis
3. Mirroring of points around x-axis/y-axis
4. Gaussian jittering of all xyz, rgb and geometric attributes

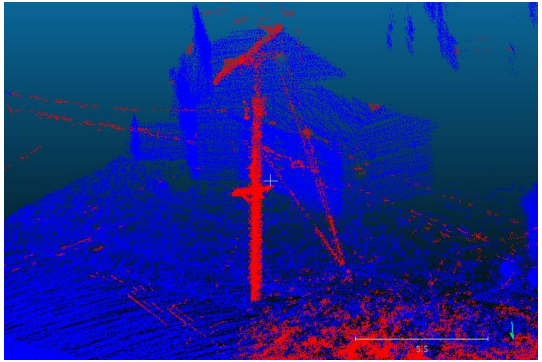
Past (Apr 19): Superpoint Graphs

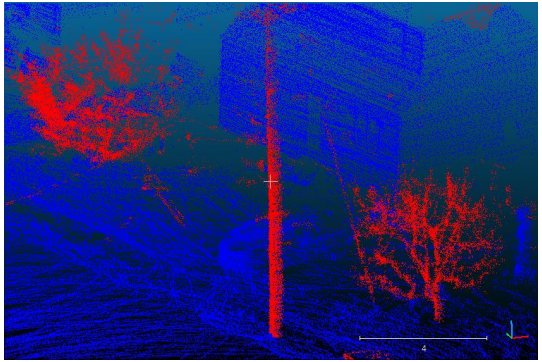
4. **Quantitative Results**

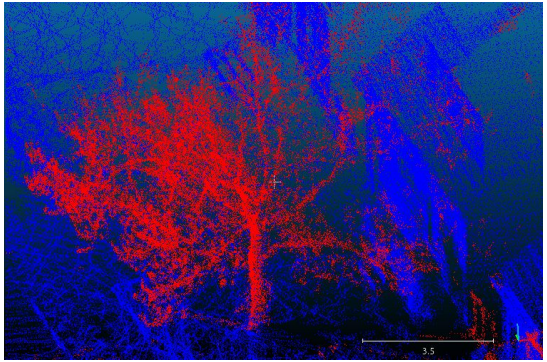
- Separate each of “circuit1” and “circuit3” to train and validation parts
- Use three classes: power lines, trees, and background
- ~80% accuracy on training set
- ~85% accuracy on validation set

Past (Apr 19): Superpoint Graphs

5. **Qualitative Results**





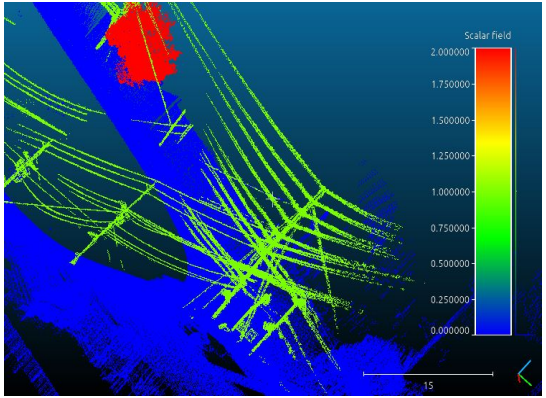


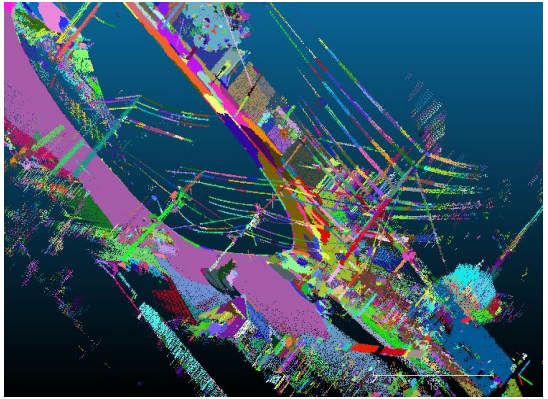
Carnegie Mellon University

Past (Apr 19): Superpoint Graphs

6. Problems

1. The unsupervised partitioning is “really” parameter-specific
2. GRUs are not able to capture the spatial context over long powerline sections





Carnegie Mellon University

Past (Apr 19): PointNet++

Chang:

