Fakultät für Bauingenieur- und Vermessungswesen
Lehrstuhl für Kartographie
Prof. Dr.-Ing. Liqiu Meng

Algorithm Design and Implementation of Map Matching of City-wide Floating Car Data

Technische Universität München

Master's Thesis

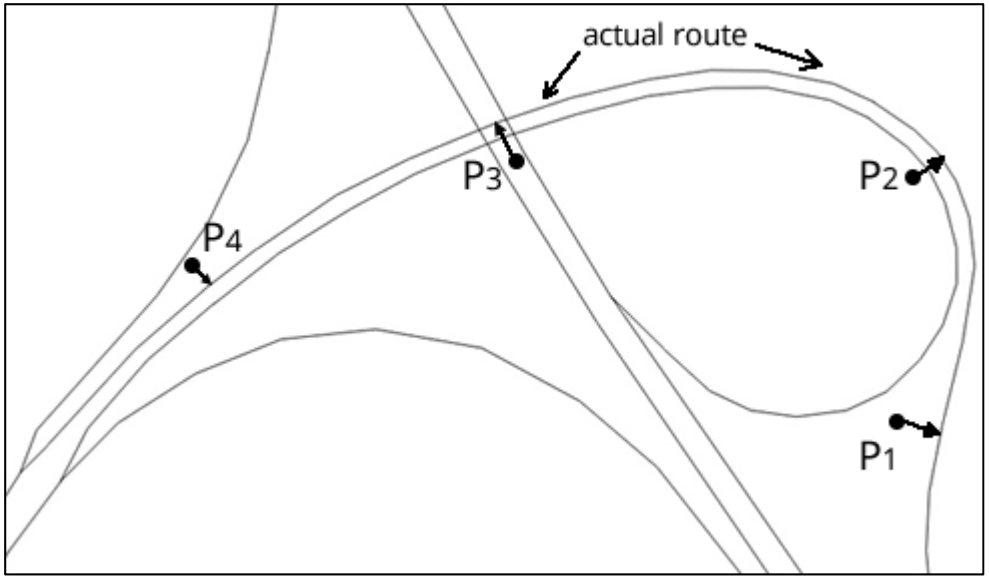
Author: Luyi Han

Supervisor: M.Eng. Jian Yang

Table of Content

Problem Statement

Positioning data is one of the sufficient data to find, analysis and solve the problems in transpiration system. The GPS data contains various measurement errors, and need to be preprocessed before being used.



Map matching definition:

Map matching is the process that serves the purpose of recovering the original route on a road network from a sequence of noisy GPS observations.

- Jian Yang, Liqiu Meng

Problem Statement

Float Car Data (FCD) is gathered by GPS devices installed in the vehicles (normally taxis).

- Hugh amount and quickly updated
- Noisy, sparseness and not well organized

Since 1970s, various map matching algorithms based on different techniques have been developed. But most of them have focused on map matching either experimental data with a data collection under control or small real world dataset with arbitrary spatial distribution.

None of the previous work has proposed a complete and clear work flow of map matching.

Map Matching Classification

- Executing time : Real time, Post-processing

  Real time: car navigation
  Post-processing: transportation system analysis with large amount of data

- Calculation factors: Topological and probabilistic methods

- The range of trajectory in each calculation: Incremental and global methods

Map Matching Classification

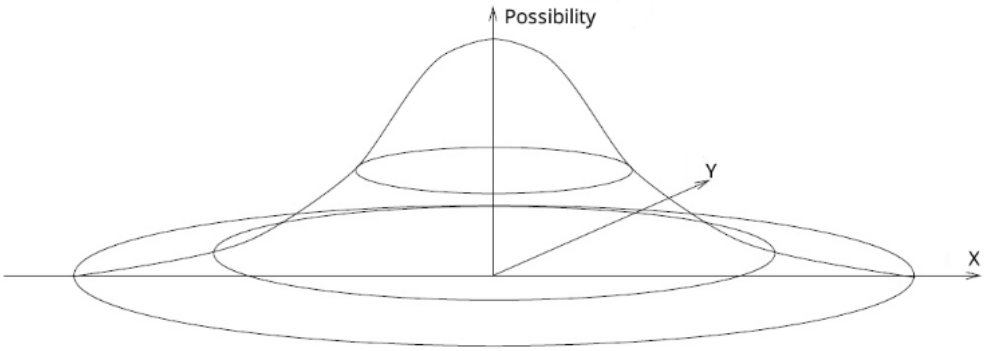
- **Topological methods**

The topological map matching algorithm makes use of the geometry of the links as well as the connectivity and contiguity of the links.

- Compare the shape of the route
- Use heading, proximity, link connectivity and turn restriction information

- **Probabilistic methods**

The probabilistic algorithm requires the definition of an elliptical or rectangular confidence region around a position fix obtained from a navigation sensor.



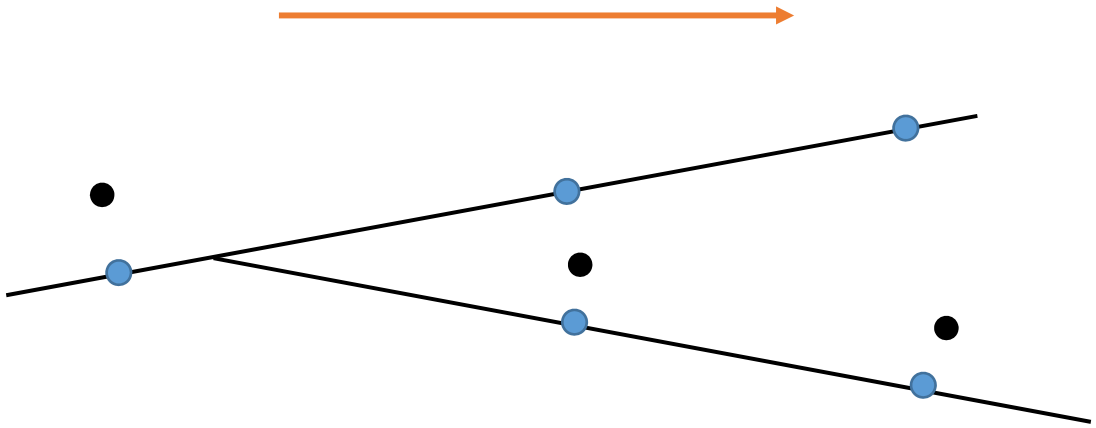
Map Matching Classification

- **Incremental methods**

The incremental algorithm iteratively calculates the next max-weight candidate based on the current best candidate or a summary of all previous samples.

