# IOT PROCESSING OF GPS DATA

**VISHNU BYREDDY,VENKATA KARTEEK PALADUGU**

**Abstract:**

Turns and Stop detection from GPS coordinates are very important as they are the building blocks for autonomous vehicles. These points help in improving road safety and allow autonomous vehicles to be a good reliable option. This project is all about reading GPS coordinates, cleaning it appropriately so that all redundant, invalid data and anomalies are removed and then processing it to detect stops/signals along with left and right turns. Since its real data the coordinates might be missed some times and also there will be noise in the data and so preprocessing it is very important and 60% of the project goes into it. After that remaining 40% goes into feature selection, threshold selection to build classifiers that classify points into either a stop/signal or turn. After classifying we found out that most of the turns and stops/signals are detected successfully.

## Overview[1]:

The project is divided into various phases, the first phase being reading GPS data and converting it into KML which is clearly described in Section F which is all about cleaning the data by choosing appropriate fields and correct entries and then converting the entry into appropriate KML format.We faced many problems when reading data i.e. some of the characters are not UTF-8 encoded so we had to ignore those entries, some of the entries have $GPGGA but not $GPRMC and vice versa so we had to take care of it and ignore both so that next entries do not get corrupted and lastly we had entries with $GPGGA and $GPRMC both being in one line so we had to ignore them.

Next phase is about building a classifier to detect turns and stops/signals from the cleaned coordinates which is clearly described in Section G and Section H which is all about selecting appropriate features such as speed, time and angle and selecting an appropriate threshold for those features so that they can classify if a point is a signal/stop or turn based on those values. Based on our analysis we found out that if a point has a speed below 12mph and if the waiting time is more than 10seconds and less than 90 seconds and if angle is less than zero its a stop else if angle is more than 30 degrees and time duration is in between 10 and 120 seconds then its a turn. We had to try different angles, speeds and time duration and since the data is not labeled we had to manually check if the resulting classified points are correct or not and so this took a lot of time to carefully select those thresholds.

## Team Composition:

We both initially discussed and came up with the architecture of the project.Later Venkata Karteek Paladugu did the reading part i.e reading and cleaning phase of the project. Vishnu Byreddy also contributed to the cleaning phase of the project. Later Vishnu Byreddy implemented the classification phase where we both discussed the features and thresholds to start with. Later both of us contributed to improve the classification part by manually inspecting and checking what values can give better results and then both of us were continuously contributing to the code from then by making it more sophisticated. Finally both of us have collaborated in finishing the write-up for the project.

**Project Decomposition:**

Firstly we both discussed and divided the project into parts as described below:

a.Reading Data: In this part we read the file and consider fields that are useful for processing the data and classifying. We also consider lines that are proper and not corrupted.We can consider it as a **Demand push** here because data is pushed at us to process.

b.Cleaning Data: In this part the anomalies, the invalid data, empty records and records not having all the required fields are removed from the collected data.

c.Data Visualization: After reading and cleaning few files plotted a histogram plot of speed of the vehicle with appropriate bin size of 15 as shown in Figure A to see the distribution behind the data as histogram approximates the density of underlying distribution of the data and we saw the probability density of speed below 12mph is less so we took that as a threshold to check for stops.

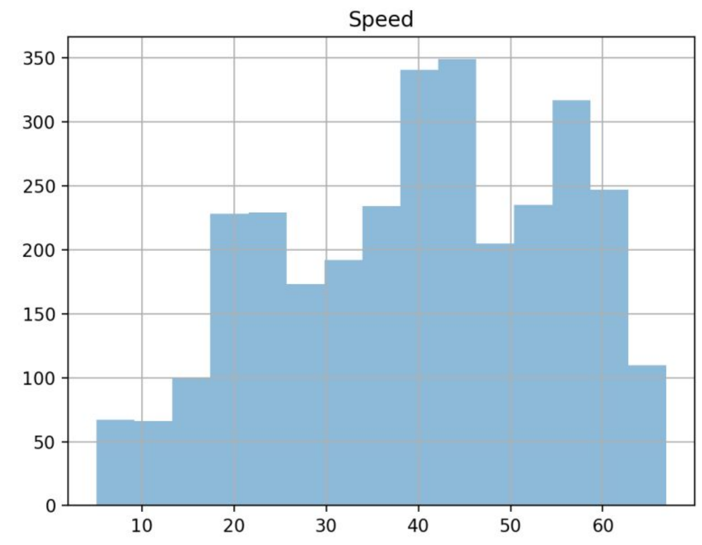


Figure A

d.Classifier:Designed a  classifier by choosing the appropriate features for the turns as well as the stops that impact them the most. Based on above speed we can know if a point was a stop point or not and later we also considered angle and time to classify a stop point to be a turn or a signal point. Also during some turns the speed might be high so based on angle alone considered few turn points.

e.Validation: Since the data is not labeled we have to see visually after classifying if its correct or not by checking manually in Google Earth and if we see any misclassification we discuss it and tune the parameters we chose and keep refining the classifier until the no of misclassifications are less.