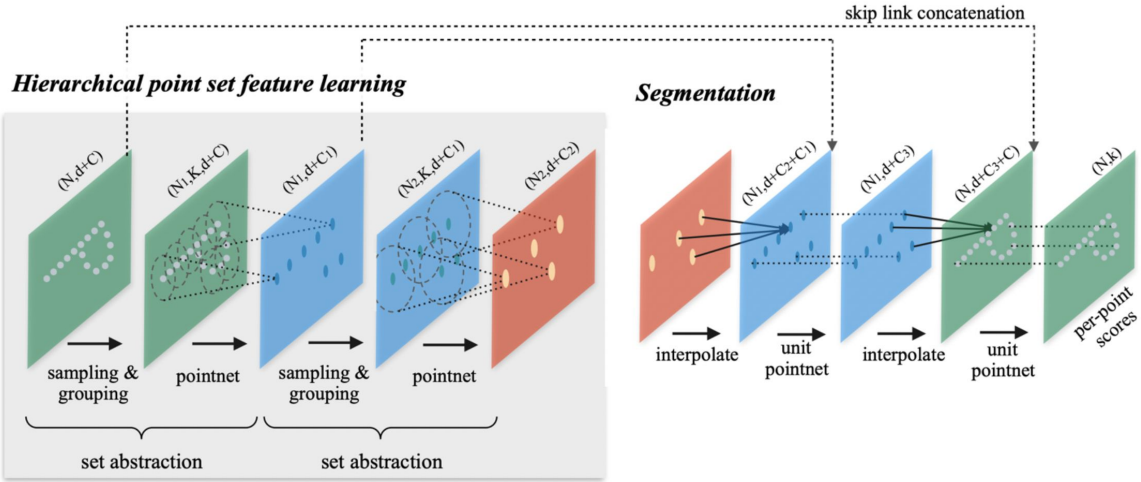
1. Model Introduction
2. Processing Pipeline
3. Data Augmentation & Preprocessing
4. Quantitative Results
5. Qualitative Results
6. Potential Problems

Past (Apr 19): PointNet++

1. **Model Introduction**
   - **PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation**
     - Charles R. Qi et al, Stanford University, CVPR 2017
     - Segmentation based on unordered 3D points
     - Use NN as feature extractor and max pooling as order invariant processor
   - **PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space**
     - Charles R. Qi et al, Stanford University, NIPS 2017
     - Hierarchical PointNet
     - Focus on both local and global information

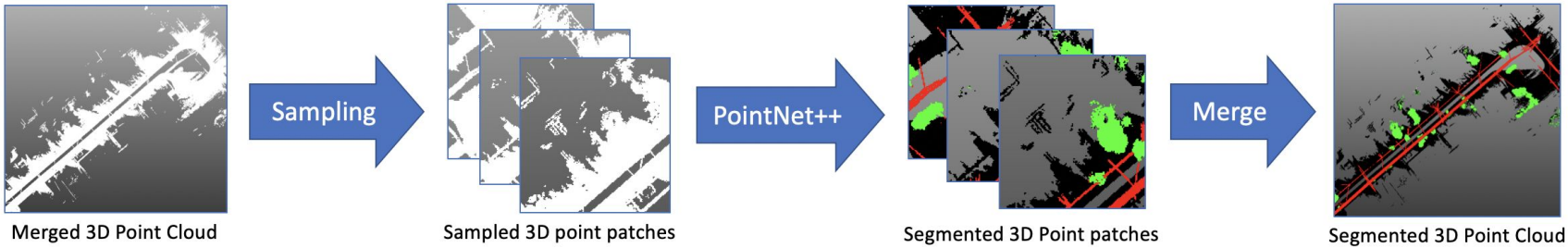
Past (Apr 19): PointNet++

- **PointNet++**
  - Sampling & Group at centroids with K neighbors in radius R
  - Interpolate back to original number of points



Past (Apr 19): PointNet++

2. Processing Pipeline



Carnegie Mellon University

Past (Apr 19): PointNet++

3. Data Augmentation & Preprocessing

1. Train-time Random Rotation
   - Rotate the whole scene for random angle after random steps
2. Voxel Down-sampling
   - Downsample point cloud to only one point per unit cube
   - Reduce density on trees and houses vs power lines
   - Still interpolate to all points in inference stage
3. Weighted Labels
   - Weight points in loss according to their percentage in the whole point cloud

