A Global-Voting Map Matching Algorithm on the Base of Taxi GPS Data

Xu-Hua Yang, Xiang-Fei Wang
College of Computer Science and Technology
Zhejiang University of Technology
Hangzhou, 310023, China
P.R.China
xhyang@zjut.edu.cn

Abstract: Since the existing floating car map-matching algorithms lead to high error rate when GPS data sampling rate is low, we propose a global-voting map matching algorithm. Based on floating car GPS track data, the algorithm do not only consider the influence to matching process caused by the topological information of road network but also the different spatial distance of GPS track data. In this matching algorithm, we device a new indicator to model the influence of geometric and topological information of road network and define a static matching matrix (SMM) as intermediate results. Based on the SMM, we define a distance weighted function to revise the SMM and build a dynamic matching matrix (DMM), and the function reflects the strength of the influence weighted by the distance between GPS points. After that, referring the DMM, we design an efficient voting algorithm to identify the optimal trajectory as map matching results. In this paper, we apply the algorithm to real Hangzhou taxi data. Results show that this map matching algorithm can make full use of existing information and perform well when GPS data sampling rate is low.

Key-Words: floating car, taxi GPS data, low-sampling-rate, global-voting, map matching, topological information, road network

1 Introduction

Floating car refers to the buses or taxis which installed GPS positioning device and driving at the city’s main street [1-4]. Floating car technology is a kind of dynamic traffic information detection technology which spring up in recent years, and it’s one of the advanced technological ways which been used in intelligent transportation system for road traffic information [5-6]. Floating car can regularly transmit the data including the vehicle number, time, direction, latitude and longitude to the control center. After information processing, we can easily get the real-time traffic information of the whole city road network [7-10]. Map matching algorithm is one of the key technologies of floating car data processing, and it can maximum rectify the GPS satellite positioning error and the stray from trajectory path [11]. In this paper, we proposed a global-voting map matching algorithm on the base of taxi GPS data.

In order to ensure the matching accuracy, the traditional map matching algorithm described in [12-15], the GPS sampling rate is often very high. But in fact, for the reason of energy conservation and the characteristics of floating car itself, the sampling rate that transmit to the dispatch center tend to be low [16]. In the case of low sampling rate (such as 2 minutes), even the taxi speed is only 30 km/h, the distance between two sampling points can reach 1000 meters, so most information between the two GPS points will be lost. Therefore, the traditional
map matching algorithm cannot be used in this condition.

The map matching algorithms which consider the geometry feature described in [17-21], according to the matching process, these algorithms can be divided into point-to-point matching, point-to-curve matching, and curve-to-curve matching. For lacking of considering of the whole road network topological information, this kind of matching algorithm can easily lead to low accuracy in the complicated road environment and low sampling rate of floating car date. At the same time, these matching algorithms do not consider the global interaction between GPS points. In the process of one GPS point matching, these kind of algorithms use only the information contains in the point itself or just consider its neighbor nodes [12-25], but for the low sampling rate floating car data which lost most of connections information between nodes, it's hard to get the right matching results [26-27].

Floating car data not only reflects the location information of the vehicle, to some extent, it also reflects the road topological information and the time sequence information of the GPS track points. So the map matching algorithm for low sampling rate floating car data proposed in this paper consider the topology structure of road network and the global correlation information between GPS points, and the degree of influence between them is weighted according to the distance between the GPS point, so as to improve the accuracy of matching.

2 Global-Voting Map Matching Algorithm

2.1 Basic strategy of the algorithm
In the case of low sampling rate, the traditional map matching algorithm cannot obtain a correct result for the insufficient information input. So we need a more adaptable and robust matching algorithm to deal with this condition. In this paper, we put forward a global-voting map matching algorithm. The algorithm considers two basic strategies:
(1) Considering the topological structure of the road.

1) As shown in Figure 1 (on), although the P2 GPS point is closer to the road segment BE, but if we consider the topological structure of road network, the vehicle may not take a detour to the vertical road and then back to the horizontal road.

