LiDAR Road Sign Deterioration Team

Ariel Steele, Danny Taylor, Gabby Germanson

Research Problem

- Traffic sign visibility is critical to driver safety
- Sign reflectivity deteriorates over time due to many factors
  - **Age**
  - Weather / Climate
  - Urban / Suburban location
  - Poor fabrication / installation

Background

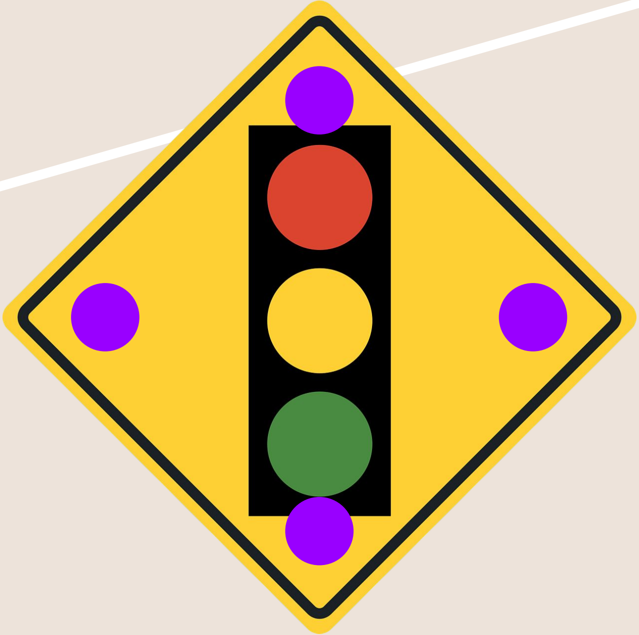
Methods of Evaluation

Visual inspection...

subjective *time consuming* *error prone*

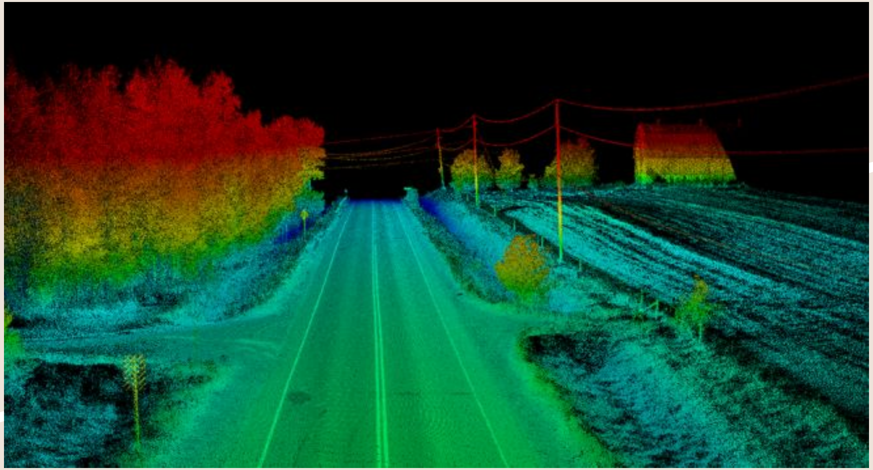
Retroreflectometer...

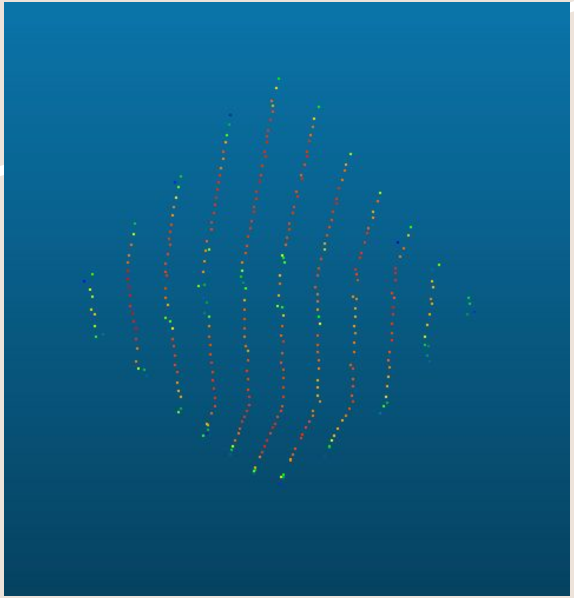
removes bias *very tedious* *not sufficient data*



LiDAR!

quick *objective* *full retro-intensity point clusters*





Problem Statement & Research

Improve current LiDAR algorithm for optimization and efficiency to automate the process of sign detection and deterioration analysis.