**<u>Converting GPS to KML</u>**:

Inorder to write the KML from GPS data we first need to read the GPS data from the given text files and had to choose relevant data from both GPRMC and GPGGA for processing so choose the ['Lat', 'Lat_dir', 'Long', 'Long_dir', 'Quality', 'Dilution'] fields from GPGGA and ['Speed', 'Validity] fields from GPRMC but during the reading part we have faced the following issues:

a.Firstly there were some characters that were not UTF-8 encoded so we got exceptions while reading, and we had to handle these exceptions by ignoring the characters that were not UTF-8 encoded.

b. Secondly the records in the text files were irregular i.e. we needed both the GPGGA and GPRMC logs for our processing but some of the lines had either of those and not both so we had to ignore lines of that format.

c. Thirdly both GPGGA and GPRMC logs were in one line so we checked the length of each line and if it's greater than 15 we ignored those lines as they will mostly contain both the logs in one line which can cause errors while processing the data.

d.Finally if there were any empty lines we ignored those lines.

After the reading part is done we had to clean the data as there was redundant data which could increase the processing time.Below are the following steps taken to clean the data:

a.Firstly all the duplicate data having the same latitude and longitude are removed, because if the vehicle is parked at a place we get multiple data points at the same location which are redundant.

b.Secondly retained only the records that have the valid field set for the Validity field in the dataframe because invalid data will cause issues while processing.

C. Thirdly we only choose entries whose quality field is good i.e. quality should be less than or equal to 2[3].

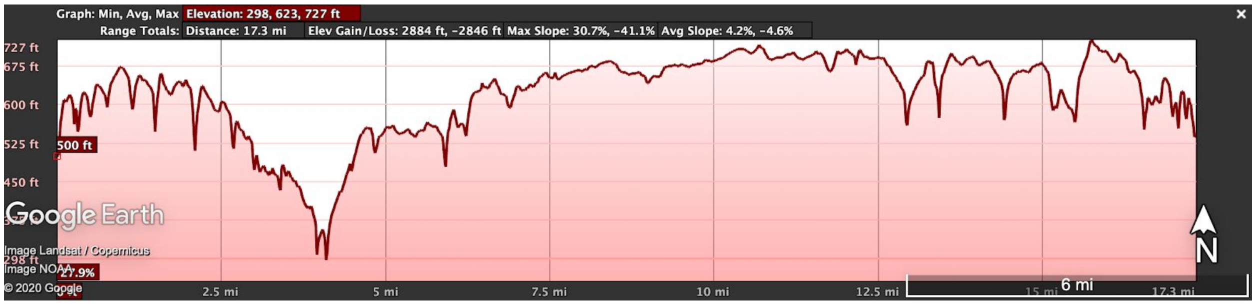
c.Thirdly removed all the records whose dilution of precision[2] is greater than or equal to 5 as this indicates the precision of a point and if it's greater than 5 we get a point that is not so accurate which causes issues while processing.

**<u>Stop Detection:</u>**

For detecting stop signs and stop lights, we used a rule based classifier. If the speed at a particular position is below some threshold, we check if the vehicle has decelerated to that point and starts accelerating again. We do this by checking 10 points on either side of the selected point and check if the speed is greater on either side. This would mean the car slowed down to stop at that particular location and started accelerating again. Once such a point has been found the next objective is to differentiate points at stop signs and stop lights from the points where the vehicle has simply stopped for some errands.

