Retroreflectivity Team

Ariel Steele, Danny Taylor, Gabby Germanson

Research Problem

- Traffic sign visibility at night is critical to driver safety
- Manually driving roads to detect road sign replacement is inefficient
- Deterioration has not been attributed to any single factor besides age

Background

Methods of Evaluation:

- Visual inspection - subjective, time consuming
- Retroreflectometer - tedious, only grabs four sign points
- **LiDAR - quick, objective, and returns retro-intensity points to identify sign point clusters**

Objectives

1. Using only LiDAR data:
   a. Locate
   b. Identify
   c. Classify
2. Be able to send GDOT automated sign locations and conditions so they know when to replace road signs at greatest sign life length.

(eventually *predict* sign deterioration)

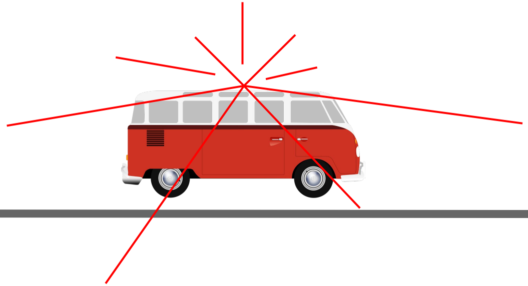
Why should GDOT care?

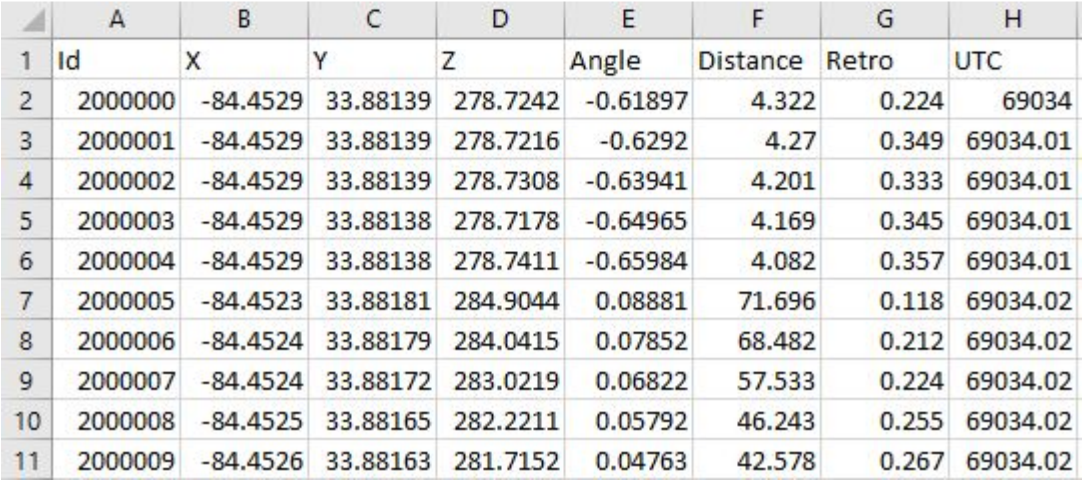
- Calculating a very conservative (lower limit) cost saving analysis where we compare an autonomous vehicle collecting traffic sign info against the current manual empirical method (GDOT employees working 8 hour days, collecting data for 50 signs a day, at $15.00/hr), the savings is over $3 million dollars per year.
  - This figure is only estimating saving on labor costs
  - More savings in only replacing signs when needed.

- This system (along with our other projects revolving Smart City Infrastructure) really starts to shine as autonomous vehicles become mainstream, as we will have access to enormous amounts of data.