2) As shown in Figure 1 (below), although the P3 GPS point is closer to road B, but if we consider the topological structure of road network, the P3 must be matched to road A for there is no way connect road A and B.

Fig.1 Consider the topological structure of road network

(2) Considering the effect of the different distance GPS trajectory data on the matching process.

As shown in Figure 2, in the matching process of P3, in order to elect the optimal matching result from the candidate point set, the neighbor nodes: P1, P2, P4, P5 should be involved in the matching process of P3. Conversely, in the matching process of P1, P2, P4 or P5, the GPS point P3 will also affect their matching results. What's more, in the map matching process of P3, the impact from P1, P2, P4, and P5 can be different for the distance from them to P3 is different. Obviously, the impact of P2, P4 on the matching process of P3 is greater than the GPS point P1, P5.

Fig.2 Mutual influence of different distance adjacent GPS track data
As shown in Figure 3, for example, if we assume that GPS point P1, P2 and P4 have correctly matched to the road segment AB and BC respectively, besides, the distance from P3 to road segment BC and BD is almost equal, the existing map matching algorithms are quite easy to obtain an error matching result, this problem is known as the Y-junction problem [28]. However, if we consider the interaction influence between GPS points, it will be very easy to get the correct result.

![Fig.3 Y-junction problem](image)

**2.2 The procedures of the algorithm**

Firstly, the global voting algorithm calculates the candidate set of the current track points which represent the possible matching result. Then, considering the short-range probability and the connective probability to form the static matching matrix. After that, considering the different distance between every two GPS points to form the dynamic matching matrix. On this basis, come the global GPS points voting procedure to complete matching algorithm. Specifically, the algorithm can be divided into five steps:

A) The data pre-processing procedure. In this procedure, we finish the road network construction and the GPS track data arrangement.

B) The calculation of the candidate points set. According to the road segment projection process, we calculate the candidate set of each GPS point.

C) Static analysis procedure. We define the matching function to obtain the static matching matrix of each GPS point.

D) Weighted analysis procedure. We define the distance weighted function to reflect the relationship between the distance of points and the influence between them. After the global computation, we will obtain the dynamic matching matrix.

E) The global-voting procedure.

![Fig.4 Flow chart of the global voting algorithm](image)

**2.2.1 Data pre-processing**

As shown in Figure 5 (on), a GPS point record includes longitude, latitude and time. We define the set of all GPS points as GPS records. As shown in Figure 5 (below), the GPS track is defined as a group of GPS point that sampling interval is $\Delta t$. This paper studies the situation that $\Delta t \geq 2\text{ min}$.
Fig. 5 GPS track data

A road segment is defined as a part of a road which is separated by two adjacent intersections. Each road segment contains the starting point, ending point and the road length.

The road network is defined as a directed graph $G(V, E)$, where $V$ is the intersection or the road starting point or ending point. $E$ is the road segment which is separated by two adjacent intersections. What's more, according to the definition of the road network, we can define a path: selecting two points $V_i, V_j$, to find a set of interconnected sections $s_1 \rightarrow s_2 \rightarrow s_n$ , where $s_{start}=V_i, s_{end}=V_j$.

2.2.2 Calculate the candidate points set

For every taxi GPS data $p_i$ included in a path $T$: $p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow \cdots \rightarrow p_n$, selecting all the sections within the range of a radius $r$ as a candidate road segment $s_i^k$, where $k$ represents that the segment is $k$-th segment. We get the corresponding candidate point though a projection process: if the GPS point $p_i$ have a vertical point within the range of road segment $s_i^k$, we will choose this vertical point as the candidate point marked as $c_i^k$. Or we will choose the starting or ending point of the road segment which is nearer to $p_i$ as the candidate point. Thus, we can get the candidate points set of every GPS point. As shown in Figure 6, we will choose $c_i^1, c_i^2, c_i^3$ as the candidate point set of $p_i$.

