ITSC 2020 UAS4T Competitoin Report

Li Zeng, Mohammad Al-Rifai

September 11, 2020

1 Algorithm overview

We proposed an algorithm as shown in the Algorithm 1. Our idea is: for a fixed timestamp, we first assign the vehicle to the lanes based on the distance from the vehicle coordinates to the lanes, then for each lane we detect if there are queues by measuring the distance between the vehicles next to each other, at last step we calculate the length of the queue and report the longest queue.

The raw data consist vehicle coordinates every 0.04 seconds. We do not expect that a queue has a dramatic change with in 0.04s and thus we will first search the longest queue for every 3 seconds to get rough time $t_{max_{3s}}$ when the longest queue occurs. Then we fine seach every 0.04 second among the (-3,+3) seconds around $t_{max_{3s}}$ and get the final result. This allows the algorithm to avoid search the whole time domain and saved the computation.

Input : Time series vehicle trajectory data V, lane information data L
Output: longets queue q_{max} with information about time, GPS and so on.
get the the period of time T of S;
foreach *3 seconds* $t_i \in T$ **do**
    step1: extract all the vehicle at timestamp t_i and get a subset V_{t_i};
    step2: assign each vehicle in the subset V_{t_i} to the predefined lanes in L;
    step3: **foreach** *lane* $l_i \in L$ **do**
        detect all the queues and get Q_{t_i,l_i} ;
        calculate the longest queues and get $Q_{t_i,max}$;
    end
end
Compare all queues in $Q_{t_i max}$ and get the longest queue in all time $q_{max_{3s}}$;
Search from -3 seconds to +3 seconds around the time when $q_{max_{3s}}$ occurs, and get the longest queue q_{max}

Algorithm 1: Algorithm overview

2 Define the Lanes with coordinates

Based on the Google Maps for the interested area, we use the coordinates to represents the lanes. Each lane will be represented with at least two coordinates. For the lane that cannot be represented with a straight line. We divided lane in segments and each segment will be represented by two coordinates. As an example in Figure 1, the lane "far right 0" is defined with two red coordinates.

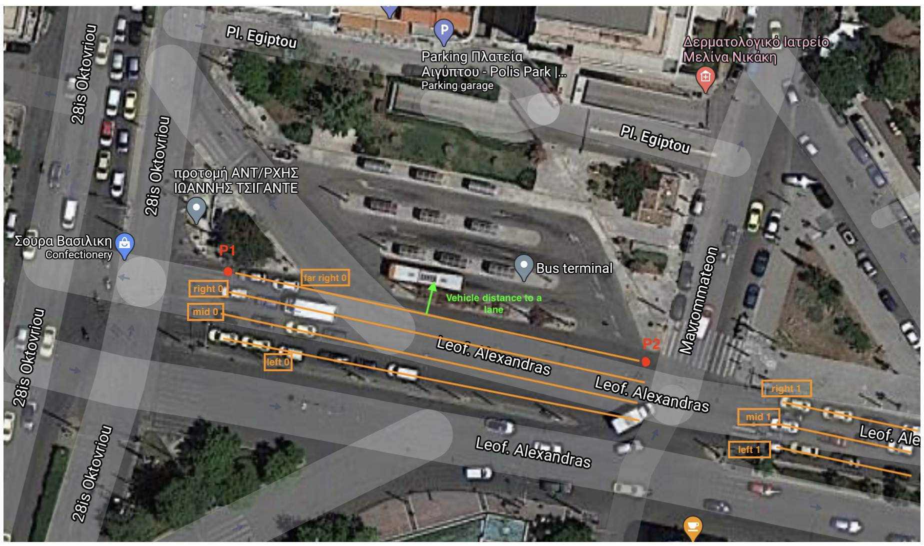

Figure 1: define lane based on GPS.

3 Assign vehicle to lane

Based on the coordinates of the vehicles and the Lane coordinates we assign vehicle to the lane. By the two steps:

- First, check if the vehicle locates between head and tail point of all the lanes
- Second, for the lanes that vehicle locates between, calculate the distance between vehicle and lanes, and assign vehicle to the lane that has minimal distance.

4 Detect and return longest queue

With the result from the last step, here for each lane we have a list of vehicles with their coordinates.

- First, sort the list of vehicles in the lane in ascending order with the distance to the head coordinate of the lane, we get a list of vehicles $[v_0, v_1, ...v_N]$
- Second, check the distance between the vehicles next two each others $Distance_{v_i,v_{i+1}}$, if $Distance_{v_i,v_{i+1}} < Threshold$, assign the v_{i+1} to the same queue with v_i, otherwise assign v_{i+1} to a new queue.
- Third, for all the detected queue that calculate the length of the queue and return the longest queue as an result.

5 Outcomes

overall the whole program takes roughly 20 minutes to run on a Macbook Pro with a 2.9 GHz Quad-Core Intel Core i7 CPU and 16 GB LPDDR3 RAM, this including import and process the datas, run the algorithm to get the results, and write result to csv file. and the results of this algorithm are showing in the following table.

Area	**Max len**	**Lane**	**start lat**	**start lon**	**end lat**	**end lon**	**Timestamp**	**spillback**
top 28is.	13	far left	37.991971	23.731369	37.992793	23.731535	340.4	FALSE
bot 28is.	10	mid	37.991401	23.731407	37.991569	23.731821	507.32	FALSE
Leof. Alex.	23	left	37.99139	23.733775	37.991587	23.732528	806.44	FALSE

Table 1: longest queue in three interested area