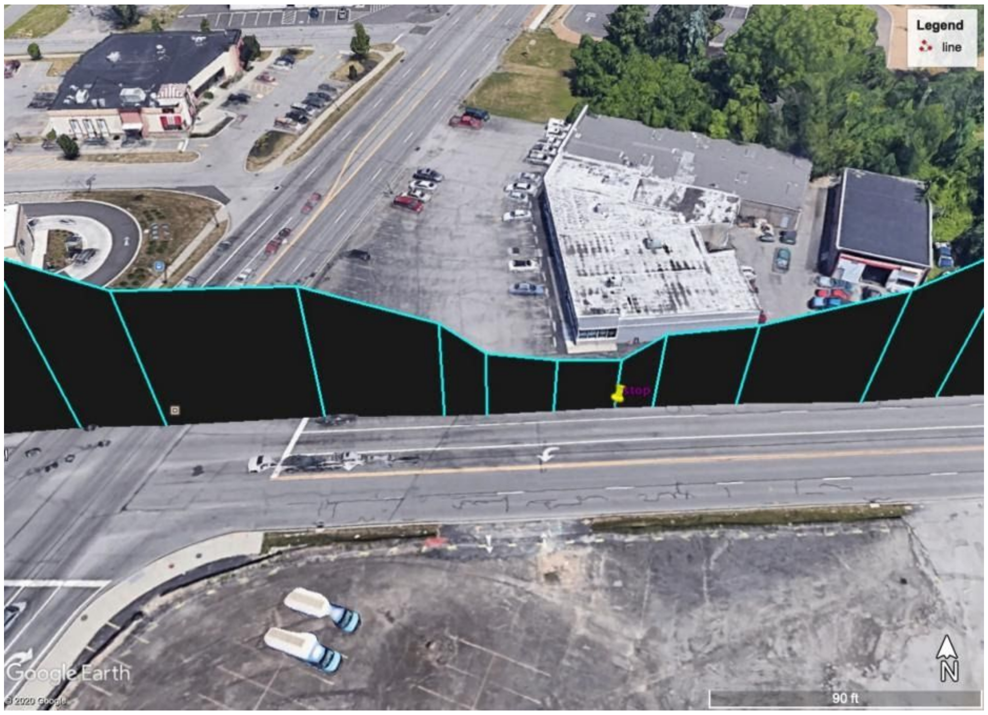
We used time difference between points immediately before and after the stop point to determine if the vehicle stopped for a sign or for an errand. If the time difference is greater than 10 seconds and less than 120 seconds it will be considered as a stop. The reason for the lower limit was to make sure that slowing down before a signal to avoid stopping is not considered as a stop. The upper limit is the average wait for an American signal light.

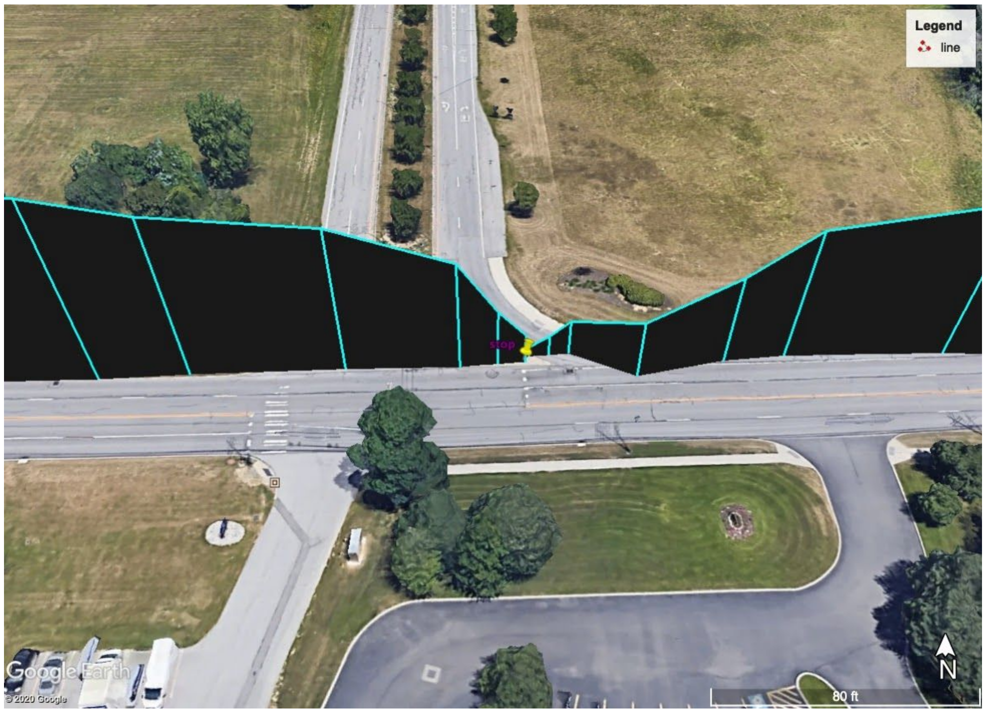
**Following is graph which gives a speed profile of one of the path(speed represented by altitude in ft)**



To identify stops we are trying to find the dips in the graph where the speed reduces.

In the images below we can observe two stop lights. Both the placemarks indicate stop lights and we can observe that speed on either side of the stop point is greater than the speed(indicated by the altitude) at that point.