Past (Apr 19): PointNet++

4. Quantitative Results

- Separate each of “circuit1” and “circuit3” to train and validation parts
- Use three classes: power lines, trees, and background
- To perform validation, we train models with
  - Training Set: “circuit1_train”
  - Validation Set: “circuit1_val”, “circuit3_train”, “circuit3_val”

  and

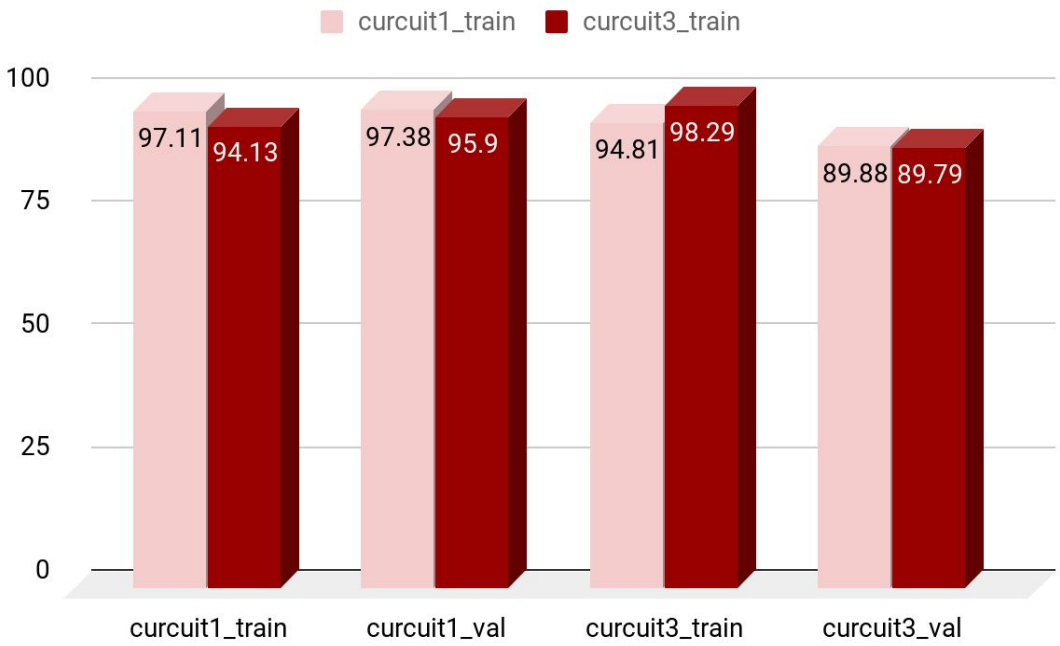
  - Training Set: “circuit3_train”
  - Validation Set: “circuit1_train”, “circuit1_val”, “circuit3_val”

Carnegie Mellon University

Past (Apr 19): PointNet++

4. Quantitative Results

- 97 ~ 98% Training Accuracy
- 94 ~ 95% Validation Accuracy
- mIOU:
  - 77 ~ 90% for power line
  - 83 ~ 82% for tree
  - 91 ~ 96% for background
- Except on “circuit3_val”



Carnegie Mellon University

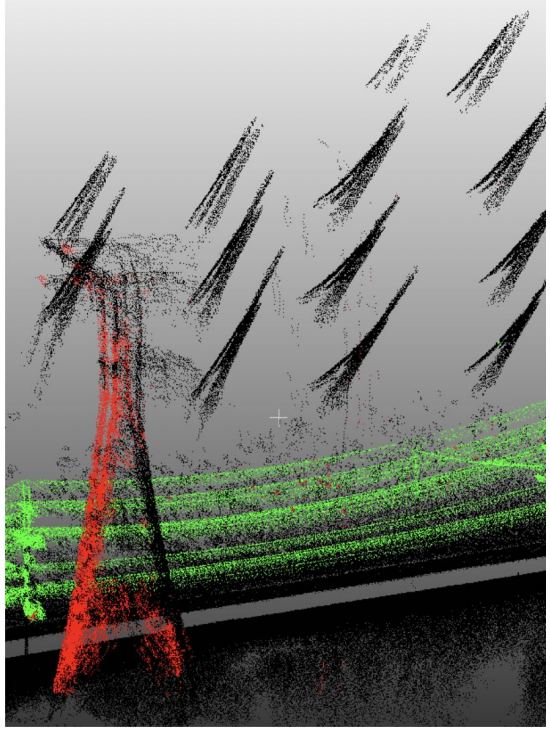
Past (Apr 19): PointNet++

Hard Case 1: High-voltage Tower

- Only one high-voltage tower in “circuit3_val”
- Looks very different from regular powerlines
- Affects validation results

