

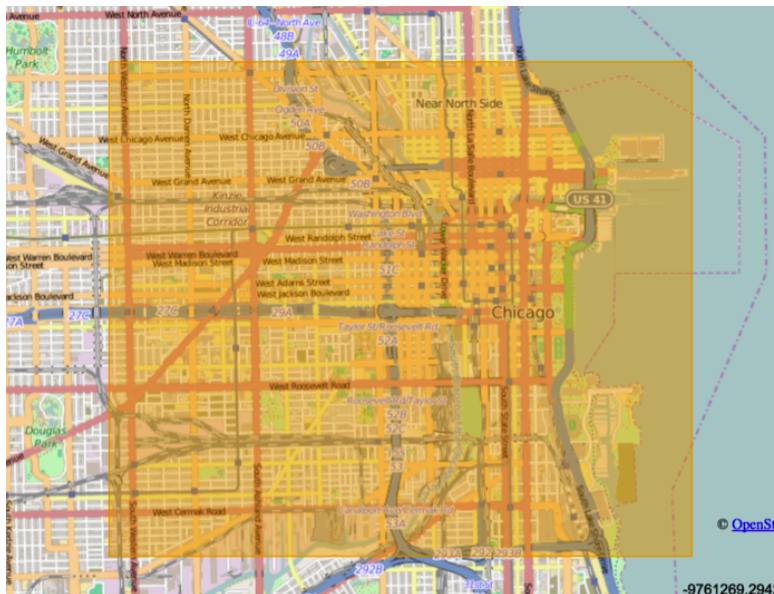
Generate Trajectories with POIs Inserted

Tian Zhang

The "POI.txt" is a modified version based on Di's data, just to make its format easy to run my script. I set the number of stop time units for coffee places [200, 250], restaurants [600, 650], fast food [50, 100], bars [500, 550], and theaters [900, 950].

There are several folders, and "500v120t" means trajectory data of 500 vehicles and 120 time units.

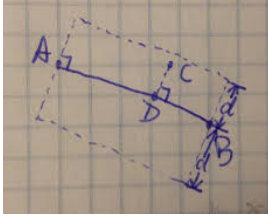
In each folder, there is one original trajectory file generated with [MNTG](#)'s random traffic model. The zone in which I generated trajectories with MNTG is shown as bellow (based on Di's zone data). The upper bound of MNTG is 5000v1000t, however, it actually doesn't work stably above 500v1000t. So I merged some original trajectories to get the ones with large factors (labeled "merged").



When merging the trajectories with same number of vehicles to the ones with more time units (Aggeragate_t.java), I connected the trajectories with same Object_id. For every two trajectories that need to be connected, I inserted 29 points uniformly spaced between the end point of the former and the begin point of the latter. When merging the trajectories with same number of time units to the ones with more vehicles (Aggeragate_v.java), I just aggregated the objects together.

The "Points" and "MoveStop" txt files were generated by MultiPoint_MultiTrj .java. The "Points" ones are the results after inserting some POIs to the original trajectories.

The logic I implemented when inserting a POI is shown in the picture below. Say A and B are two adjacent points on the original trajectory, and C is a POI. If 1) the distance between C and D is less than or equal to d and 2) D is between A and B, C will be inserted into the trajectory. d was set as 30 meters. When computing the distance between C and D based on there latitudes and longitudes, I used the method here <http://www.geodatasource.com/developers/java> .



To make the final trajectories simple, I didn't insert all the POIs every time. The number of POIs inserted into a trajectory is $[0, 1]$ for 200t, $[0, 2]$ for 1000t, $[0, 3]$ for 2000t and 3000t, and $[1, 4]$ for 4000t and 5000t. The exact numbers were randomly selected in the corresponding range. For example, for an object's trajectory of 3000v4000t, I first selected a number in $[1, 4]$. Assuming that I selected 3, and then 3 POIs would be randomly selected from all POIs that could be inserted into this trajectory. Furthermore, a POI could be inserted into a trajectory only once.

The "MoveStop" txt files were the aggregated version of "Points" ones. Each POI is a "stop", and between every two stops is a "move". If there is no POI in a trajectory, I would just pick the time midpoint and split the trajectory into two moves.