- **Global methods**

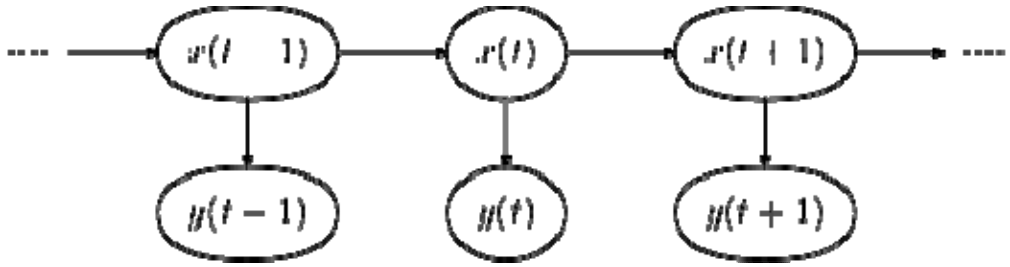
The global algorithm tries to calculate the max-weight values of the whole trajectory, and iteratively go back to get the candidates that can get the max-weight values.



Incremental methods Global methods

Definition

A HMM is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states.



Source: wikipedia

Hidden Markov Model based map matching algorithm

In map matching category, the states of the HMM are the individual road segments, and the state measurements are the geographical locations measured by the GPS sensor, and the vehicle's movement along the road segments from point to point describes the states transition.

Measurement possibility: the circle distance between the trajectory point to the road segment
Transition possibility: compare the circle and route distance between successive trajectory points

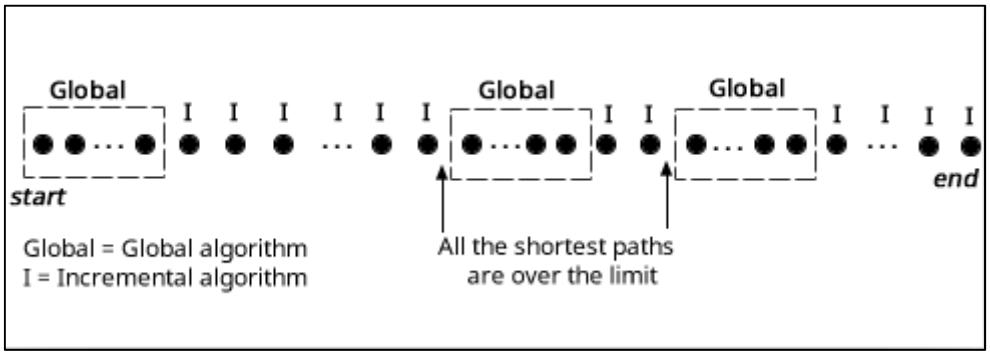
Newson, Lou (Microsoft)
Wei (HP)

Fusion Map Matching algorithm

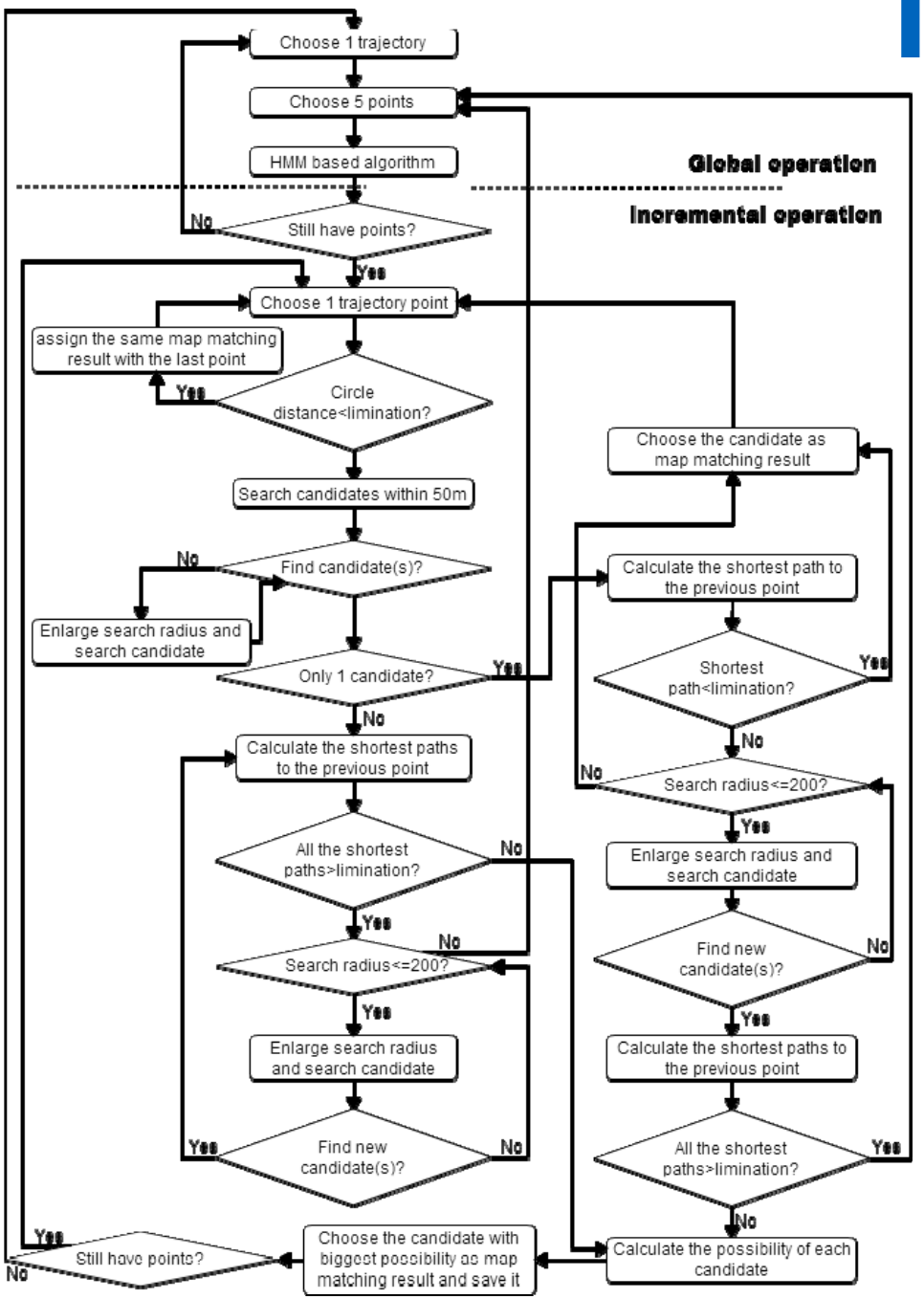
"Fusion" means the alterations between the global and incremental algorithms.