Solution

- Include this in the training set
- If necessary, could simulate high-voltage towers
- Best solution could be collect more LiDAR data

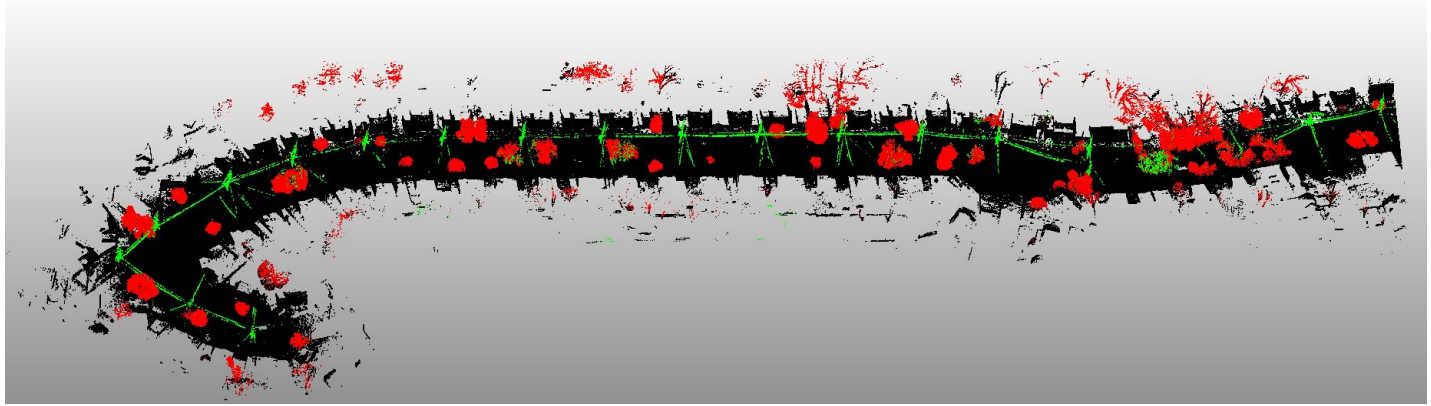


Carnegie Mellon University

Past (Apr 19): PointNet++

5. **Qualitative Results**

- Train on full “circuit1” and “circuit3” scenes to test on “circuit2”
- Correct for most objects

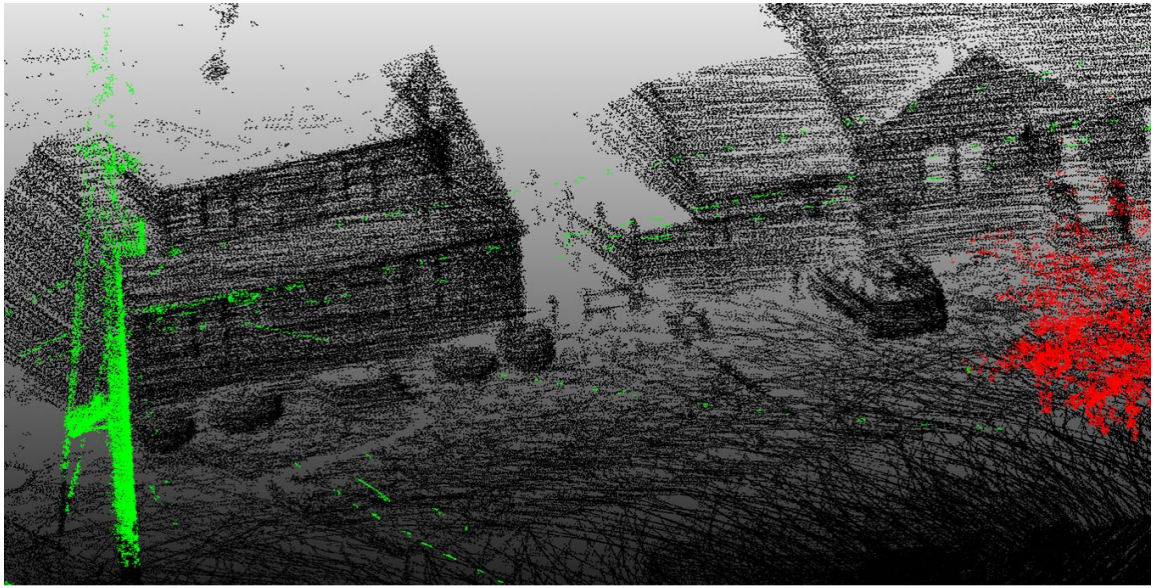


Carnegie Mellon University

Past (Apr 19): PointNet++

5. **Qualitative Results**

- Zoom-ins on “circuit_2”

