UAS Collected Traffic Data Analysis Competition

Team TransLink: Bo Wen, Ahmed Youssef

1 INTRODUCTION

The data derived from Unmanned Aerial Systems (UAS) can provide very precise measurement of vehicle movements at major intersections and arterials. The detailed vehicle trajectories derived from UAS can provide additional insights on queue formation and spillback in real-time – information previously very difficult to collect. This data competition solution is focused on deriving information on queues using UAS vehicle trajectory data.

2 DATA PREPARATION

In this part of the code, we read in the UAS4T raw data with wide table format with varying number of columns into a long table format with set columns.

We also calculated the flat coordinates x and y using the latitude and longitude provided and the datashader package utility function "lnglat_to_meters". Performing this geospatial coordinate transformation simplifies any distance or clustering operations since we no longer need to work with great circle distances and can instead rely on Euclidean distances. This reduce compute time for distance calculations across the board.

Extra attributes were computed to assist with model development. We initially calculated the direction of acceleration using latitude and longitude acceleration vectors but found this attribute too unreliable. Instead, we used bearing derived from current and next location of the vehicle. We imputed the bearing of the last vehicle location for each track id with the second last vehicle bearing.

3 DATA EXPLORATION

We visualized various attributes using matplotlib, seaborn, datashader and holoview with GIS base map. We were able to visually establish roadway geometry based on point density variations in the entire data set. Here is a summary of visible difference in point density across spatial features of the dataset:

- Higher point density towards the centerline of each travel lane
- Higher point density near major roads with more traffic and congestion
- Lower point density for vehicle traces off road (parking, side streets, private properties

We also examined the speed and acceleration profiles of sampled track id, but we did not attempt to develop models to separate and cluster low and high moving vehicle speeds. This can be a useful measure in establishing a soft threshold for low speed moving vehicles in congestion.

Another aspect of the data competition we did not pursue is examining the difference in trajectory characteristics of different vehicle types. This can further refine queue measurement, but it proved challenging due to the different nature of these vehicle movements. Buses and taxis may stop at seemingly random points for passenger pick up and drop off, potentially causing congestion and queues, while motorcycles may meander between vehicles and make less centered or straight trajectory on a travel lane. Due to these complexities, we decided not to expand on these issues for this data competition in our model development.

4 Model Development

4.1 Road Segment Clustering

Due to the large number of points (over 6 million) in the dataset and lack of road or lane geometry information, we need to cluster the vehicle trajectory data to generate roadway geometry. We started by building a set of unique locations by frequency, filtered out infrequent points with less than 10 occurrences in the dataset. This helps spatially separate large segments of the roadway and reduce the initial size of the training data.

The intention of this step is to build chunks of network with sensitivity to the density of observed points. The desired output is roadways naturally divided at intersection traffic lights or major turning movement, where less dense points are expected due to less stopping in these areas. We performed 2-stage of HDBSCAN with the following parameters: *min_cluster_size* of 150 sample points for first stage, 75 sample points for second stage; *min_samples* for core points should be flexible for both stages; *cluster_selection_epsilon* of 5 meters for both stages.

As postprocessing, we reassigned un-clustered points into clusters using nearest neighbour distance criteria with a constraint of 20 meters. This ensures all points have a cluster to be trained on for lane clustering by segments. Some initial clusters generated were too small or sparse to be analyzed, we visually inspected these clusters and remove ones outside of areas of interests and combined suitable ones. Finally, we built a convex hull for each roadway geometry cluster. Then recaptured all the unique points from the dataset. This helps us prepare the data for road segment-lane clustering.

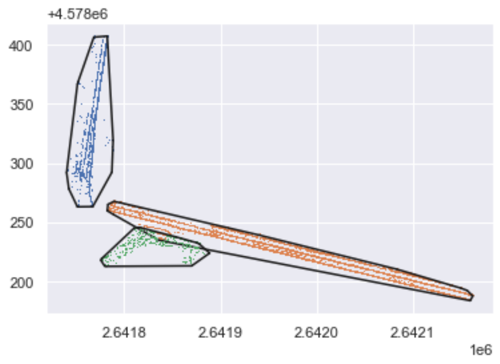

Figure 1: Convex hulls from Road Segment Clustering for selected Clusters

4.2 Lane and Directional Clustering by Road Segment

Using the dataset with recaptured points by convex hull, we ran DBSCAN clustering on location to separate the major movements, lane and directions of road segments, see cleaned up final clusters in Figure 2. We relied on visual inspection to further aggregate and tag smaller clusters into major road segment-lane clusters. Using these clusters, we built convex hull to recapture all data points falling within these clusters for queue analysis.

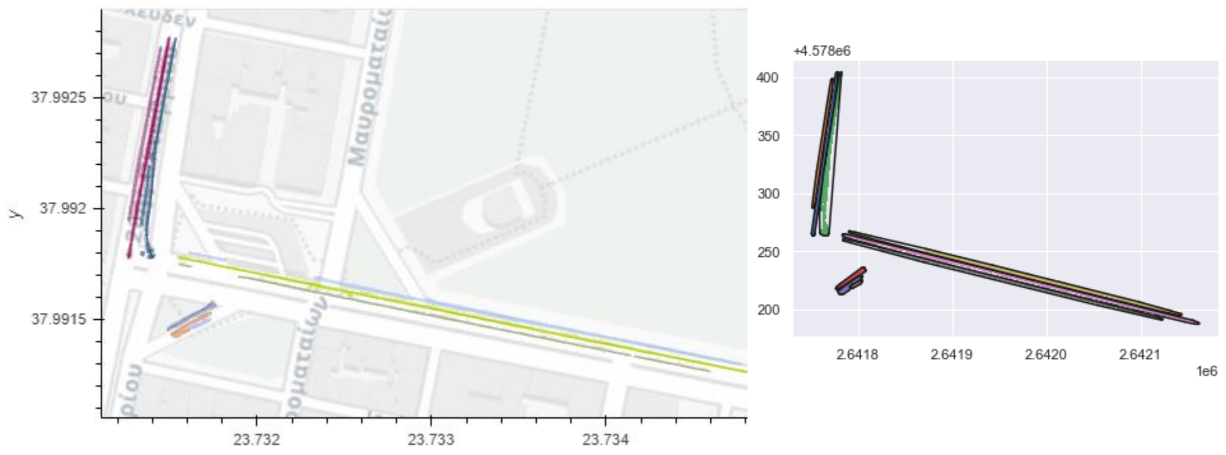

Figure 2: Directional Clustering for Road Segments results, left: point clusters, right: convex hulls.

5 Queue Clustering and Evaluation

To report on the maximum queue per cluster (road segment and lane), we prepared the data by applying a speed threshold parameter of 10 kilometers per hour to capture slow and stopped vehicles. In addition, we grouped the data within the road segment-lane using convex hull areas by cluster and time step (0.02s), then ran HDBSCAN. We used DBSCAN parameters of min_samples of 2, and cluster_selection_epsilon of 20, which are more suitable for finding queues within a small cluster.

The outlier clusters from the queue clustering were discarded from the dataset since the outliers represent points with no neighbours within 20 meters and may be stopped or parked vehicle outside of traveling lanes.

By finding the maximum distances between clustered queues, we can find the length of queues within each cluster at each time step. Then, to report on the maximum length of queue, start and end points of the queue, and time stamp, we summarized the queue length data by each clustered road segment and lane.

For more reference of the result as well as the detailed implementation using Python, please refer to the attached files:

- Python implementation in Jupyter Notebook: **uas4t_tl_team-revised.ipynb**
- Results data file: **uas4t_tl_team_results-revised.csv**