Fig. 6 Candidate points selection process

2.2.3 Static analysis procedure

According to [29], the measurement error of a GPS point obeys the Normal distribution $N(\mu, \delta^2)$. Therefore, according to the candidate point sets calculated from the second procedure, we define the short-range probability $MP$ which represent the possibility that the candidate point $c_i^j$ is the correct matching result:

$$MP(c_i^j) = \frac{1}{\sqrt{2\pi} \delta} e^{-\left(\frac{x_i^j - \mu}{2\delta}\right)^2}$$ (1)

Where $x_i^j$ is the Euclidean distance between the GPS point $p_i$ and the candidate point $c_i^j$. Obviously, the short-range probability can represent the probability of that the candidate point be the correct matching results without considering the neighbor nodes but it is easy to lead to errors.

To avoid this kind of errors, we define the connectivity probability $CP$ based on the topology structure of road network and the shortest path length between every two candidate nodes:

$$CP(c_{i-1} \rightarrow c_i^k) = \frac{l_{i-1 \rightarrow i}}{s_{(i-1,j) \rightarrow (i,k)}}$$ (2)

where $l_{i-1 \rightarrow i}$ is the Euclidean distance between the two GPS point $p_{i-1}$ and $p_i$, $s_{(i-1,j) \rightarrow (i,k)}$ is the shortest path length between $c_{i-1}^j$ and $c_i^k$ which is the two candidate points of $p_{i-1}$ and $p_i$ respectively. In particular, in this paper, we choose the Dijkstra
algorithm as the shortest path algorithm [21] [30].

In summary, the short-range probability considered the geometry properties of road network, and the connectivity probability take the topology structure of road network into account. On this basis, we define a matching function $F$:

$$F(c_{i-1}^j \rightarrow c_i^k) = MP(c_i^k) + CP(c_{i-1}^j \rightarrow c_i^k)$$  \hspace{1cm} (3)

Selecting the two adjacent GPS points $p_{i-1}, p_i$, and their candidate points $c_{i-1}^j$ and $c_i^k$ respectively. After the calculation of matching function, we obtain the static matching matrix $M$ as an intermediate result which does not consider the interaction between GPS points. The following selects the GPS track data $p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow p_4$ as an example. The set candidate points are as shown in Figure 7:

![Candidate Points Set](image)

**Fig.7** The candidate points set

For each path $c_{i-1}^j \rightarrow c_i^k (i \geq 2)$, we can get the static matching matrix:

$$M = \begin{bmatrix}
M_1 & 0 & \ldots & 0 \\
0 & M_2 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & M_n \\
\end{bmatrix}$$  \hspace{2cm} (4)

where $M_i$ is the matrix calculated by the matching function of the path $c_{i-1}^j \rightarrow c_i^k (i \geq 2)$:

$$M_i = \begin{bmatrix}
F(c_{i-1}^j \rightarrow c_i^k) & F(c_{i-1}^j \rightarrow c_i^k) & \cdots & F(c_{i-1}^j \rightarrow c_i^k) \\
F(c_{i-1}^j \rightarrow c_i^k) & F(c_{i-1}^j \rightarrow c_i^k) & \cdots & F(c_{i-1}^j \rightarrow c_i^k) \\
\vdots & \vdots & \ddots & \vdots \\
F(c_{i-1}^j \rightarrow c_i^k) & F(c_{i-1}^j \rightarrow c_i^k) & \cdots & F(c_{i-1}^j \rightarrow c_i^k) \\
\end{bmatrix}$$  \hspace{2cm} (5)