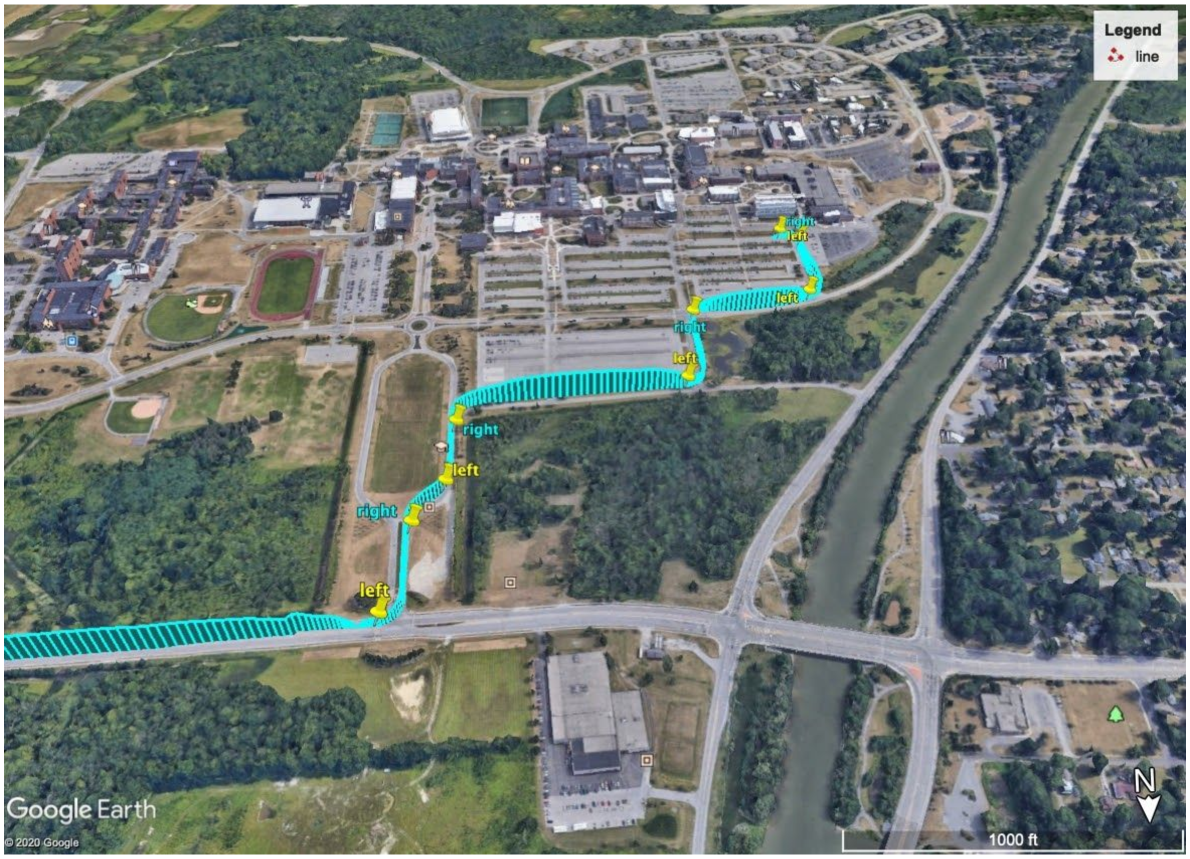


**Note:**

**Path is also shown along with the hazards in the above pictures for reference.**

**Turn Detection:**

For detecting turns we used the geographic library in python to compute the azimuthal angles between two points indicated by latitude and longitude. Angle between two points is calculated wrt to the Greenwich Meridian. So for every point we checked if the angle between its predecessor point(differing by 10 units) and its successor point(differing by 10 units) lies in the range -90 to -180 or 180 to 360 - **a right turn,** or -180 to 180 - **a left turn**. We have also included the time difference to avoid classifying lane changes or any other anomalies while driving in a straight line as turns. In the below image we can observe the predicted turns for vehicle driving from the bottom left of the map to the top right.

