7-2012

Calibrating Large Scale Vehicle Trajectory Data

Siyuan Liu

Ce Liu

Qiong Luo

Lionel Ni

Ramayya Krishnan

Citation

Liu, Siyuan; Liu, Ce; Luo, Qiong; Ni, Lionel; and Krishnan, Ramayya. Calibrating Large Scale Vehicle Trajectory Data. (2012). *MDM '12 Proceedings of the 2012 IEEE 13th International Conference on Mobile Data Management (mdm 2012)*. 222-231.

Available at: https://ink.library.smu.edu.sg/larc/5

LARC
LIVING ANALYTICS
RESEARCH CENTRE

Calibrating Large Scale Vehicle Trajectory Data

Siyuan Liu, Carnegie Mellon University, Pittsburgh, PA
syliu@andrew.cmu.edu
Ce Liu, University of Pittsburgh, Pittsburgh, PA
Cel38@pitt.edu
Qiong Luo, HKUST
luo@cse.ust.hk
Lionel Ni, HKUST
ni@cse.ust.hk
Ramayya Krishnan, Carnegie Mellon University, Pittsburgh, PA
rk2x@cmu.edu

February, 2012

LARC-TR-01-12

LARC Technical Report Series: http://smu.edu.sg/centres/larc/larc-technical-reports-series/

SMU
SINGAPORE MANAGEMENT
UNIVERSITY

Carnegie
Mellon
University

ABSTRACT

An accurate and sufficient vehicle trajectory dataset is the basis to many trajectory-based data mining tasks and applications. However, vehicle trajectories sampled by GPS devices are usually at a relatively low sampling rate and contain notable location errors. To address these two problems in GPS trajectory data, we propose WI-matching, the first vehicle trajectory calibration framework to take advantage of road networks topology and geometry information and trajectory historical information in large scale. WI-matching consists of a Weighting based map matching algorithm and a trajectory interpolation based *matching* algorithm. In our WI-matching framework, we first integrate the vehicle GPS data with digital road networks data, to identify the roads where a vehicle traveled and the vehicle locations along the roads. Then our weighting-based map matching algorithm considers (1) the geometric and topological information of the road networks and (2) the spatiotemporal trajectory information to efficiently and effectively calibrate the GPS data points. Finally, our interpolation algorithm identifies paths between consecutive GPS points, and adds points with estimated vehicle status (location and time stamp) along the paths to construct sufficient vehicle trajectories. We have evaluated our algorithms on a large-scale real life data set in comparison with the state of the art. Our extensive and empirical results indicate that our WI-matching achieves a high accuracy as well as a high efficiency on real-world data which beats the state of the art.

Index Terms—Calibration, vehicle trajectory, map matching

Calibrating Large Scale Vehicle Trajectory Data

Siyuan Liu [*], Ce Liu [$], Qiong Luo [#], Lionel Ni [#], Ramayya Krishnan [*]

[*] *iLab, Heinz College, Carnegie Mellon University*
[$] *School of Information Sciences, University of Pittsburgh*
[#] *Department of Computer Science and Engineering, Hong Kong University of Science and Technology*

***Abstract*—An accurate and sufficient vehicle trajectory data set is the basis to many trajectory-based data mining tasks and applications. However, vehicle trajectories sampled by GPS devices are usually at a relatively low sampling rate and contain notable location errors. To address these two problems in GPS trajectory data, we propose WI-matching, the first vehicle trajectory calibration framework to take advantage of road networks topology and geometry information and trajectory historical information in large scale. WI-matching consists of a Weighting-based map matching algorithm and a trajectory Interpolation-based $matching$ algorithm. In our WI-matching framework, we first integrate the vehicle GPS data with digital road networks data, to identify the roads where a vehicle traveled and the vehicle locations along the roads. Then our weighting-based map matching algorithm considers (1) the geometric and topological information of the road networks and (2) the spatiotemporal trajectory information to efficiently and effectively calibrate the GPS data points. Finally, our interpolation algorithm identifies paths between consecutive GPS points, and adds points with estimated vehicle status (location and time stamp) along the paths to construct sufficient vehicle trajectories. We have evaluated our algorithms on a large-scale real life data set in comparison with the state of the art. Our extensive and empirical results indicate that our WI-matching achieves a high accuracy as well as a high efficiency on real-world data which beats the state of the art.**