Where $1 \leq j \leq m, 1 \leq k \leq n, m, n$ is the total number of the candidate points of $c_{i-1}, c_i$. To facilitate the instructions, we assume:

$$M_2 = \begin{bmatrix}
F(c_{i-1}^j \rightarrow c_i^k) & F(c_{i-1}^j \rightarrow c_i^k) & \cdots & F(c_{i-1}^j \rightarrow c_i^k) \\
F(c_{i-1}^j \rightarrow c_i^k) & F(c_{i-1}^j \rightarrow c_i^k) & \cdots & F(c_{i-1}^j \rightarrow c_i^k) \\
\vdots & \vdots & \ddots & \vdots \\
F(c_{i-1}^j \rightarrow c_i^k) & F(c_{i-1}^j \rightarrow c_i^k) & \cdots & F(c_{i-1}^j \rightarrow c_i^k) \\
\end{bmatrix} = \begin{bmatrix}
0.8 & 0.7 \\
0.6 & 0.5 \\
0.4 & 0.5 \\
0.3 & 0.4 \\
0.3 & 0.4 \\
0.9 & 0.7 \\
0.6 & 0.5 \\
\end{bmatrix}$$

And obtain the static matching matrix:

$$M = \begin{bmatrix}
0.8 & 0.7 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.5 & 0 & 0 & 0 & 0 & 0 \\
0.4 & 0.5 & 0 & 0 & 0 & 0 & 0 \\
0.0 & 0.3 & 0.6 & 0 & 0 & 0 & 0 \\
0.0 & 0.2 & 0.4 & 0 & 0 & 0 & 0 \\
0.0 & 0.0 & 0.3 & 0.5 & 0.4 & 0 & 0 \\
0.0 & 0.0 & 0.0 & 0.9 & 0.7 & 0.6 & 0 \\
\end{bmatrix}$$

2.2.4 Weighted analysis procedure

Based on the static matching matrix obtained in step 3, we consider that the farther of the distance between GPS points, the weaker of the influence between them. Therefore, for each GPS point, we define an $n-1$ dimensional distance influence matrix $W_i$:

$$W_i = \begin{bmatrix}
W_i^1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & W_i^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & W_i^{i-1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & W_i^{i} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & W_i^{n} \\
\end{bmatrix}$$  \hspace{2cm} (6)

Where $W_i^j = f(dist(p_i, p_j))$ defined as the distance weighted function which represent the phenomenon of that the distance between nodes farther away, the weaker the effect between them. So, the distance weighted function should meet the following conditions:

1) $f(0) = 1$
2) $f(\infty) = 0$
3) $0 < f(x_1) < f(x_2) < 1$, when $0 < x_1 < x_2$
As for the distance weighted function, the following algorithm evaluation part in this paper, we will carry on the simulation analysis to the different weighting functions. On the basis of experiment, we think the interaction between the GPS points is proportional to the distance, but once the distance exceeds a threshold, the interaction between them decreases quickly. So we choose \( f(x) = \exp(-x^2/\beta^2) \) as the distance weighted function, where \( \beta \) is the threshold, and we will carry on the simulation analysis to different threshold in the experiment. Here, to facilitating the instructions, we select \( f = 2^{-\beta^2} \) to illustrate the algorithm process.

We define the dynamic matching matrix to amends

### The static matching matrix to reflect the influence of the distance between GPS points \( G_i \) \((1 \leq i \leq n)\):

\[
G_i = \begin{bmatrix}
G_{i1} & 0 & 0 & 0 & 0 & 0 \\
0 & G_{i2} & 0 & 0 & 0 & 0 \\
0 & 0 & G_{i3} & 0 & 0 & 0 \\
0 & 0 & 0 & G_{i4} & 0 & 0 \\
0 & 0 & 0 & 0 & G_{i5} & 0 \\
0 & 0 & 0 & 0 & 0 & G_{i6}
\end{bmatrix}
\]

where \( G_{i1} = M_2W_{i1}^1, G_{i2} = M_3W_{i2}^2, G_{i3} = M_4W_{i3}^3, \)

\[
G_{i4} = M_5W_{i4}^4, G_{i5} = M_6W_{i5}^5 .
\]