This required research into fulling understanding past algorithms of previous sub-teams and pinpointing all edge cases possible in sign detection for first half of the semester...

Why should GDOT care?

- Cost savings over 3 million annually
    - Very conservative calculation
    - Only labor savings
- Our work becomes more valuable as autonomous vehicles become more popular
- Currently, GDOT is mandated by USDT to check signs every 7~10 years

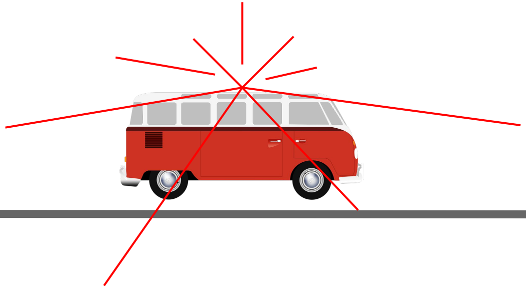
Objectives

1. Using only LiDAR data:
    a. Locate
    b. Sign Identification
    c. Classify Condition
2. Send GDOT automated sign locations and conditions so they know when to replace road signs at greatest sign life length.

(eventually *predict* sign deterioration)

Methodology

Drove around and collected LiDAR data of van's surroundings for past 6-7 years of various roads

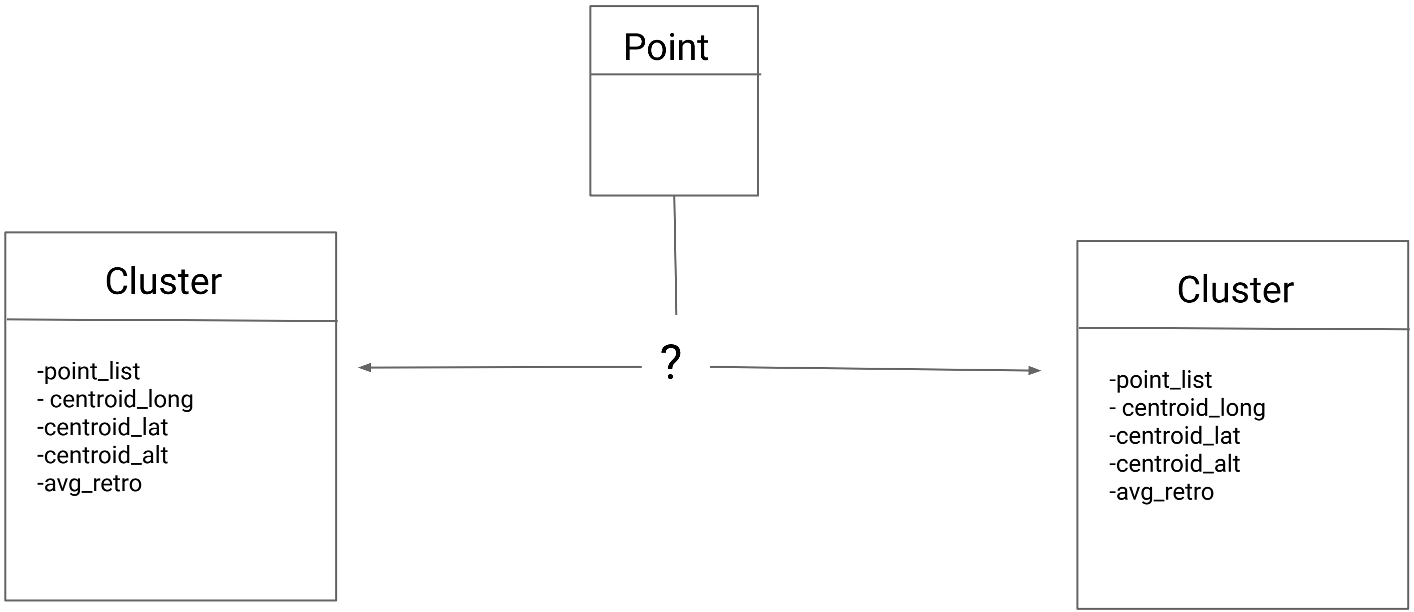