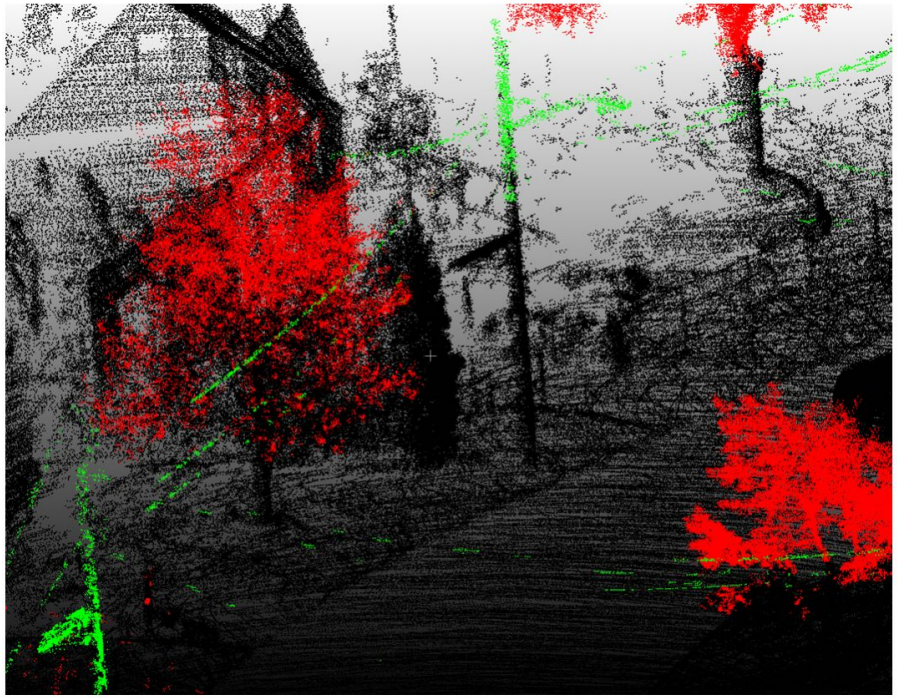


Carnegie Mellon University

Past (Apr 19): PointNet++

5. **Qualitative Results**

- Zoom-ins on “circuit_2”