At the beginning of each trajectory

During the running of incremental algorithm



FMM workflow



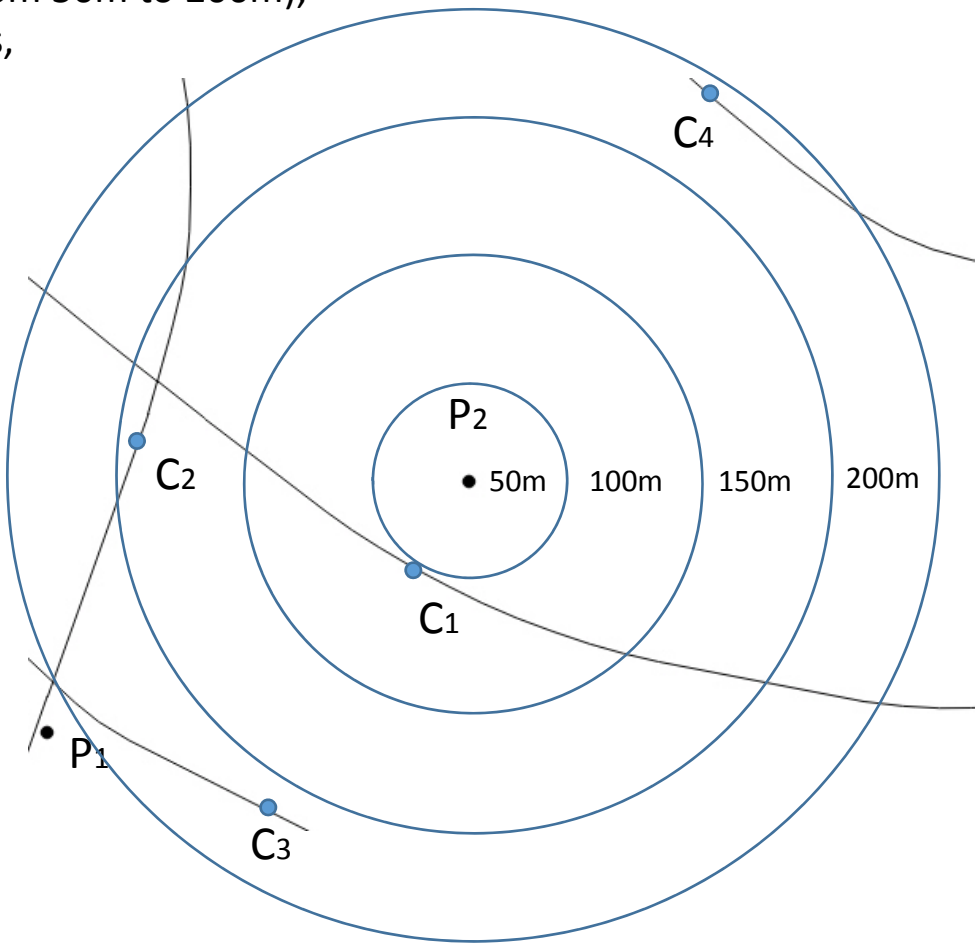
Candidate Preparation

In most of the exist algorithms, the search radius is defined by:

- a fixed maximum value (normally from 50m to 200m),
- the maximum number of candidates,
- both

In FMM:

- Dynamic
- 50m to 200m



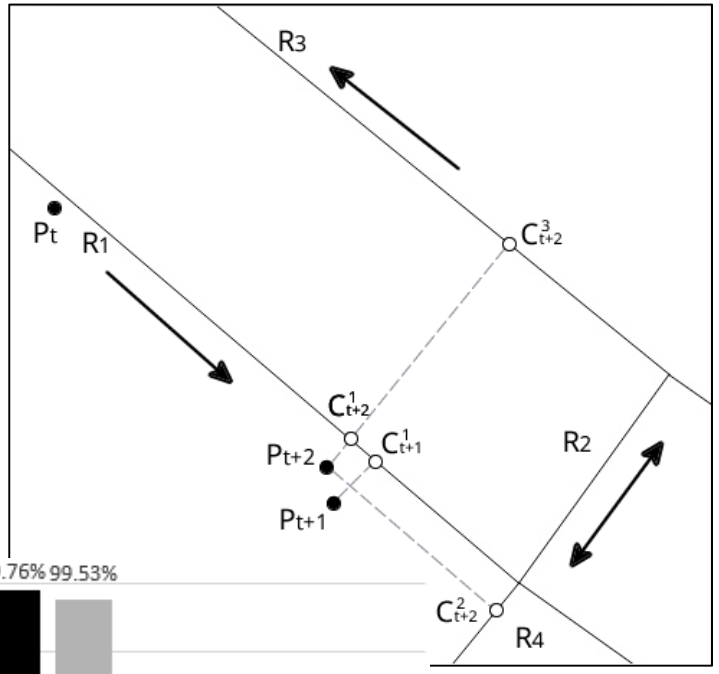
Handling Special Cases

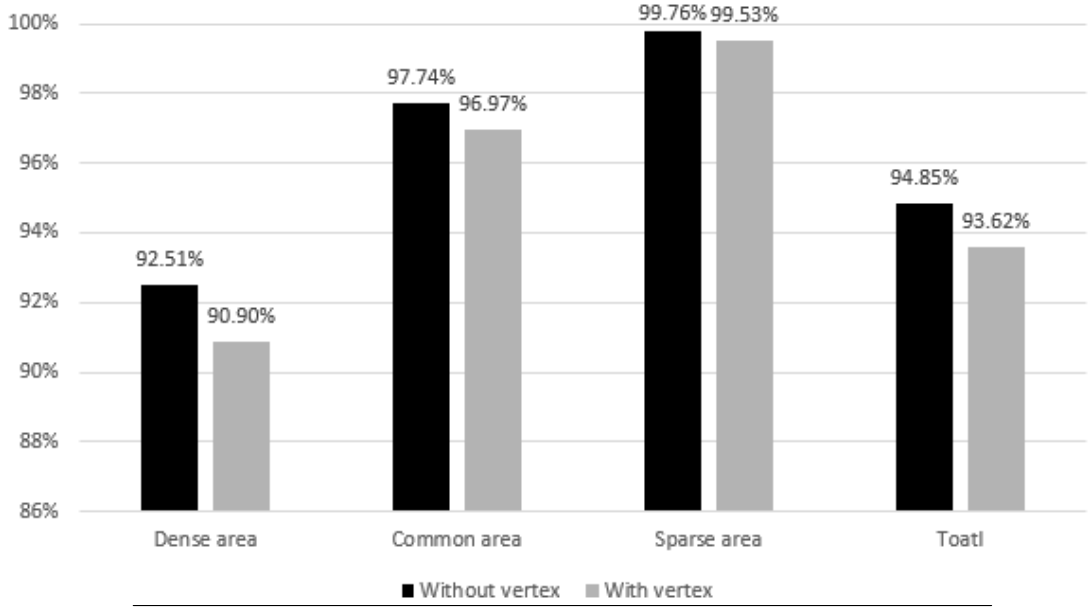
Only one candidate found

Successive points within a distance

Candidates on the vertexes of the road segment

47257/108426(43.85%) candidates are on vertexes, only 0.2% are correct matching results





Global Operation – HMM based algorithm

Measurement possibility

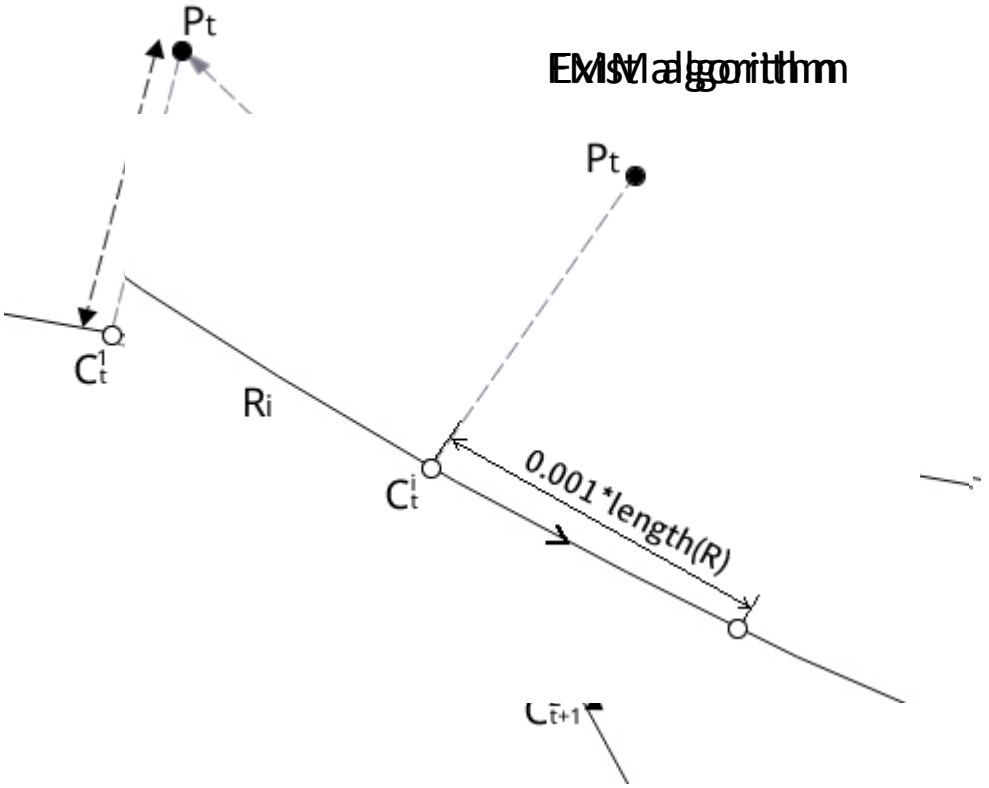
$$p_m(C_t^i) = \frac{1}{\sqrt{2\pi}\sigma_p} e^{-0.5\left(\frac{|P_t - C_t^i|}{\sigma_p}\right)^2}$$

Transition possibility

$$p(d_t) = \frac{|P_{t+1} - P_t|_{great\ circle}}{|C_{t+1}^i - C_t^j|_{route}}$$

Azimuth possibility

$$p_a(C_t^i) = \cos^5 \theta$$



Global Operation – HMM based algorithm

$$\arg\max \prod_{t=0}^{n} \left(\frac{1}{\sqrt{2\pi}\sigma_p} e^{-0.5\left(\frac{P_t - C_t^i}{\sigma_p}\right)^2} \frac{|P_{t+1} - P_t|_{great\ circle}}{|C_{t+1}^i - C_t^j|_{route}} \cos^5\theta \right)$$

Transfer the original algorithm to a logarithm type to avoid the underflow problem:

Azimuth possibility: $p_a(C_t^i) = \cos^5\theta \quad \Rightarrow \quad p_a(C_t^i) = -10 \times \ln(\cos^5\theta + 1)$

$$\arg\min \sum_{t=0}^{n} \left(0.5\left(\frac{P_t - C_t^i}{\sigma_p}\right)^2 - log\left(\frac{|P_{t+1} - P_t|_{great\ circle}}{|C_{t+1}^i - C_t^j|_{route}}\right) - 10 \times \ln(\cos^5\theta + 1) \right)$$