***Index Terms*—Calibration, vehicle trajectory, map matching**

I. Introduction

Traffic management applications such as route suggestion, traffic monitoring and traffic flow analysis require an accurate and sufficient vehicle trajectory data set [1], [2]. Due to privacy and cost concerns, trajectory data sets are hard to obtain. So far, GPS (Global Position System) devices, are the main source for trajectory data sets.

Typically a vehicle GPS trajectory comprises a sequence of sampled data points with location and time stamp information [3]. However, the GPS system on vehicle (e.g., taxi) is usually installed for civic applications. Due to the low cost of these applications, such raw data usually contains insufficient sample points due to the low sampling frequency as well as lossy communication channel (especially in urban areas, the high buildings seriously block GPS signal) . Moreover, the locations reported in the GPS points are often off the actual locations with an error range up to hundreds of meters [1]. As shown on the left of Figure 1, there are only nine GPS points from a vehicle in an area of four square kilometers, and the GPS points are around 150 meters off the road. In many application scenarios, we desire an accurate and sufficient trajectory calibrated from the nine GPS points as shown on the right of Figure 1.

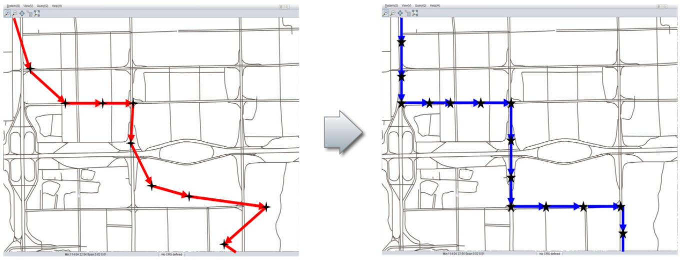


Fig. 1. Vehicle GPS trajectory

To address the sufficiency and accuracy challenge of GPS data, we propose WI-matching, a novel and practical GPS trajectory data calibration framework. Our main idea is to utilize the road network map information together with the spatiotemporal information in the GPS raw data points to correct errors and to add more points through interpolation. As illustrated in Figure 2, WI-matching consists of three components - (1) GPS and map data preprocessing; (2) map matching; and (3) interpolation. The pre-processing component divides a digital map into cells and records the link (road) information within each cell. The map matching algorithm matches GPS points to the links. Finally the interpolation algorithm estimates the status of the vehicle, i.e., timestamps and positions, along the links where GPS reports are scarce.

In the map matching component, we propose a novel weighting-based map matching algorithm that takes into account (1) the geometric and topological information of the road networks and (2) the spatiotemporal trajectory information. Specifically, our weighting scheme considers the proximity between the GPS sample points and the roads, the differences between the vehicle heading and road heading, and information about road connectivity and turns.

We have empirically evaluated our WI-matching framework using a real-world GPS trajectory data set. The data set contains one year's GPS trajectories of over 5000 taxis in a major city in China. Our results show that our framework achieves a high efficiency as well as high accuracy in comparison with two other state-of-the-art methods.

