Emergency vehicle data map-matching utilizing the Dijkstra's Algorithm

Yanan Wu, Yalin Yang
Geospatial Information Sciences, EPPS, The University of Texas at Dallas

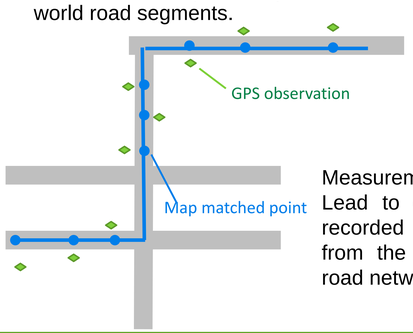
Introduction

Why emergency vehicle data?

- Goals
  - Examine whether traffic accidents hindered emergency vehicle dispatches
  - how spatial and temporal distribution of traffic accidents related to successful and unsuccessful dispatch rate
- Issues
  - Some GPS points off the road networks.
  - Low-frequency (sparse) GPS points along trajectories
  - Too dense GPS points cause strange trajectories

Why and what is map matching?

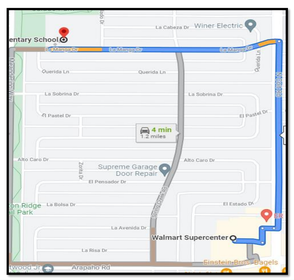
- Matching the GPS raw points (e.g., a sequence of (lat, long) coordinates) to the spatial network from the real-world road segments.



Measurement errors:
Lead to deviations in the recorded GPS coordinate from the true positions in road network

Why and what is Dijkstra's algorithm?

- Finding the shortest path between two given locations
- In the algorithm, it considers the road intersections, end or start points as nodes
- Roads that connect nodes are considered as edges (routes)
- The length of a path is considered as the movement cost



Study Area

- Area

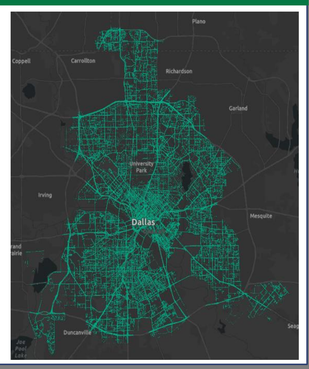
Total area of 385.8 square miles

- 340.5 square miles is land
- 45.3 square miles is water

- Population

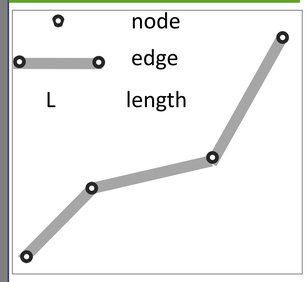
1,343,573 estimated in 2019

Ninth-populous city in the U.S



Data

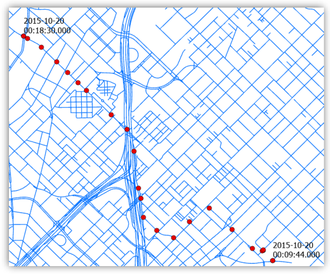
Road network



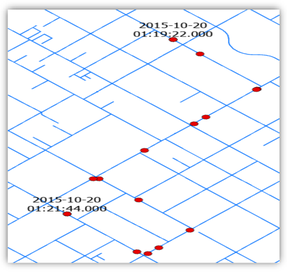
Data description

- GPS Points (2015-2017):
  - 532,653 trajectories in total
  - Thirty million GPS points in total
  - Timestamp
  - Heading
  - Speed
- Complex road networks
  - Complicated intersections
  - Multi-layer roads
- Simple road networks