Specially, the dynamic matching matrix of the GPS track data \( p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow p_4 \) are as follows:

<table>
<thead>
<tr>
<th></th>
<th>0.4</th>
<th>0.35</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.075</td>
<td>0.15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.05</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0375</td>
<td>0.0625</td>
<td>0.05</td>
<td>0.075</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1125</td>
<td>0.0875</td>
<td>0.075</td>
<td>0.075</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0.4</th>
<th>0.35</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.15</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.125</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.075</td>
<td>0.125</td>
<td>0.1</td>
<td>0.15</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.225</td>
<td>0.175</td>
<td>0.15</td>
<td>0.15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0.2</th>
<th>0.175</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.15</td>
<td>0.125</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.125</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.15</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.125</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.15</td>
<td>0.25</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0.35</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0.1</th>
<th>0.0875</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.075</td>
<td>0.0625</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.05</td>
<td>0.0625</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.075</td>
<td>0.15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.05</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.15</td>
<td>0.25</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0.35</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>
2.2.5 The global-voting procedure

For example, the vote processes for the candidate point \( c_1 \) are as follow: first, select \( G_2 \) as the dynamic matching matrix, and then look for the maximum value in this matrix. We can get a candidate points sequence which contains the point \( c_1 \): \( c_1 \rightarrow c_2 \rightarrow c_3 \rightarrow c_4 \) and the votes of each candidate point in this sequence plus one. We call this as single point voting procedure. Generally, the single point voting procedure for candidate point \( c_i \) is as follows: first, the single point voting: select the max value in the column \( j \) of \( G_i \) in the dynamic matching matrix. Then search up and down in the matrix to find the maximum sequence that contains \( c_i \). For all the candidate points, by repeating the operation above, we can get the vote results for each point. Finally, for each GPS point \( p_i \), the algorithm selects the largest of the votes number in the candidate points set as the global matching results. The votes result for \( p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow p_4 \) is shown in Table 1, so the global matching result is \( c_1 \rightarrow c_2 \rightarrow c_3 \rightarrow c_4 \).

<table>
<thead>
<tr>
<th>Candidate point ( c_1 )</th>
<th>( c_2 )</th>
<th>( c_3 )</th>
<th>( c_4 )</th>
<th>( c_1 )</th>
<th>( c_2 )</th>
<th>( c_3 )</th>
<th>( c_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Votes</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
</tbody>
</table>

3 The result of the experiment and analysis

3.1 The sources of experiment data

In the experiment, as shown in Figure 8 (on), this paper is based on the city’s road network, which is acquired from the GaoDe map API and open source website OpenStreetMap [31]. In addition, as shown in Figure 8 (below), the trajectory data is the real taxi GPS data of Hangzhou. The data format is shown in Table 2.