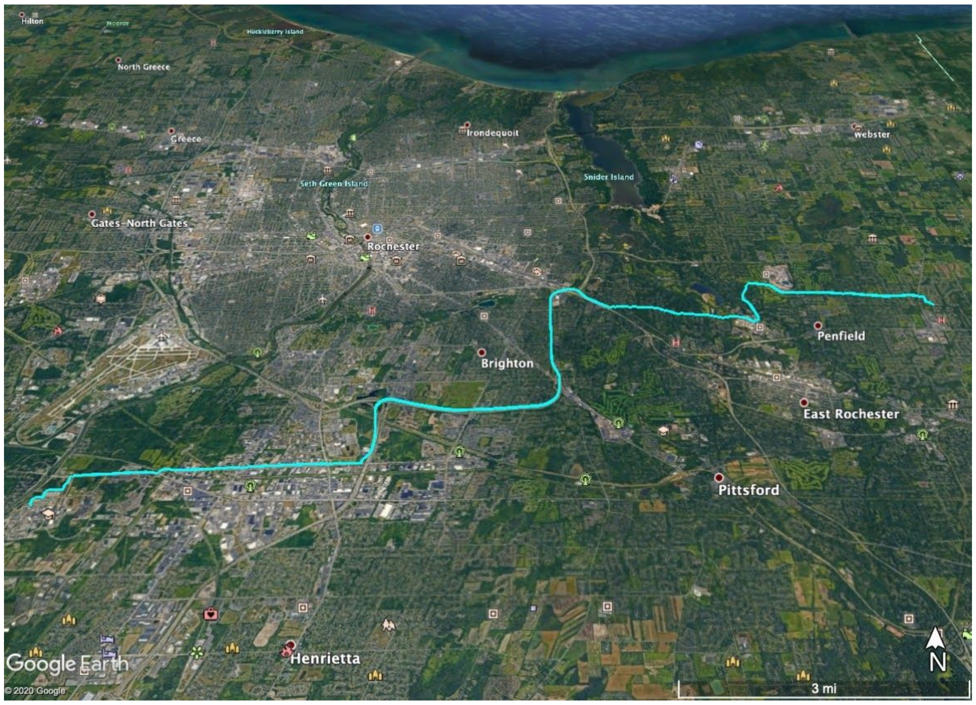


**Data Assimilation across tracks:**

All the .txt files that correspond to various paths are read one by one and all the coordinates are stored into memory. While storing the coordinates we checked if a certain point already exists in our set of points(O(1) to find a certain element) or not. For turns if there is a path going in the opposite direction it would result in two turns at the same point representing both right and left, while multiple stops would not be placed one on top of another. Since the coordinates might also change a little there is a possibility multiple hazards might be placed in close proximity to one and other. To eliminate this problem we used a delta value of 0.001 to check if there already exists a hazard. Only if it's not present do we put a placemark.

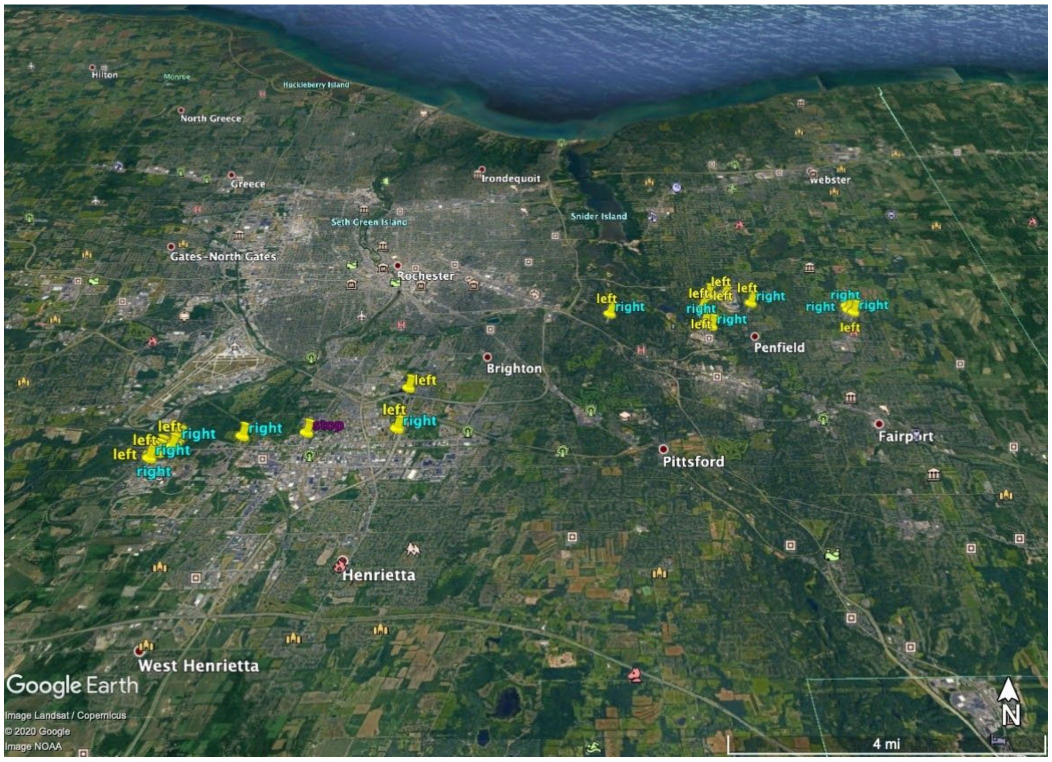
**Results of path file:**

The path from Penfield to Roechester.