Complex GPS raw points



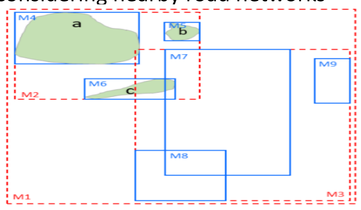
Simple GPS raw points



Methodology

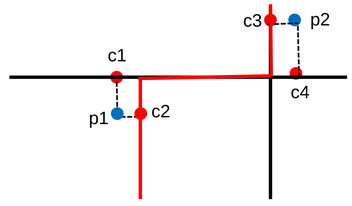
R-tree spatial indexing

Goal: Reduce the computation by only considering nearby road networks



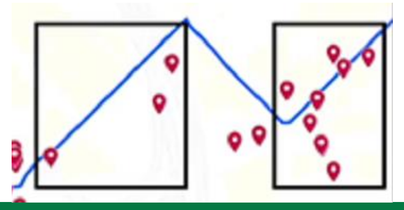
Candidate search

Goal: Candidate graph to represent multiple candidate matched paths



Dijkstra's Algorithm with constraints

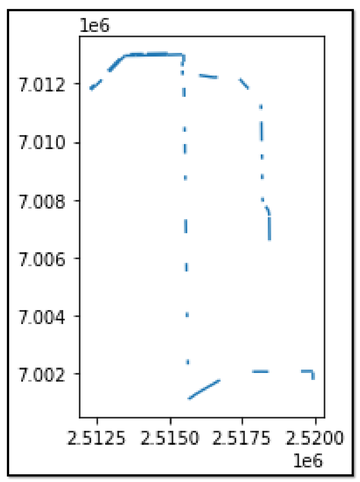
Goal: Solve the arc-skipping problem
Constraints: speed and search range



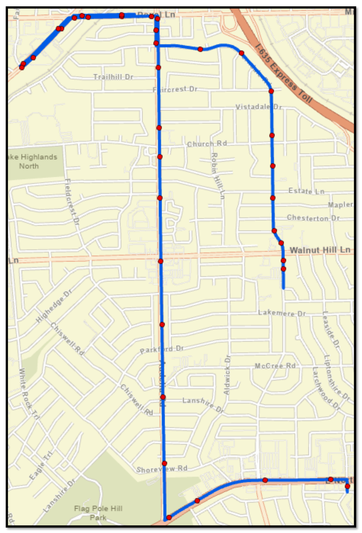
Results

Simple GPS raw points

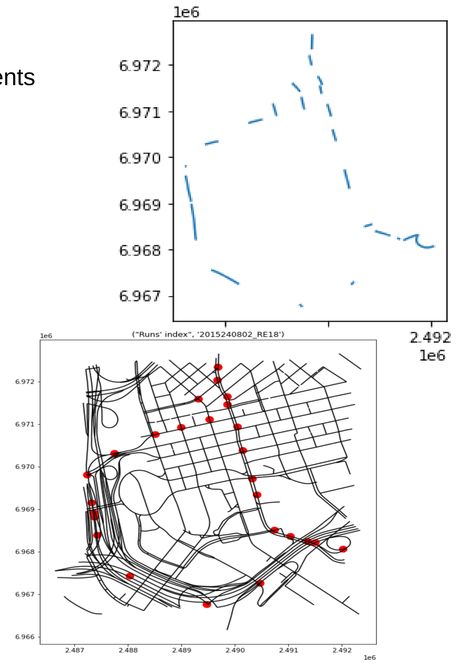
- Before Dijkstra's Algorithm
  There are several missing road segments



- After Dijkstra's Algorithm
  Successfully trace all the road segments



Complex GPS raw points



Results

Complex GPS raw points

Highway Intersections

Roundabout road

Conclusions

- The map-matching algorithm works well for capturing the trajectory path of simple GPS row points
- Dijkstra's Algorithm
  - Dijkstra's algorithm is limited to the highway intersection or roundabout.
  - Considering the direction of road.
  - The time required to find the optimal path becomes long when the search scope is broad.
  - Improve efficiency.
- The algorithm is implemented in Python, which is friendly for programming beginners.
- Validating by the results generated by Snap on Road API.

Future work

- Embrace the probability model for road candidates' selection
- Using the Hidden Markov model for best route calculating

References

Algizawy, E., Ogawa, T., & El-Mahdy, A. (2017). Real-time large-scale map matching using mobile phone data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *11*(4), 1-38.

Taguchi, S., Koide, S., & Yoshimura, T. (2018). Online map matching with route prediction. *IEEE Transactions on Intelligent Transportation Systems*, *20*(1), 338-347.

Zhang, D., Guo, Z., Guo, F., & Dong, Y. (2021). An offline map matching algorithm based on shortest paths. *International Journal of Geographical Information Science*, 1-24.

Zhang, X., & Du, Z. (2017). Spatial Indexing. The Geographic Information Science & Technology Body of Knowledge (4th Quarter 2017 Edition), John P. Wilson.