Incremental Operation

Uses the same definitions as HMM based algorithm

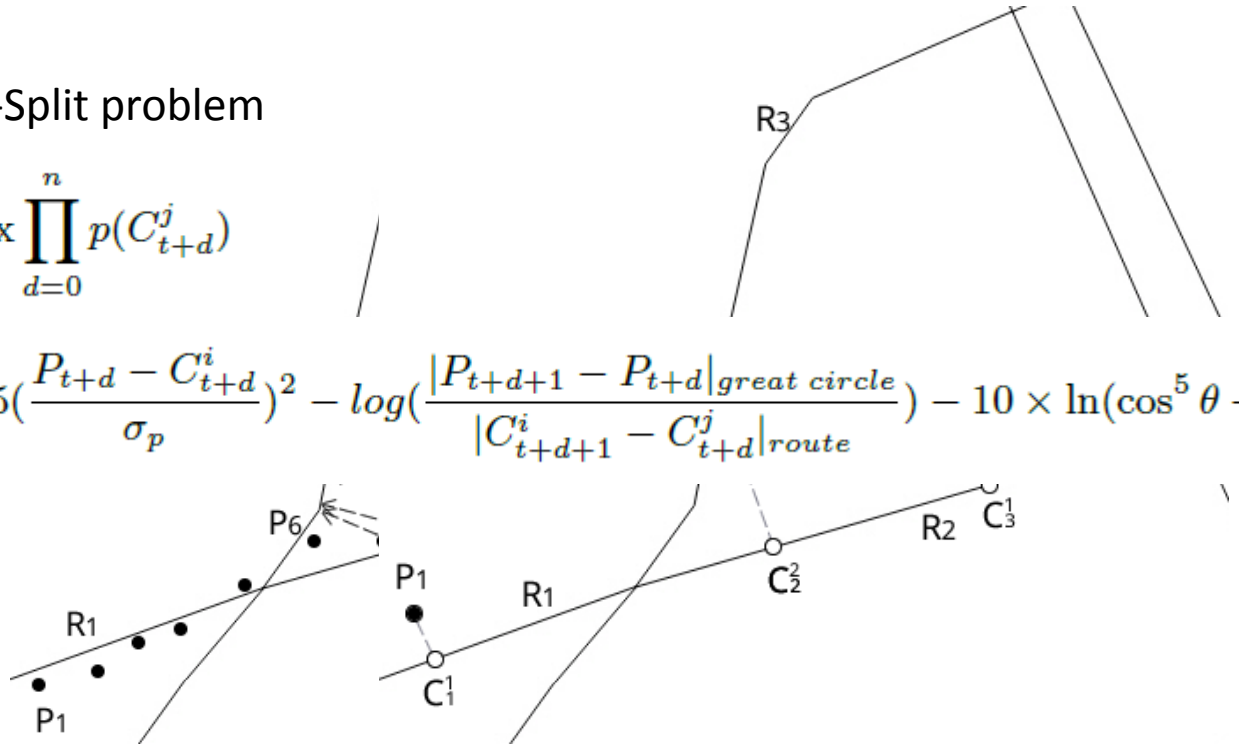
$$max_{(i=0\ to\ n)}\left(\frac{1}{\sqrt{2\pi}\sigma_p}e^{-0.5(\frac{P_t-C_t^i}{\sigma_p})^2}\frac{|P_{t+1}-P_t|_{great\ circle}}{|C_{t+1}^i-C_t^j|_{route}}\cos^5\theta\right)$$

Forward-looking Incremental Algorithm

Y-Split problem

$$p(C_t^i)=\arg\max\prod_{d=0}^{n}p(C_{t+d}^j)$$

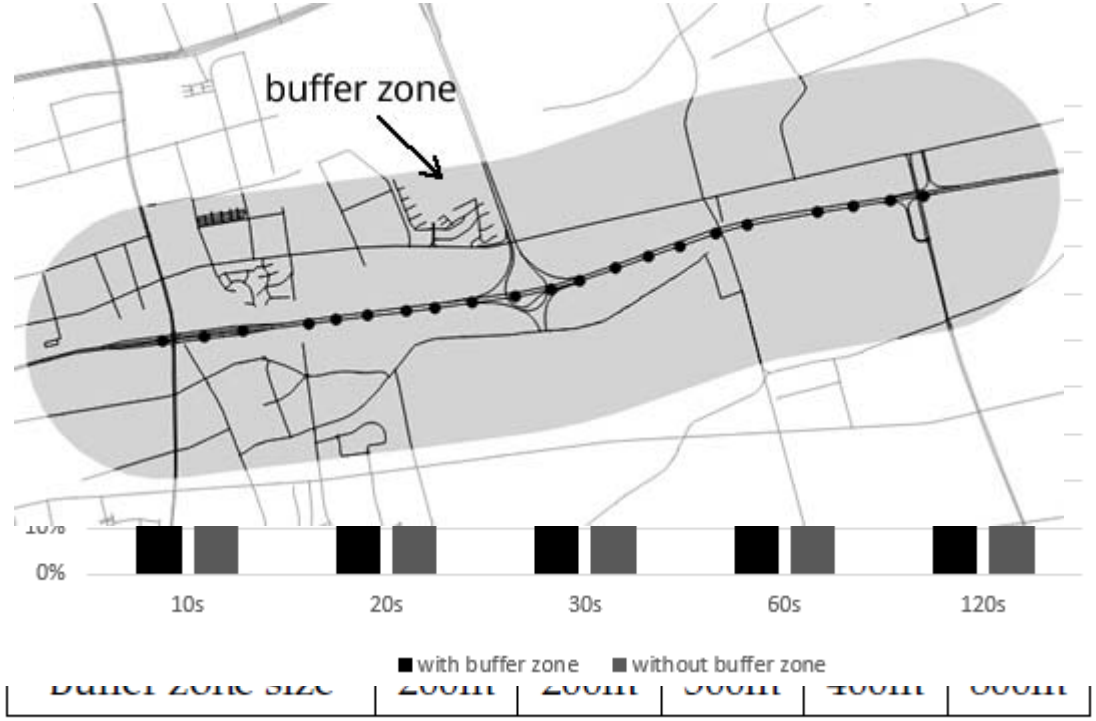
$$p(C_t^i)=\arg\min\sum_{d=0}^{n}\left(0.5(\frac{P_{t+d}-C_{t+d}^i}{\sigma_p})^2-log(\frac{|P_{t+d+1}-P_{t+d}|_{great\ circle}}{|C_{t+d+1}^i-C_{t+d}^j|_{route}})-10\times\ln(\cos^5\theta+1)\right)$$



pgRoutingin PostgreSQL

Dijkstra algorithm
Can only calculate the shortest routes between the vertexes of the roads
New function is designed to calculate the routes between two arbitrary points.
The Most time consuming procedure in the HMM and FMM algorithms.