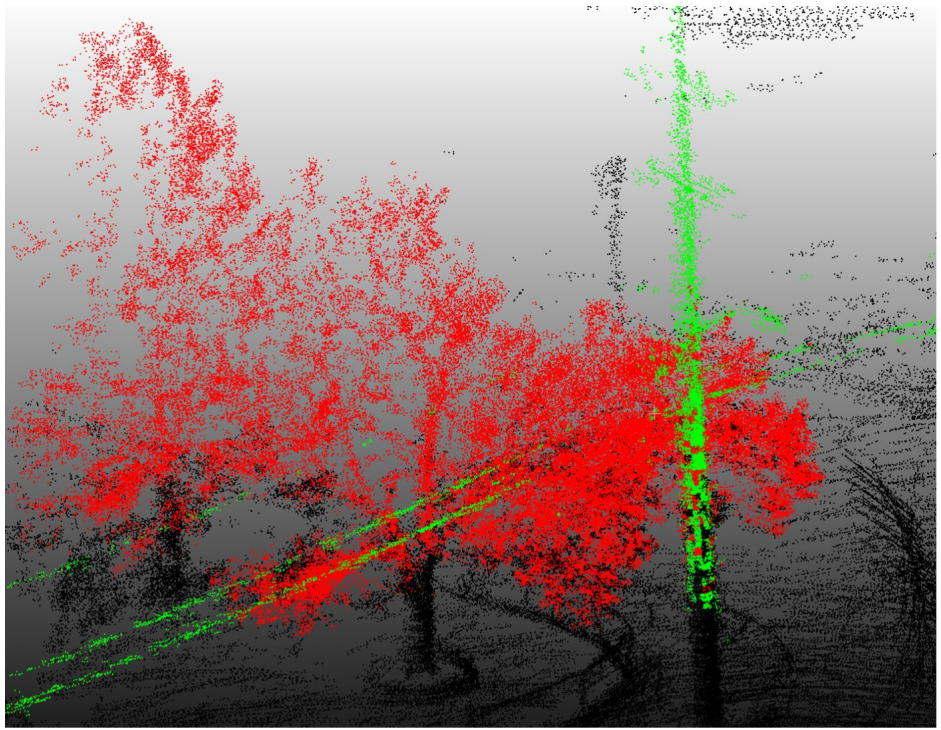


Carnegie Mellon University

Past (Apr 19): PointNet++

5. **Qualitative Results**

- Zoom-ins on “circuit_2”

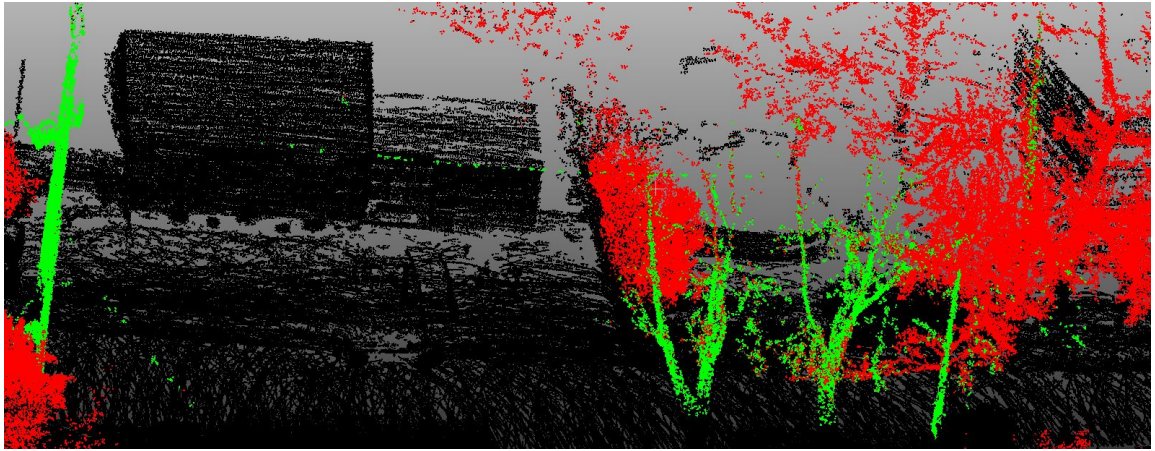


Carnegie Mellon University

Past (Apr 19): PointNet++

Hard Case 2: Trees Without Leaves

- Some trees don’t have leaves and look like poles or parts of power lines
- The test scene “circuit2” contains lots of trees like this