Table 2 Hangzhou taxi data

<table>
<thead>
<tr>
<th>MESSAGE_ID</th>
<th>VEHICLE_ID</th>
<th>VEHICLE_NUM</th>
<th>LONGI</th>
<th>LATI</th>
<th>SPEED(Km/h)</th>
<th>DIRECTION</th>
<th>STATE</th>
<th>SPEED_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3879981013</td>
<td>21621</td>
<td>120.112833</td>
<td>30.317983</td>
<td>0.00</td>
<td>180.00</td>
<td>0</td>
<td>2012-1-27 7:38:52</td>
</tr>
<tr>
<td>2</td>
<td>3879981004</td>
<td>26325</td>
<td>114.490036</td>
<td>33.853073</td>
<td>0.37</td>
<td>350.00</td>
<td>0</td>
<td>2012-1-27 7:38:53</td>
</tr>
<tr>
<td>3</td>
<td>3879981202</td>
<td>25368</td>
<td>120.161750</td>
<td>30.246517</td>
<td>27.00</td>
<td>135.00</td>
<td>0</td>
<td>2012-1-27 7:38:50</td>
</tr>
<tr>
<td>4</td>
<td>3879981301</td>
<td>23390</td>
<td>120.165283</td>
<td>30.222650</td>
<td>0.00</td>
<td>180.00</td>
<td>0</td>
<td>2012-1-27 7:38:52</td>
</tr>
<tr>
<td>5</td>
<td>3879981300</td>
<td>23017</td>
<td>120.064117</td>
<td>30.248517</td>
<td>0.00</td>
<td>135.00</td>
<td>0</td>
<td>2012-1-27 7:38:53</td>
</tr>
<tr>
<td>6</td>
<td>3879981400</td>
<td>23439</td>
<td>120.114983</td>
<td>30.312417</td>
<td>0.00</td>
<td>135.00</td>
<td>0</td>
<td>2012-1-27 7:38:53</td>
</tr>
<tr>
<td>7</td>
<td>3879981400</td>
<td>16351</td>
<td>120.212006</td>
<td>30.318755</td>
<td>0.00</td>
<td>250.00</td>
<td>0</td>
<td>2012-1-27 7:38:54</td>
</tr>
</tbody>
</table>
3.2 Experimental parameters
To evaluate the efficiency of matching algorithm, we choose point-to-point map matching algorithm [10] and multi-criteria dynamic programming map matching algorithm [14] as comparative algorithm. This point-to-point map matching algorithm (P2PMM) also find candidate points and the road segments first, then calculate the distance between the GPS point and the candidate point, and choose the shortest distance candidate point as the matching result. And this multi-criteria dynamic programming map matching algorithm (MDPMM) considers the topological information of road network and shortest path between GPS points, but it take only the neighbor points into account instead of the global interact of all the GPS points. We also define the correct matching proportion (CMP) which is the proportion of the number of correctly matched GPS points and the total number of the GPS points.

3.3 The comparison process
In step 4, we define the distance weighted function \( W_i = f(dist(p_i, p_j)) \) to represent the phenomenon of that the distance between nodes farther away, the weaker the effect between them. Therefore, the selection of distance weighted function is very important. In step 3, in order to facilitate, we select the function \( f = 2^{-|i-j|} \). But in the actual experiments, we select two different distance weighted function: a. linear function \( f = -x/\beta + 1 \); b. exponential function \( f = \exp(-x^2/\beta^2) \), where \( \beta \) is the threshold and we compared the different matching accuracy of different threshold. What's more, for the global-voting map matching algorithm is designed for low sampling rate data, we compared different matching algorithms under different sampling rate of data input.

3.4 Results
As shown in Figure 9, under the condition of different sampling intervals, the global voting map matching algorithm which selects exponential function as the distance weighted function performs better and the correct matching proportion is higher. On the other hand, as shown in Figure 10, no matter we select the exponential function with any threshold(\( \beta=12, 10, 7 \) or 5) as the distance weighted function perform better than if we select the linear function under different sampling rate.

4 Conclusion
Based on analyzing the existing map matching algorithm, the existing floating car map-matching algorithms lead to high error rate when GPS data sampling rate is low. In fact, floating car data not only reflects the location information of the vehicle, to some extent, it also reflects the road topological information and the time sequence information of the GPS track points, so we proposed a global voting map matching algorithm which takes the topological structure of road network and the interaction of the GPS points into account. The algorithm defines a
distance weighted function to reflect the relationship between the distance and the interaction of points. At last, we used the algorithm to match the real Hangzhou taxi date and the results show that this map matching algorithm can make full use of existing information and perform well when GPS data sampling rate is low.

5 Acknowledgments

The work is supported by the National Natural Science Foundation of China under Grant No. 61374152, the commonweal application technique research project of Zhejiang Province under Grant No. 2012C23126.

References:
[15] Xu Hao, Liu Hong-Chao, Tan Chin-Woo, Bao Yuan-Lu, Development and Application of an Enhanced Kalman Filter and Global Positioning...