Optimization



Work Flow of Map Matching FCD

FCD Validation

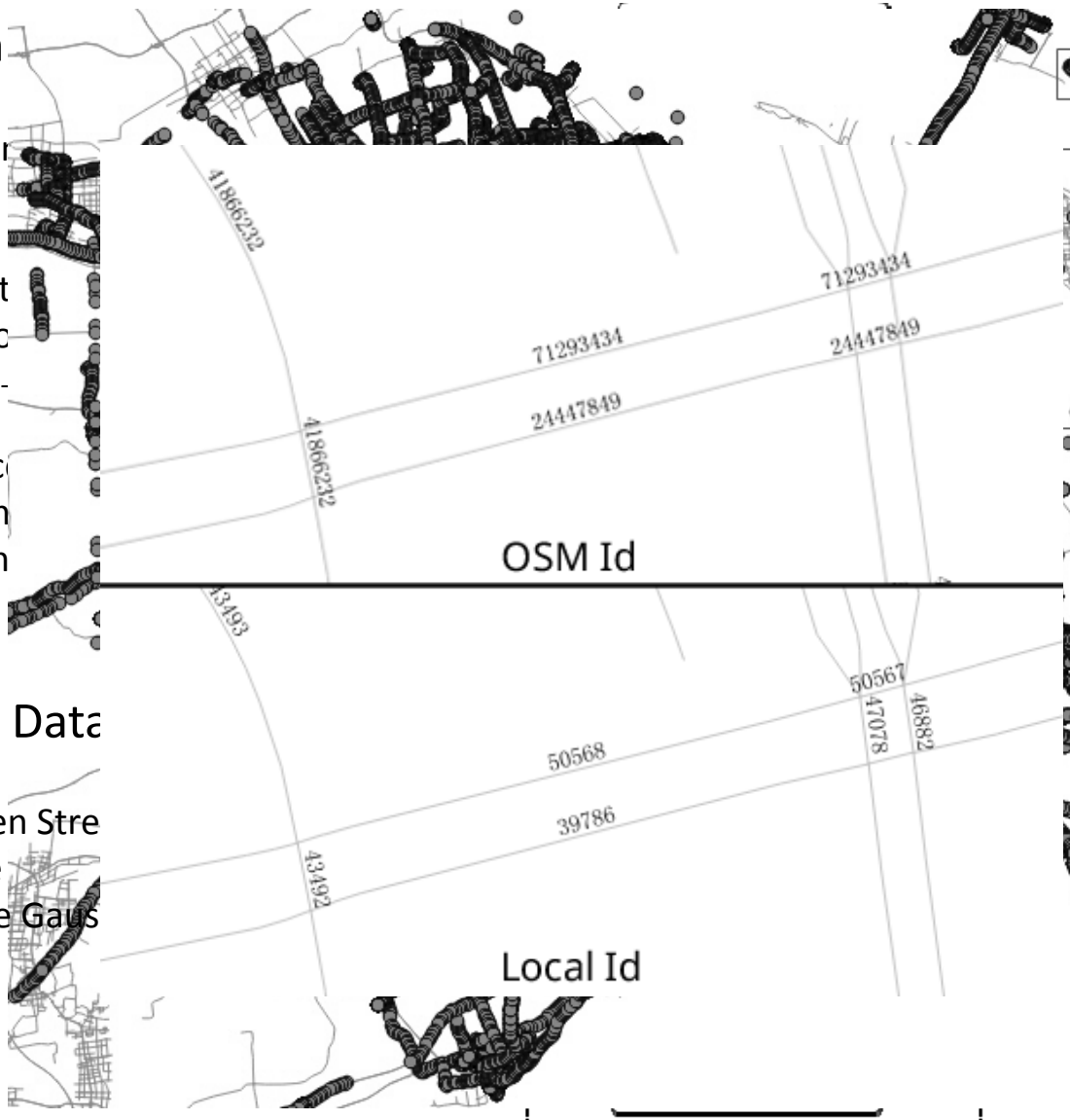
1st level: check the r
attributes.

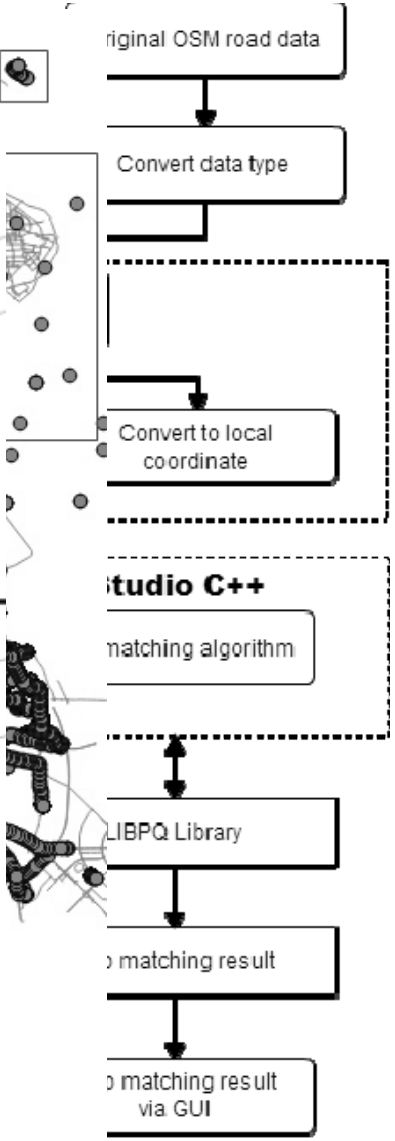
2nd level: delete out
convert cc
Xian 1980/3-

3rd level: delete suc
speed 0 an
delete unn

Road Network Data

Gathered from Open Stre
Convert coordinate
Xian 1980/3-degree Gaus





Shanghai FCD Overview

20703863 GPS trajectory points
6973 taxis
From 20:30:00 31.03.2010 to 20:00:00 01.04.2010
10s sampling interval

After data validation, 69% of data remains.