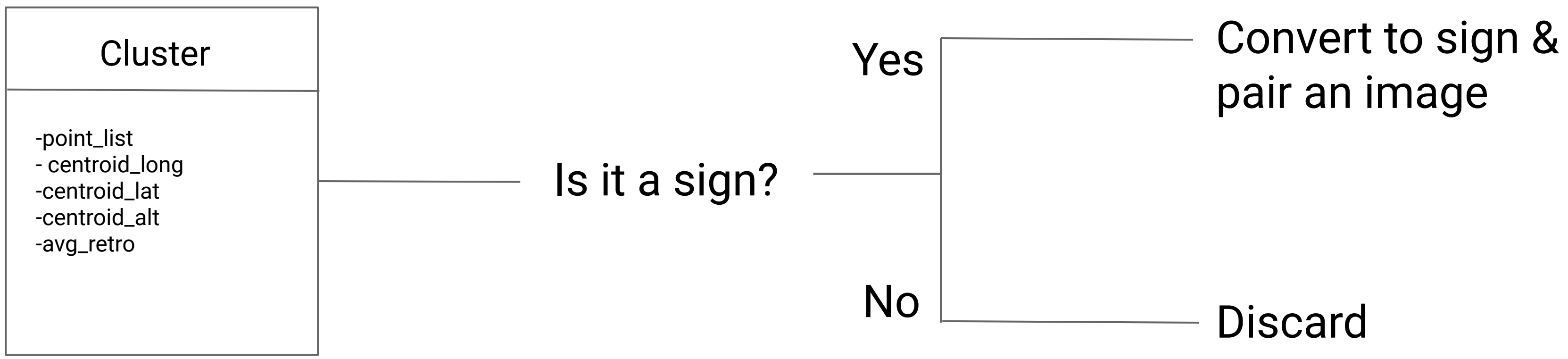
	A	B	C	D	E	F	G	H
1	Id	X	Y	Z	Angle	Distance	Retro	UTC
2	2000000	-84.4529	33.88139	278.7242	-0.61897	4.322	0.224	69034
3	2000001	-84.4529	33.88139	278.7216	-0.6292	4.27	0.349	69034.01
4	2000002	-84.4529	33.88139	278.7308	-0.63941	4.201	0.333	69034.01
5	2000003	-84.4529	33.88138	278.7178	-0.64965	4.169	0.345	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

Point

- -id
- -long
- -lat
- -alt
- -angle
- -distance
- -retro
- -utc

	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


Point
Cluster
-point_list
- centroid_long
-centroid_lat
-centroid_alt
-avg_retro
?
Cluster
-point_list
- centroid_long
-centroid_lat
-centroid_alt
-avg_retro



Cluster
-point_list
- centroid_long
-centroid_lat
-centroid_alt
-avg_retro
Is it a sign?
Yes
Convert to sign &
pair an image
No
Discard

Metrics

Comparison to previous implementation:

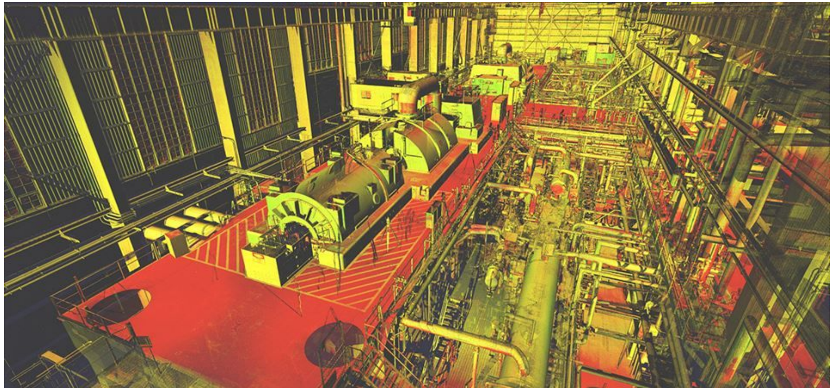
- Got rid of duplicates
  - Old: 157 duplicates   New: 0 duplicates
- Missed fewer signs
  - Old: 18 missed   New: 4 missed
    - Stacked signs
    - Found smaller signs
    - Found signs hidden in brush
- More false positives
  - Old: 0 fp   New: 3 fp

Challenges

- Visualization
  - Understand false positives.
  - Create ground truth dataset.

- Testing new algorithms efficiently.

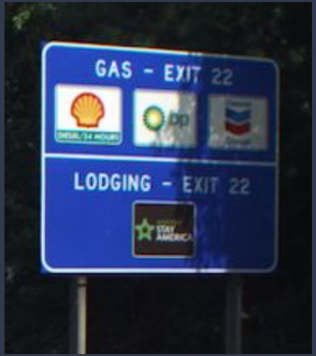
Visualization

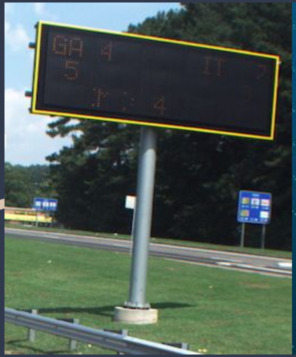


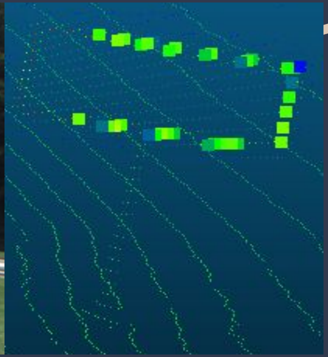
- Autodesk - ReCap
- Bentley: Pointools
- Leica: Cyclone
- Faro: Scene
- Rigel: RiSCAN Pro
- Trimble: RealWorks

- CloudCompare
- pptk

Key Findings

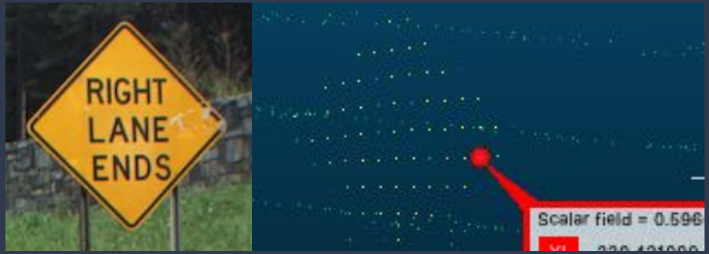


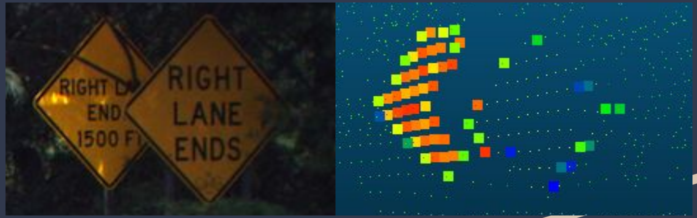




- Defined targeted signs
  - Signs immediately right of van excluding construction and overhead signs
- False positives
  - Large signs identified as more than one sign
  - Retro-reflective material on non-sign objects

Key Findings





- False negatives
  - <20 points higher than .7 Ra
  - Stacked signs considered one
  - Signs not fully identified

Take away from key findings

- Simple filtering can lead to mistakes.
- Need more robust clustering and classification algorithms.

