Tanzeel Ur Rehman KHAN

Comparison of map matching algorithms with floating car data (GPS)

Duration of the Thesis: 6 months

Completion: December 2012

Tutor: Dipl.-Ing. Jürgen Schweitzer

Examiner: Prof. Dr.-Ing. Volker Schwieger

Background

The Map matching process is to find the current vehicle location by means of digital map, as a result map matched position on the digital road network calculated from estimated position and corresponding to vehicle actual location. In a multi sensor positioning system different sensors are used like GPS, DGPS, Gyro, and odometer. Every sensor has its own error ellipse and accuracy constraints. This can be compensated by the map matching algorithms. Three kinds of inputs are used

i. Positioning data (sensor data)
ii. Digital map
iii. Map Matching algorithm

Floating Car Data can be used for basic vehicle telemetry, GPS data from the Smartphone is used as a position data and two other GPS data sets (Exemplary) are used for comparison and analysis. In this master thesis different map matching algorithms are developed in python scripting language as a script and then it can be run in Arc GIS 10 software for evaluation and analysis of the results. The first algorithm is the simplest algorithm which determines the matched coordinate to the nearest distance to the edge (street). In second algorithm velocity information is also obtained. There are two algorithms of map matching with coordinate pattern; the evaluation procedure depends upon the sensor used. Similarity (4 parameters) and affine (7 parameters) approximations are used. The aim is to find the best allocation of the position data with respect to some accuracy constraints (Residuals and standard deviation) by map matching algorithms. Comparison and evaluation of different data sets with different map matching algorithms is also the part of the research.
Software and data used.

In context of floating car data, GPS data is used which is taken by smart phone. Meanwhile three exemplary data sets are used for the analysis. Which are created by own in ArcGIS software to check the accuracy and implementation of the map matching algorithms.

Figure 1. GPS data (original)
Another input data is prepared which is used in the process of map matching,

Streets are generated from the base map (open street map) and this data is stored in the shape file format. This data is needed in the process of map matching. It has different attributes as shown in the figure above.

The basic goal of the master thesis is the development and implementation of map matching algorithms using positioning data from the GPS sensor and streets data. This implementation would be in the ArcGIS script language python. And the results would be an ArcGIS toolbox (in which different algorithms will be implemented in the form of script.). For each algorithm, there would be scripted in python. Which is capable to implement the map matching algorithms.
ArcGIS 10 software is used to implement the python script in arc tool box. all script are made in python programming language.

**Methodology**

There are some algorithms are defined to perform map matching. How they are categorized.

1. Map Matching with Nearest distance
2. Map Matching with Nearest distance with extended velocity information
3. Map Matching with similarity transformation (co-ordinate pattern)
4. Map Matching with affine transformation

First algorithm is the simplest algorithm of map matching. Where GPS point data is matched to street data on the basis of minimum distance. This algorithm defines the simplest way to find the location of the vehicle on the digital map.

![Nearest distance algorithm](image)

Figure 5. Nearest distance algorithm

Sometimes this algorithms matches the GPS points on the wrong street network which creates error in map matching algorithm and it is not defined a correct rout, so that, why velocity information is incorporated.
Now referring the figure, these are the matched points and GPS points. It can be seen that arrow is defining the route of the vehicle normally but due to algorithm limitation (matched to the nearest distance to the edge) point B is matched to another street. So the velocity between the GPS point A and GPS point B is very high. Algorithm matches the GPS point to another nearest street edge which would be automatically on the second nearest edge. This is the correct route of the vehicle. The algorithm will choose another street edge if the difference of the both velocities is greater than the threshold. So this algorithm depends upon the velocity information. Threshold velocity can be considered from the feature class velocity which is given in the attribute table of street data.

**Algorithm flow chart with coordinate pattern (Similarity, Affine)**
As described above all allocation method with observed coordinate’s divides a measured trajectory in line elements of constant length (e.g. 1m) by interpolation and same is the case with digitized track (equidistant line segment) and constant length is one meter. Transform the m measurement points to first m reference points with the helmert transformation. Estimate the transformation parameter and standard deviation. For example in above one measurement point is selecting six map points so for each point there will be 6 transformations and transformation parameters will be calculated separately. In every step measurement points are transformed to the reference points shifted by one line element. Store again transformation parameters and standard deviations. The shift of the
corresponding points is repeated until the last measuring point of the last point Comparison road matches.

**Similarity Transformation (4-parameter)**

4 unknowns are calculated and which are the transformation parameters and

Estimation is calculated with this formula

\[ \hat{x} = (A^T PA)^{-1} A^T P l \]

A=Design Matrix, P=weight Matrix, l =reduce Observation 4 unknowns are calculated.

After calculating estimated parameters, which are the four parameters of transformation. Now residuals can be calculated separately. And adjusted trajectories are allocated with residuals. For the best allocation of GPS points, Choose the route with the minimum standard deviation and residuals should be minimum.