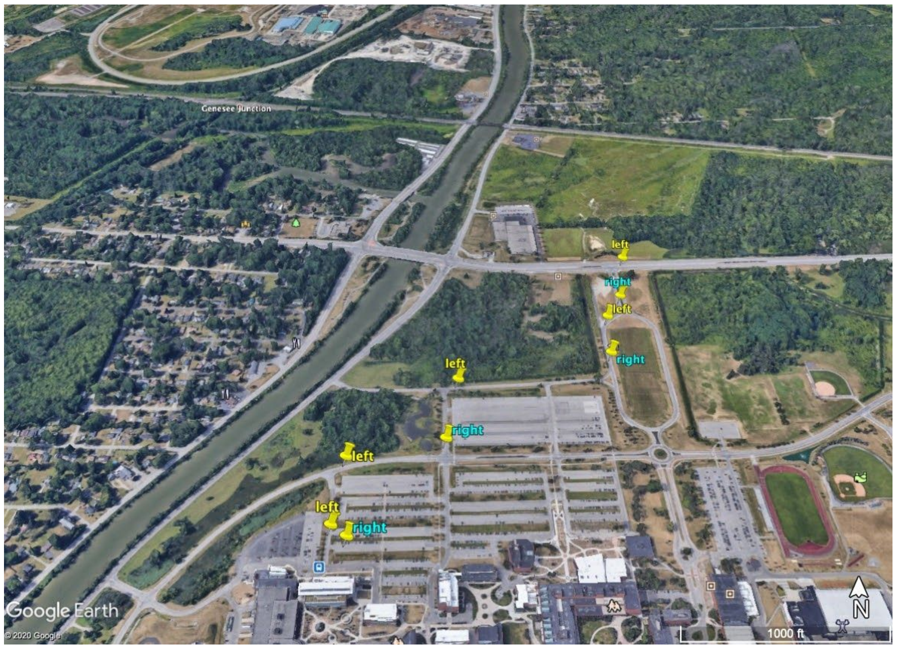


**Results of hazards file:**

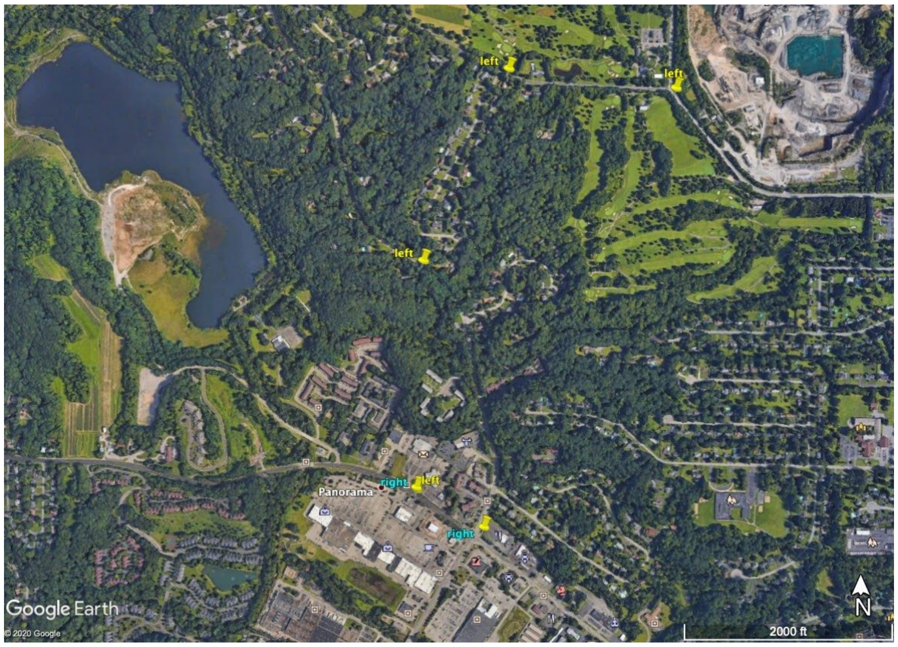
The hazards encountered in the previous image.(Zoomed out image for an overview)



Zoomed in part of the bottom left of the map for greater visibility(The placemarks are duplicated).