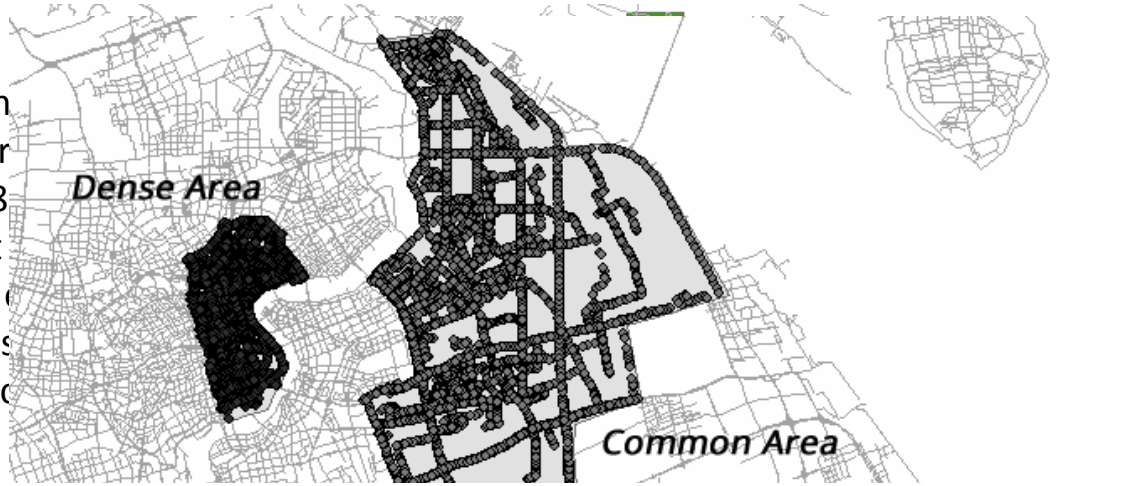
Fields	Type	Description
id	integer	unique ID of the trajectory point,ie. 1
car id	integer	unqiue ID ededof the car,ie. 18384
time	timestamp without time zone	Date and time of gps measurement.In format of '2010-03-31 22:28:00'
geometry	gemotry(point,2385)	geometry value of the GPS points
driving direction	integer	driving direction of the car, ranges from 0 to 355, with 0 pointing north, and increase clock wisely

Test Data

Derived from Sh
Divided into thr
dense area (>=8
Choose the test
the boundaries
Randomly choos
points in each s

ay or arterial road as

n of the total



	Grid Number	Training Area (km^2)	Original Points	Test Points
Dense Areas	7	30.76	1675638	13884
Common Areas	52	230.01	1121043	9749
Sparse Areas	119	524.18	118831	849

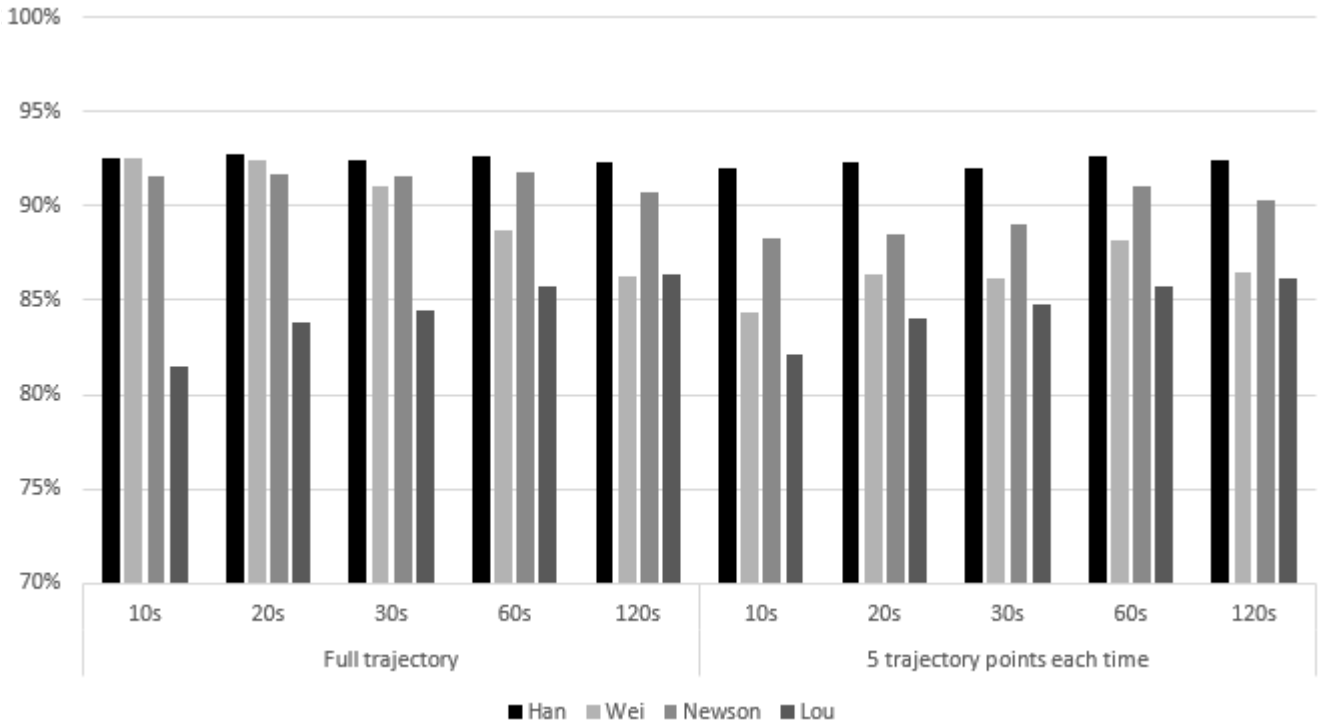
Create test data with different sampling intervals

	20s	30s	60s	120s
Dense Area	7995	5802	3301	1807
Common Area	5625	4108	2315	1261

Experiment Results

The accuracy performance of a map matching algorithm is defined as:

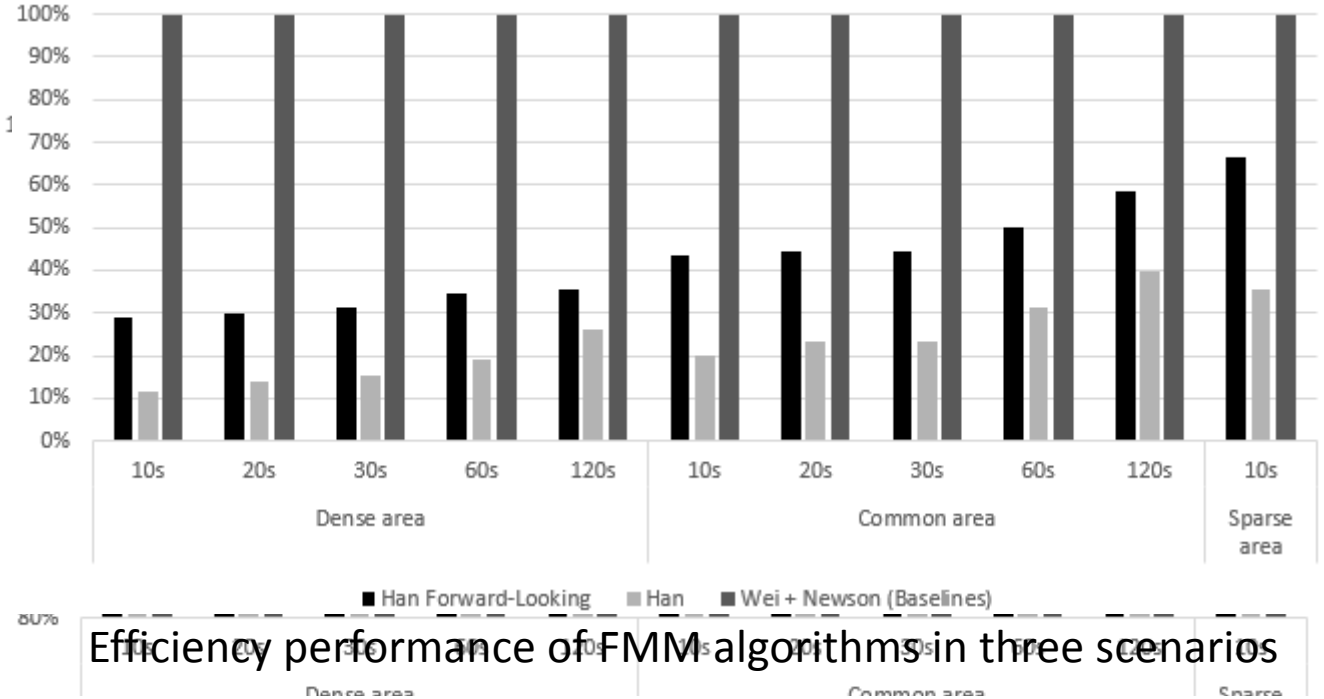
$$\frac{number\ of\ points\ assigned\ with\ correct\ local\ road\ segment\ id}{number\ of\ total\ trajectory\ points}$$



Accuracy performances of HMM based algorithms with full trajectories and 5 points each time

Experiment Results

Baselines are made by the exist HMM based algorithms with best performances in different sampling intervals and scenarios.



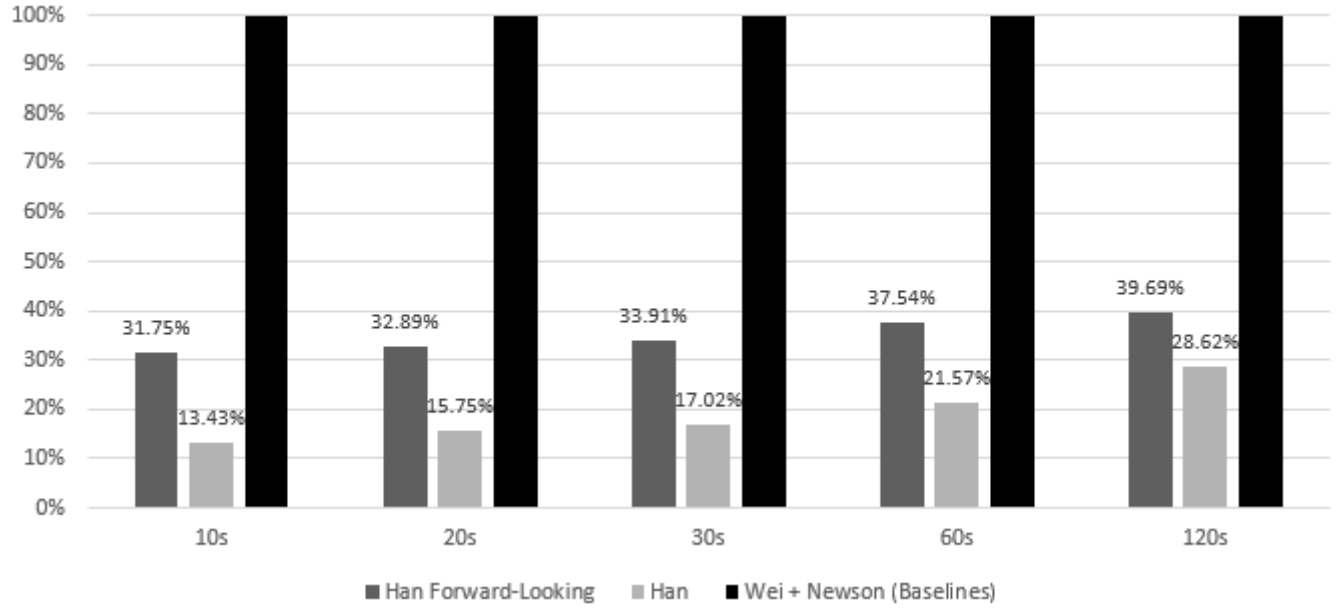
Efficiency performance of FMM algorithms in three scenarios

HMM	Normal incremental	F-L incremental	Normal FMM	F-L FMM
mn^2	mn	$4mn$	$0.1mn^2 + 0.9mn$	$0.1mn^2 + 3.6mn$

Accuracy performance of FMM algorithms in three scenarios

Calculation complexities of HMM algorithms in large scenarios

Experiment Results



The efficiencies of FMM algorithms

Conclusion

- Fusion map matching algorithms are presented
  - Better accuracy performances
    - 95.31%(Forward looking) 95.16%(Normal)
    - over 2% improvement
  - Better efficiency performances
    - 32%(Forward looking) 14%(Normal) running time
- A workflow of map matching FCD
  - From data preparation to presentation
- Test data consider the distribution of FCD in three scenarios
  - Test results is robust for the city-wide data

Future work

- Improve the efficiency performance
  - Multi-cores parallel calculation techniques

Thank you!