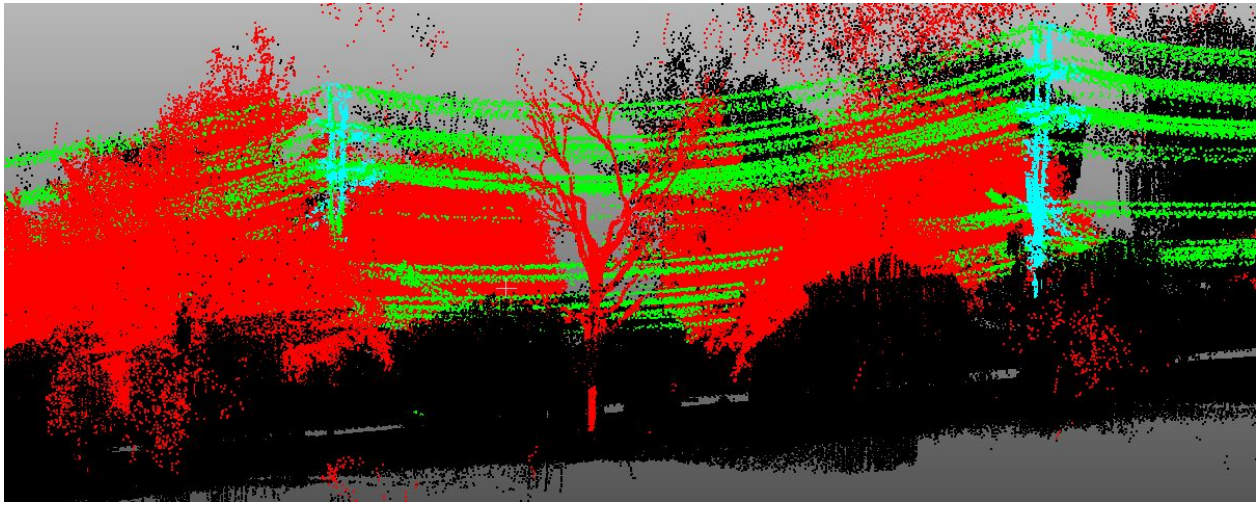


Carnegie Mellon University

Past (Apr 19): PointNet++

Solution: Trees Without Leaves

- Add synthetic trees without leaves to the scene

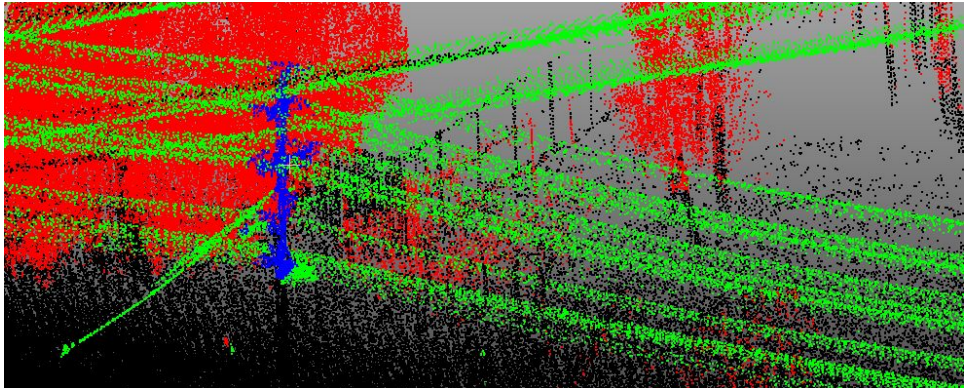


Carnegie Mellon University

Past (Apr 19): PointNet++

6. **Potential Problems**

- Ground-truth labels are not consistent
  - Poles are sometimes labeled as houses or ground
  - Trees, ground and houses are not clearly separated