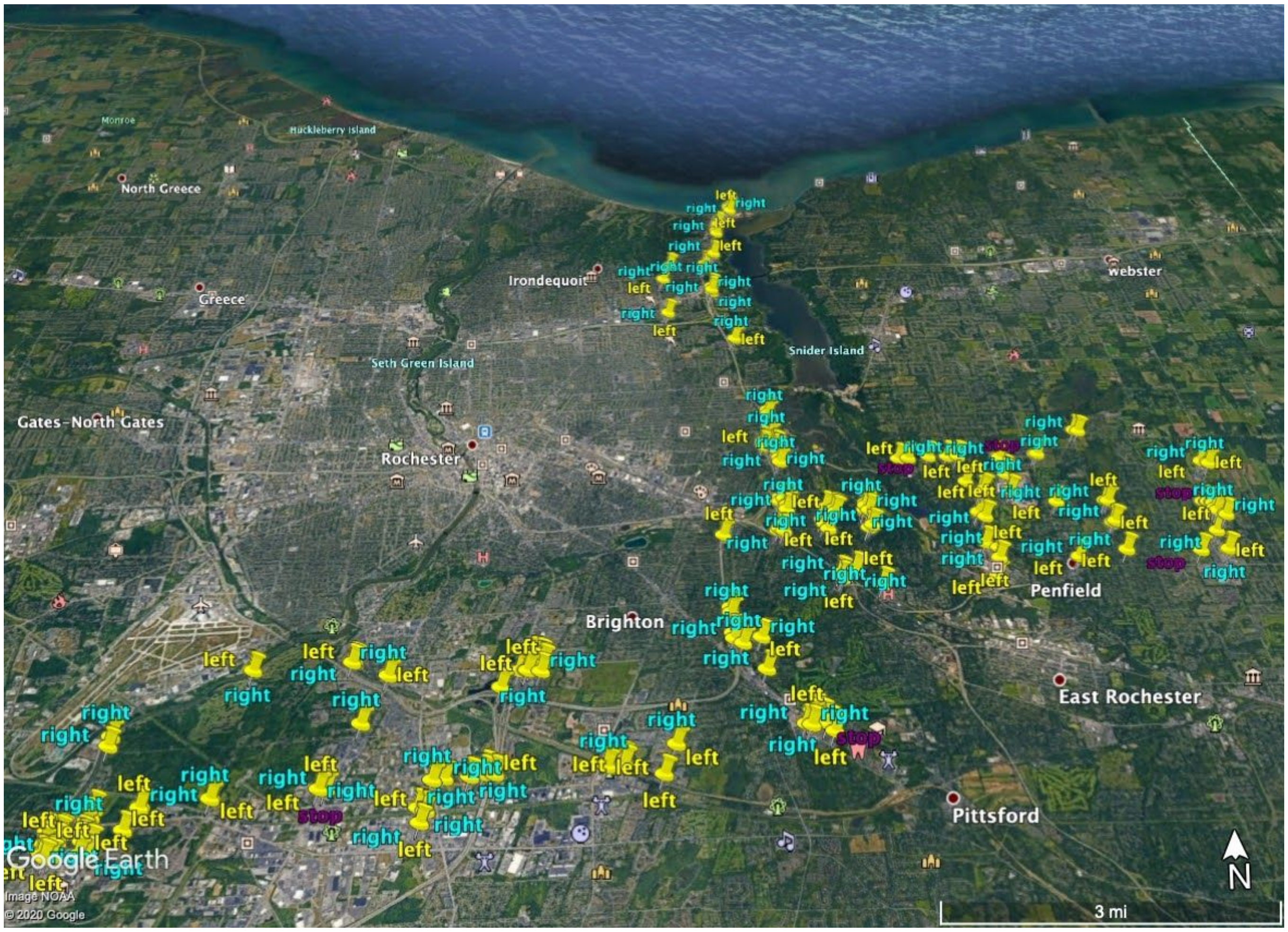


Zoomed in part of the bottom right of the map for greater visibility(The placemarks are duplicated).