In a summary, the contributions of our work are as follows. First, we are the first to utilize both geometric/ topological information of the road networks and historical trajectories to

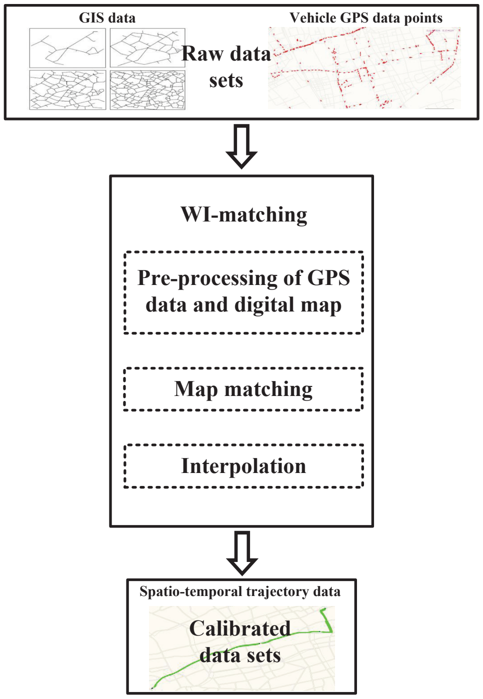


Fig. 2. Vehicle trajectory calibration

devise a novel and practical trajectory calibration framework in a large scale data scenario. Second, WI-matching can handle large scale trajectory data and return us accurate and sufficient processed trajectories. Third, we employ real life large scale datasets to evaluate our methods and the state of the art, which confirms the high and promising efficiency, accuracy and scalability of our methods, and sheds lights into the future vehicle trajectory related research.

The rest of this paper is organized as follows. In Section II, we present some preliminary information of our work and then introduce the related work. Our proposed map matching algorithm is discussed in Section III and the interpolation algorithm is presented in Section IV. Section V reports the empirical experiment results and describes the evaluation in terms of accuracy, efficiency and scalability. Last, we conclude the paper and outline the future work.

II. Preliminary and Related Work

In this section, we provide preliminary information of our work and survey the related work. In our work, we study the vehicle trajectory in digital map and road networks. Hence, first, we introduce vehicle GPS trajectory; second, the digital map and road networks are discussed; at last, a survey of related work is provided.

A. Vehicle GPS trajectory

A GPS trajectory data set is collected from vehicles equipped with GPS receivers. Each GPS receiver periodically reports the vehicle's current status information to a data center via GPRS.

Definition 1: A vehicle GPS sample data point P_i is defined as a vector including the vehicle ID, the time stamp when the report was sent out, the vehicle's location (longitude and latitude), the instantaneous speed, and the heading of the vehicle.

For example, a sample data point is [0793, 2007-02-18 00:04:27, 121.444800, 31.260500, 32, 157]. In this example, the vehicle ID is 0793, the reporting time is 2007-02-18 00:04:27, the reported location of the vehicle is [121.444800, 31.260500], the reported vehicle speed is 32 kmph, and the heading of the vehicle is 157 degrees from the north.

Definition 2: A vehicle GPS trajectory is defined as a sequence of vehicle GPS sample data points in the ascending order of timestamps.

B. Digital map and road networks

A digital map records the spatial road data (points, lines and polygons) along with their associated labels and attributes [4]. In a digital map, road networks are generally represented in a planar model where each road is represented by a set of arcs [5], [6]. A road network consists of nodes, shape points, links and segments. Nodes represent dead-ends and intersections. An arc between two nodes is called a link. Each link is assumed to be piece-wise linear such that it can be described by a sequence of finite number of points. The first and last points in the sequence are referred to as nodes and the rest as shape points. Shape points describe a link's curvature and divide the link into a set of straight lines. Each straight line is a segment.

C. Related Work

In our work, we utilize the spatial information to calibrate vehicle GPS trajectories. This aspect is related to map matching. Map matching algorithms utilize positioning data and spatial road networks data to identify the correct links on which a vehicle is traveling and determine the location of the vehicle on that link [7]–[9]. Since reliable digital road maps are readily available and their accuracy is normally higher than that of the positioning sensors, map matching can improve the accuracy of positioning systems [10]–[13].

We discuss map matching algorithms by the following three categories. The first category is geometric map matching, which makes use of the geometric information of the spatial road network data by considering only the shape of the road links [14], [15]. It does not consider the connectivity between links. In comparison, the second category, topological map-matching, makes use of the geometry of the links as well as the connectivity and contiguity of the links [7], [14]. The third category of map matching algorithms utilize other techniques such as Kalman Filter and fuzzy logic [4].