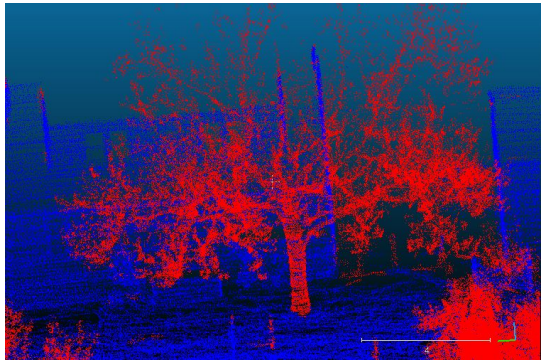
Carnegie Mellon University

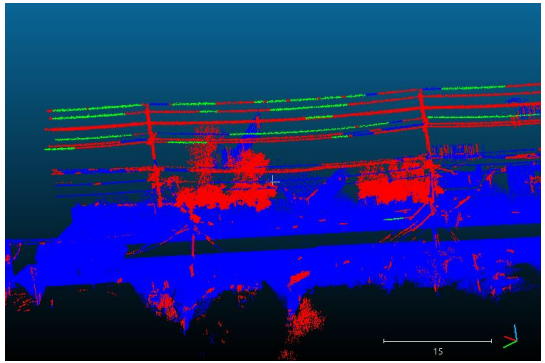
Present (May 3): Superpoint Graphs

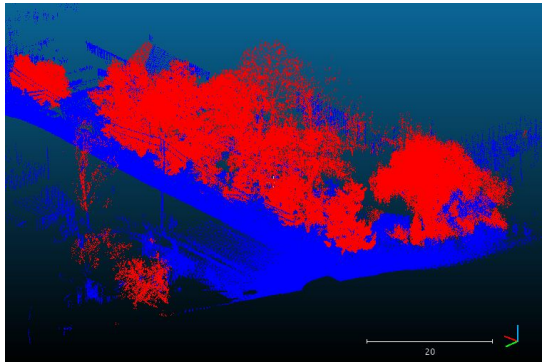
Anshuman

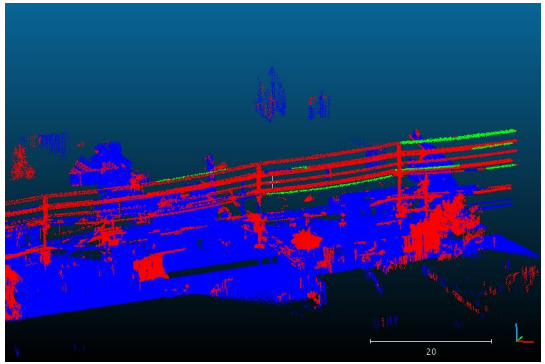
1. Understand the significance of parameters via experimentation
2. Search for ways to incorporate the detection of power lines
3. Try to prove why this system is not performing well on our data

Present (May 3): Superpoint Graphs









Carnegie Mellon University

Present (May 3): PointNet++

Chang

1. Finalization of PointNet++ model

Carnegie Mellon University

Present (May 3): PointNet++

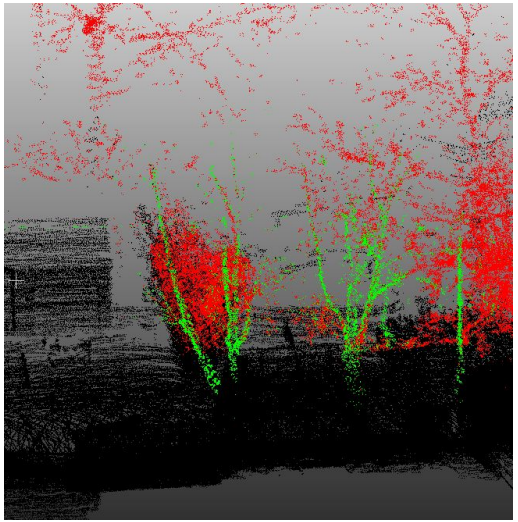
Finalization of PointNet++ model

1. **To resolve issues for trees without leaves**

   - Solution 1: Insert synthetic trees into training point clouds on some nicely chosen positions (“precompute”)
   - Solution 2: Randomly insert synthetic trees into cropped patches of training point clouds (“generate”)

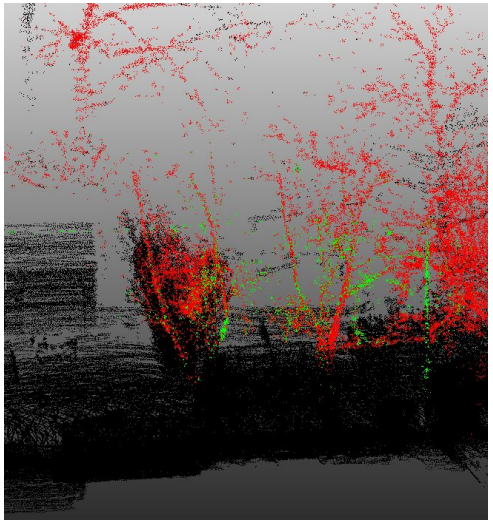
   Turns out solution 1 is slightly better, and both will improve test accuracy.

Present (May 3): PointNet++



original

precompute



generate

Carnegie Mellon University

Present (May 3): PointNet++

Finalization of PointNet++ model

2. We adopt “precompute” as our final model

1. All hyper-parameters saved in a JSON file
2. Documentation finished
3. Pre-trained model saved

Carnegie Mellon University

Future (Fall 2019): Schedule

Power Line Inspection	
May 20	*Finalize designs for Superpoint Graphs model*
Aircraft Inspection (Expected)	
Sep 30	*Finish investigation on aircraft inspection data*
Oct 31	*Finish initial implementation of a computer vision model*
Nov 30	*Finalize designs for aircraft inspection model*

Carnegie Mellon University

Conclusion

1. Attempted two 3D point semantic segmentation methods for automated power line inspection: Superpoint Graphs & PointNet++
2. Implemented Superpoint Graphs model but found it not suitable
3. Customized and implemented PointNet++ model, which achieved high semantic segmentation results both qualitatively and quantitatively
4. Finalized the 3D point semantic segmentation pipeline for automated power line inspection
5. Waiting for aircraft inspection data for the second part of our capstone project