To check the accuracy of the transformation following formula is used. Firstly standard deviation is calculated. The invariant is with respect to the rotation

\[ \sigma_{pt} = \sqrt{\sigma_x^2 + \sigma_y^2} = \sqrt{2 \cdot \sigma_0^2} \]

with \( \sigma_0^2 = \frac{\sum_{i=1}^{n}(v_x^2 + v_y^2)}{2 \cdot n - 4} \)

\( \sigma_{pt} \) is the standard deviation of point. An estimate for the variance factor can be estimated

Where \( v_x \) and \( v_y \) are the inconsistencies which are added to the original system

**Affine transformation (6 parameters)**

These are the equations

\[ X = m_x (\cos \alpha) x - m_y (\sin \beta) y + c \]

\[ Y = m_x (\sin \alpha) x - m_y (\cos \beta) y + f \]

So the transformations parameters, we can discuss as follow

- Two translations \( c \) and \( f \)
- Two rotations \( \alpha \) and \( \beta \)
- Two scale factors \( m_y, m_x \)

After this finally we get

\[ X + v_x = a \cdot x + b \cdot y + c \]

\[ Y + v_y = d \cdot y + e \cdot x + f \]

In these equations 6 parameters are given \( a, b, c, d, e, f \)
As a normal equation N for each shift of the corresponding points and all alternative roots can be found only once and only h vector is recalculated. That’s why affine transformation computational effort is limited.

And now calculation of standard deviations

$$\sigma_{p_{kt}} = \sqrt{\sigma_x^2 + \sigma_y^2} = \sqrt{2 \cdot \sigma_0^2}$$

with $$\sigma_0^2 = \frac{\sum_{i=1}^{n} (v_x^2 + v_y^2)}{2 \cdot n - 6}$$

This size can be used as a criterion for choosing the best estimate of the parameters. The resulting path is the matched path with the best estimates. It means that choosing path contains the minimum standard deviation.

**Results**

These are the results for nearest neighbor map matching algorithm and also extended velocity information with exemplary and original data set. Original data is prepared and lot of noisy data is removed.
matched point: 3512035.24379 : 5403971.67112
-> time : 5.0
velocity for each matched point 7 0.2497.1045755 : 10.80784.3422
-> matched point: 3512057.08493 ; 5405982.79198
-> time : 10.0
velocity for each matched point 4.538226845581 ; 2.22417072225
-> matched point: 3512077.67282 ; 5406009.84709
-> time : 15.0
velocity for each matched point 3.947578886885 ; 4.1510205851
-> matched point: 3512097.74504 ; 5404000.52473
-> time : 20.0
velocity for each matched point 4.01480325966 ; -0.604516656693
-> matched point: 3512120.09408 ; 5404013.25597
-> time : 25.0
velocity for each matched point 4.46974757034 ; 2.546247461350
-> matched point: 3512141.10003 ; 5404020.84237
-> time : 30.0
velocity for each matched point 4.2171830904 ; 1.41727039751
-> matched point: 3512161.25256 ; 5404046.72537
-> time : 35.0
velocity for each matched point 5.41265772666 ; 4.987601412347
-> matched point: 3512142.23922 ; 5404056.28505
-> time : 40.0
velocity for each matched point -8.20074955096 ; 2.11223546166
-> matched point: 3512126.82444 ; 5404088.14487
-> time : 45.0
velocity for each matched point 4.82095606686 ; 8.378540784535
-> matched point: 3512111.62149 ; 5403127.48206
-> time : 50.0
velocity for each matched point -3.20058336154 ; 6.9296137523
Look at the velocity between the pairs of GPS points and matched points.

Choose a threshold of velocity from feature class. (30km is chosen)

If the difference of velocity is exceeded from the threshold velocity then the GPS point should matched to the second minimum distance street.

It is only the quality check. As velocity is calculated manually so this function can check and applied further more. Similarly for the original data,
Similarity and affine adjusted trajectories with data set 1 and data set 2.

Adjusted Trajectory Using Similarity Transformation (Dataset 1)

Adjusted Trajectory Using Affine Transformation (Dataset 1)

Adjusted Trajectory Using Similarity Transformation (Dataset 2)
Comparison of Simi and affine adjusted trajectories of Original data, dataset 1, data set 2
First it is calculated the norm of the residual vector and results were not good and in exemplary data set it is about 1 to 4m but in original data it was about 1.8km error. So this method was further improved in python script and finally calculated the standard deviations of the residual vector as given below.

<table>
<thead>
<tr>
<th>Transformations</th>
<th>GPS data set Standard deviation (m)</th>
<th>Dataset 1 Standard deviation (m)</th>
<th>Dataset 2 Standard deviation (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affine</td>
<td>14.06</td>
<td>0.0237</td>
<td>0.179</td>
</tr>
<tr>
<td>similarity</td>
<td>4.54</td>
<td>0.0664</td>
<td>1.91</td>
</tr>
</tbody>
</table>

Table 1. Standard deviations of the residual vector
Conclusion and suggestions

• Every algorithm has its certain limits.
• Nearest distance both algorithms are restricted to the minimum distance of GPS point to the street.
• Similarity and Affine has more chances to be accurate as it has 4 and 6 degrees of freedom.
• In exemplary data set Affine is giving better results as compared to the similarity.
• In a practical data lot of complications are involved. That it contains a huge amount of points and noisy data is also incorporated that, s why error chances are high but these errors can be compensated to some preprocessing of the data. Another thing is the efficiency of the python program. Sometimes it happens program is working well but there are some errors in the data. So avoid such kind of things.
• It depends on the user requirement which algorithm is taken into account.
• Map matching results are not always perfect and mismatches sometimes to the wrong road connection so it is necessary to watch out under what circumstances map matching algorithms is performed and also incorporated accuracies of sensor and surrounding road segments. There are three main components of map matching, position data, digital map and MM algorithm, Sometimes you have to work with noisy data so it requires input data processing and it gives then precise position. Similarly digital map errors and algorithms short coming create varying levels of matching accuracy.
• Efforts for improving map matching accuracy can be done by increasing the accuracy of estimates and by matching these estimates with the high accuracy map.