Kim *et al.* proposed an advanced map matching algorithm with efficient use of digital road maps [16]. Fu *et al.* presented a hybrid map matching algorithm based on fuzzy comprehensive judgment [17]. Geometric analysis and topology of the road networks are considered in this algorithm. Yin *et al.* [18] proposed a weight based map matching method, which selects a most likely route for a vehicle. Their weighting is based on arcs, and did not consider turns. Nanni *et al.* [19] took an approach considering the k-optimal alternative paths to reconstruct the trajectories from GPS raw data. Yuan *et al.* [20] proposed an interactive-voting method to map matching low-sampling-rate GPS traces, but they did not consider the turns in road networks or heading directions.

In comparison to the previous work on map matching, we utilize the knowledge from road networks and spatiotemporal trajectory data to calibrate vehicle GPS trajectories [1], [2], considering vehicle headings as well as road connectivity, heading, and turns. As a result, our method achieves a high accuracy as well as high efficiency.

III. Weighting-based map matching

In our WI-matching framework, map matching is done after the pre-processing of digital maps and GPS points. The pre-processing of a digital map is to divide it into rectangular cells of a predefined size and to record the information about the links within each cell. The pre-processing of GPS data is to discard vehicles whose GPS reports are so scarce that the time gap between two consecutive reports is longer than a threshold. In our work, we set the threshold to be 20 minutes. Time gaps in GPS reporting longer than this threshold are usually associated with GPS equipment faults.

After pre-processing, the map matching component will identify the link that each GPS point is on and determine the location of the vehicle on the link. This process is done in three steps for each GPS point: (1) Identifying the candidate region, (2) Identifying candidate links, and (3) Weighing candidate links. In the following, we discuss the three steps in order.

A. Identifying the candidate region

For each GPS point, it is inefficient to search the entire map to identify possible links on which the point resides. Rather, we only need to identify a small region in the map that covers all possible links for the point. The size of the region depends on the error range of the GPS data as well as that of the digital map. In a city setting with dense tall buildings and viaducts, the GPS location errors can be as large as 100 meters [2], [21]. As location errors in digital maps are within a much smaller range than GPS location errors, we use 100 meters, twice the GPS location error range, as the distance threshold to set the candidate region conservatively.

Recall that in pre-processing, we already divide the digital map into cells and record the information about links in each cell. Here to identify the candidate region for a GPS point, we only need to first find the cell that contains the GPS point and then to cover the cells around the center cell within the distance threshold. In the case the cell size is 100 meters, the candidate region will be the nine cells around the GPS point, as shown in Figure 3.

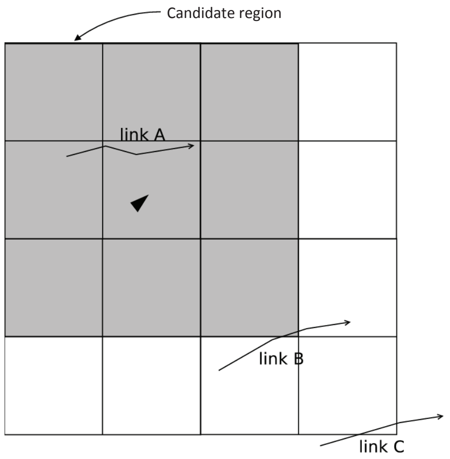


Fig. 3. Candidate region for a GPS point

B. Identifying candidate links

After setting the candidate region of a GPS point, we next identify the candidate links within the region. If there is no link within the region, we regard the vehicle is off known roads and leave the GPS point unmatched with the map; otherwise, we test each link with the region on the following two criteria: (1) whether the distance between the link and the GPS point is less than 100 meters, the distance threshold; (2) whether the difference in the headings of the link and the vehicle is less than 60°, a heading difference threshold used in a previous report [22].