Testbed

```python
# Command line and initialization ^^^
# ======================================== Filtering ==================================================
    # Filtering by Retro values
    df = df.loc[df['Retro'] > 0.7]
    df = df.reset_index(drop=True)
# ===================================================================================================

# ======================================= Clustering ==================================================
    cluster_manager = ClusterManager()
    cluster_manager.cluster_list = cluster_manager.progressive_kdmean(df)
# ===================================================================================================


# ======================================== Classifying ================================================
    sign_manager = SignManager()
    sign_manager.sign_list = sign_manager.num_points(cluster_manager.cluster_list)
# ===================================================================================================
# Book keeping vvvv
```

- Set up a testbed to try different types of algorithms.
- Each section has a set input and output.
    - Easy testing.
    - Encapsulation
- Have a quick way to validate results.
- Once optimal algos are identified, then re-write program for further optimization.

Potential Validation Benchmarks

- Percent of each sign collected
- Percent signs collected
- False positives
  - Splitting a single sign
  - Retroreflective material on non-signs
- False negatives
  - Low satisfactory retro-intensity point count
  - Stacked sign coupling

Must ignore overhead signs, construction signs, and signs beyond immediate right of the van.

New Adjustments

- GPS lat/long to UTM coords
  - Less distortion when analyzing a flattened zone
- Standardized (testbed) and documented code

Conclusions

- Goal is not for 100% automation but rather making condition groupings
    - Satisfactory, needs replacing, and unknown

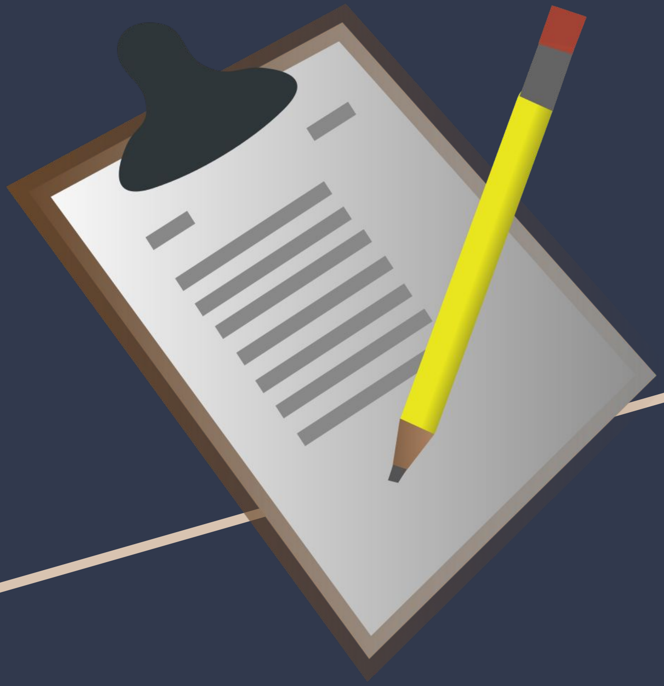
- New Problem Statement:

Develop robust LiDAR algorithms and establish validation methodology to compare adjusted LiDAR algorithms for sign detection and deterioration analysis.

Future Steps

- Test out new algorithms
  - Clustering
    - Neural Net
    - Kd Mean
- Create autograder system for a short data set for a quick way to compare algorithms.
- Identify other factors for filtering
  - Distance
  - Angle
- Consider how to ignore construction and overhead signs

Future Steps



- Implement transverse maximum distance filter
- Consider how to ignore construction and overhead signs
- Figure out at what LiDAR distance should data start collecting
    - Farther out, lower Ra

References

- 3D Point Cloud Visualization Software Documentation
  - https://docs.google.com/document/d/1lWV_zxo94Qcm9UOF9eZz7UDrjPKru4N7oiLjtXQ0SOY/edit?usp=sharing
-

Appendix

- Detailed methodology (to be referenced in case of technical questions)
-