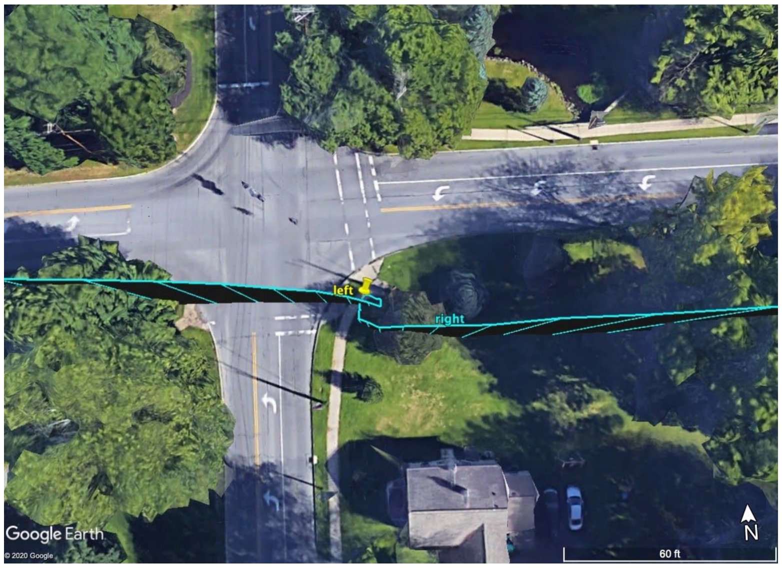
**Combined KML File**

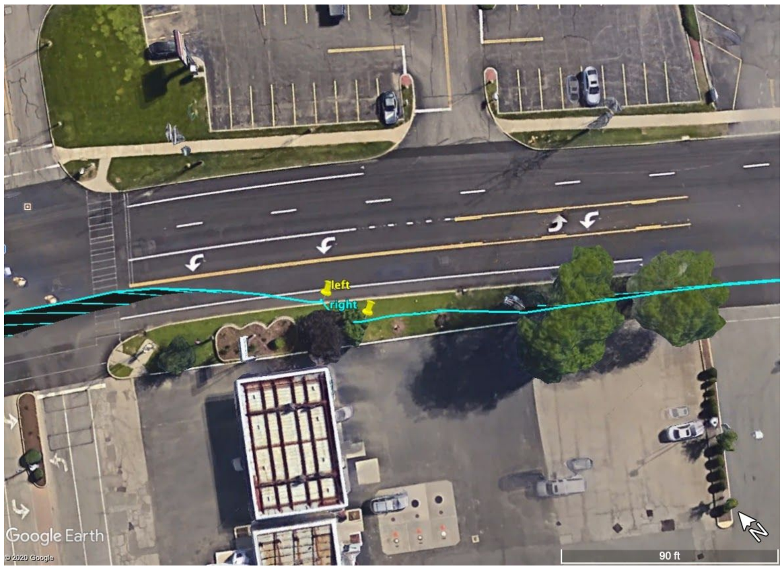
Zoomed in for greater visibility



**Discussion:**

While the classifier we built was able to predict most of the turns and stops correctly we still had some noise in our data even after our noise removal process. In the images below we can see lines which loop around(Arudino recording incorrect data) which causes our classifier to classify turns incorrectly. The noise removal techniques could not account for such kind of noise where the path suddenly changes direction, loops around and gets back to path in a matter of seconds. If we were to increase the time threshold to eliminate this noise we were also losing data when the turns actually existed. We also used Quality field


left
right
Google Earth
© 2020 Google
60 ft
N


left
right
Google Earth
© 2020 Google
90 ft
N

**Conclusion:**

We have understood how important it is to clean the data in order to do any processing, most of our work has been involved doing the cleaning part and then later processing it. Most of the points that Dr.Kinsman has mentioned in the project helped a lot during cleaning i.e. many redundant points during parking are useless as we know it's parked if we know speed is zero and time is greater than 120 seconds and so this reduces our complexity of processing as points get reduced. Also many of the entries in the given GPS coordinates were not appropriate so had to ignore entries so that they didn't hinder the later steps. After cleaning it all comes to either domain knowledge or through manually checking data and getting information data in order to build something on top of it.

We had gone through the data and got different parameters for building a good classifier that detected the stop point and turning point i.e. got to learn that if the speed is below 12mph then most probability the vehicle is either going to park or stopping for a signal or taking a turn. It's also possible that during turning the speed of the vehicle may be greater than 12mph so to make sure the angle used in order to check if a point is a turning point or not. Also used time to differentiate between the parking point and signal stop i.e. if the next point time is more than 120 seconds then most probably it's a parking point else it's a signal stop or sign stop.

The commercial application of such IOT data processing can be observed in many fields such as autonomous driving where self driving cars can identify incoming signals or turns using GPS data. Companies can also use this data to optimize their delivery routes based on the GPS data collected from the vehicles.

References:

1. Course Project document: It had the most of the good content which was provided by Dr.Kinsman.
2. https://en.wikipedia.org/wiki/Dilution_of_precision_(navigation) this link provided the dilution of precision details.
3. https://docs.novatel.com/oem7/Content/Logs/GPGGA.htm#GPSQualityIndicators this link provides details of quality.