1) Point-link distance: The point-link distance refers to the distance from the GPS point to the link. A link consists of several segments. The point-link distance is obtained by calculating the distance from the GPS point to each segment and then selecting the minimal one [14].

To calculate the distance between the segment and the GPS point, we need to find the closest point on the segment. This can be achieved by projecting the GPS point onto the segment.

Projecting the GPS point onto the segment gives us two cases:

- if the projection point lies on the segment, the projection point is the closest point on this segment.
- if the projection point lies outside the segment, we calculate the distances from the GPS point to the segment's start point and end point, and pick the one with the shorter distance as the closet point.

Figure 4 shows an example of finding the distance between a GPS point and a link. Point p is a GPS point. Link A has three segments: A_0A_1, A_1A_2, A_2A_3. For segment A_0A_1, the projection point lies outside of the segment and the closest point on the segment to p is A_1. So the point-segment distance for A_0A_1 is the distance between p and A_1, denoted as $dist(p,\, A_1)$. For segment A_1A_2, the projection point q lies on the segment, so the point-segment distance for A_1A_2 is the distance between p and q, denoted as $dist(p,\, q)$. For segment A_2A_3, the projection point lies outside the segment and the closest point on the segment is A_2, so the point-segment distance for A_2A_3 is the distance between p and A_2, denoted as $dist(p,\, A_2)$. Since $dist(p,\, q)$ is the least one among the three point-segment distances, it is taken as the distance from

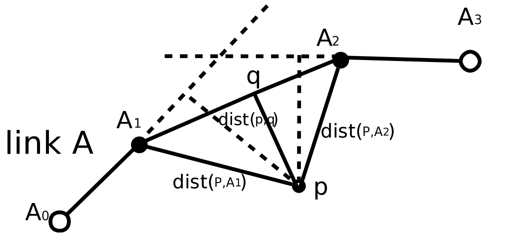


Fig. 4. Calculating distance from point to segment

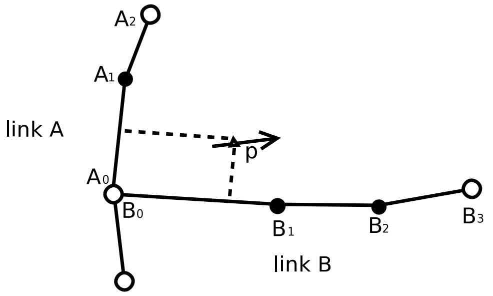


Fig. 5. Link heading

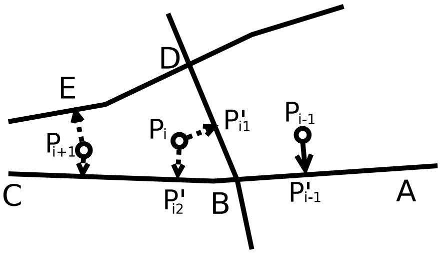


Fig. 6. Determination of actual link for P_i

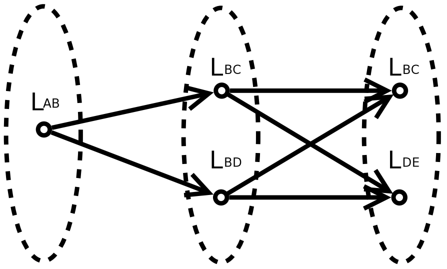


Fig. 7. Candidate routes from P_{i-1} to P_{i+1}

p to link A.

2) Similarity between vehicle and link headings: In addition to point-link distance, we further consider the similarity between vehicle heading and link heading in identifying candidate links.

For a link that consists of multiple segments, the link heading is the heading of the segment whose distance to the GPS point is the shortest. The heading of a segment can be calculated using its start point and end point. The difference between link heading and vehicle heading is denoted as $\Delta\alpha$. If $\Delta\alpha$ is less than the heading diffrence threshold, the link will be identified as a candidate link. As the error of reported vehicle heading is high when the vehicle travels at a low speed, we utilize this heading similarity criterion only when the vehicle speed is higher than a speed threshold. We set the speed threshold to 10.8 kmph as in the previous work [22].