Current Status

Drove around and collected LiDAR data for the past 6-7 years of various roads



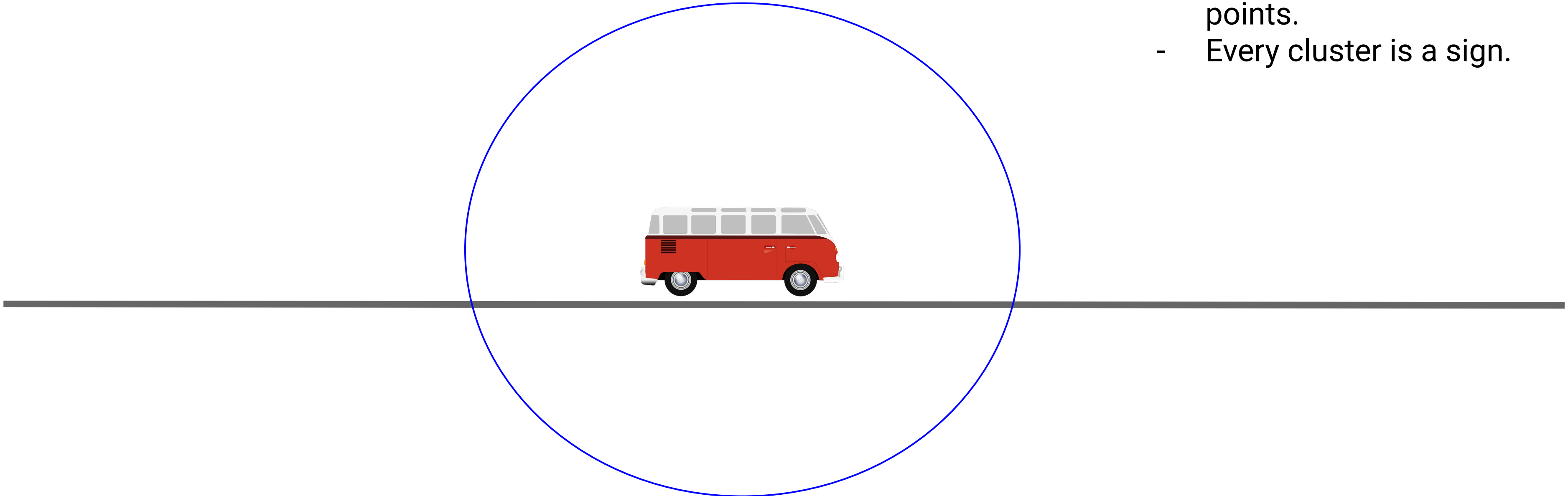


	A	B	C	D	E	F	G	H
1	Id	X	Y	Z	Angle	Distance	Retro	UTC
2	2000000	-84.4529	33.88139	278.7242	-0.61897	4.322	0.224	69034
3	2000001	-84.4529	33.88139	278.7216	-0.6292	4.27	0.349	69034.01
4	2000002	-84.4529	33.88139	278.7308	-0.63941	4.201	0.333	69034.01
5	2000003	-84.4529	33.88138	278.7178	-0.64965	4.169	0.345	69034.01
6	2000004	-84.4529	33.88138	278.7411	-0.65984	4.082	0.357	69034.01
7	2000005	-84.4523	33.88181	284.9044	0.08881	71.696	0.118	69034.02
8	2000006	-84.4524	33.88179	284.0415	0.07852	68.482	0.212	69034.02
9	2000007	-84.4524	33.88172	283.0219	0.06822	57.533	0.224	69034.02
10	2000008	-84.4525	33.88165	282.2211	0.05792	46.243	0.255	69034.02
11	2000009	-84.4526	33.88163	281.7152	0.04763	42.578	0.267	69034.02

Current Status

Step 1:
Filter out points less than a certain R (retrointensity) value.

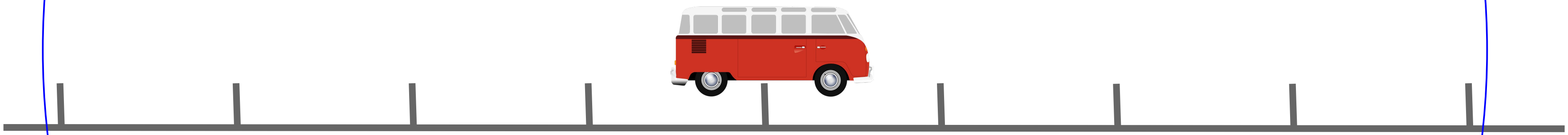
Current Status



Step 2:

- Every 5 meters, draw a 20m sphere and cluster points.
- Every cluster is a sign.

Current Status



Possible Issues

- Lots of duplicates
- Getting retrointensity off the backside of signs <<We discussed they do not>>

Metrics

To establish baseline/groundtruth:

- Analyzed 2 mile stretch on I-285 EB (2018)
    - Manually counted = 42
- Accuracy = (TP + TN) / N = (24 + 18) /42 = 1.0
- Precision = TP / (TP + FP) = 24 / (24 + 0) = 1.0
- Recall = TP / (TP + FN) = 24 / (24 +18) = 0.57

- TP - Algo said there was a sign and there was a sign
    - 24
- FN - Algo said there wasn’t a sign and there was a sign
    - 18
- FP - Algo said there was a sign but there wasn’t a sign.
    - Duplicate signs = 157
    - excluding duplicates = 0

Sources of False Positives

- Guard rails
- Mounds of dirt (construction)
- Traffic lights
- Back of trucks

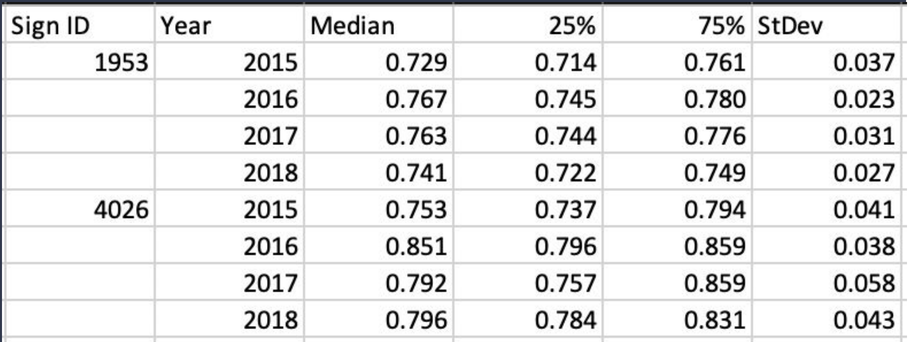
Tasks

- Implement algorithm in OOP design
- Filter out false positive and false negatives with different methods.
  - Custom/advanced clustering algorithms
  - RANSAC
- Find new way of visualizing 3d point cloud.

Expected Outcomes

- OOP implementation should help us test modularly and eliminate duplicates.
- Catch all signs (at least those on the side of the road) and eliminate false positives.
- Have a more accurate way of measuring ground truth and improvements.

Our Progress



Sign ID	Year	Median	25%	75%	StDev
1953	2015	0.729	0.714	0.761	0.037
	2016	0.767	0.745	0.780	0.023
	2017	0.763	0.744	0.776	0.031
	2018	0.741	0.722	0.749	0.027
4026	2015	0.753	0.737	0.794	0.041
	2016	0.851	0.796	0.859	0.038
	2017	0.792	0.757	0.859	0.058
	2018	0.796	0.784	0.831	0.043

- Virtual meeting with Sai Siddarth Maram
- Structuring batch analysis output in CSV format with Java to easily analyze data for deterioration analysis.

To Do Next

- Start OOP implementation
- Reengineer sampling methodology
  - Ex: Instead of every 5m, sample every 20m