Figure 5 shows an example of link heading. In this figure, Link A's heading is the heading of segment A_0A_1 and link B's heading the heading of segment B_0B_1. If the difference in the headings of link A and the GPS point p is greater than 60°, link A will not be a candidate link.

C. Weighing candidate links

After identifying a set of candidate links for each GPS point, we weigh each link based on the following four factors (1) proximity between the GPS point and the link, (2) similarity between the vehicle heading and link heading, (3) link connectivity, and (4) turns. The first two factors are also used in identifying candidate links. The weighting schemes for these two factors are developed based on geometric analysis whereas those for link connectivity and turns are based on topological analysis.

1) Weighting for proximity: The proximity criterion is based on the distance from the GPS point to the link. The smaller the distance from the GPS point to the link, the higher the possibility that the link is the actual link. So the weight for proximity WS_D is defined as:

$$WS_D = W_D * (1 - \frac{0.5 * (d_i + d_{i+1})}{MAX\_DISTANCE\_ERROR}) \quad (1)$$

where d_i is the distance from the current GPS point P_i to its candidate link, d_{i+1} is the distance from the subsequent GPS point P_{i+1} to its candidate link, W_D is the weight coefficient for proximity, and $MAX\_DISTANCE\_ERROR$ is the maximal value of location error, 100 meters.

2) Weighting for heading: The heading criterion is based on the difference in heading between the vehicle and link. The vehicle heading is highly correlated with the link heading. Figure 8 shows an example of using heading information to weigh candidate links. In Figure 8, points P_1 to P_9 are a sequence of GPS points. The actual vehicle path is marked in the figure. Both BD and BC are candidate links for point P_6. If the heading information is not taken into account, P_6 will be matched onto link BC instead of BD, the actual link traveled.

The weight for heading WS_H is given by:

$$WS_H = W_H * (1 - \frac{0.5 * (angle_i + angle_{i+1})}{MAX\_ANGLE}) \quad (2)$$

where,

- $angle_i$, : the difference between vehicle heading and link heading for the current GPS point P_i
- $angle_{i+1}$, : the difference between vehicle heading and link heading for the next GPS point P_{i+1}
- W_H: the weight coefficient for heading
- MAX_ANGLE: the maximal angle between the vehicle heading and link heading, which is set to 60°

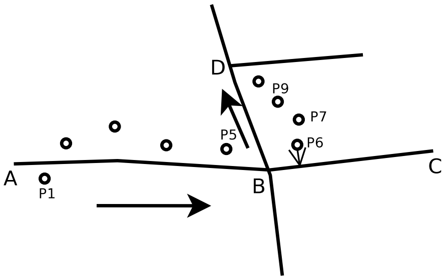


Fig. 8. An example of using heading information

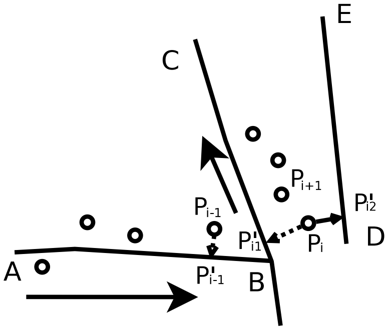


Fig. 9. An example of link connectivity

3) Weighting for link connectivity: Two links are directly connected if and only if one's start point is the end point of the other. Figure 9 shows an example of using link connectivity to weigh candidate links. In this figure, the sequence of GPS points are P_{i-1}, P_i, and P_{i+1}. After P_{i-1} is matched onto link AB, we consider the candidate links of P_i, BC and DE. $P_{i1}^{'}$ and $P_{i2}^{'}$ are the closest points from P_i on the two links respectively. However, since Link DE is not connected to link AB whereas BC is, it is less likely that P_i is on link DE than on BC.

In our weighting for link connectivity, we consider both the preceding and the subsequent GPS points of the current point:

$$WS_{LC} = W_{LC} * \frac{(w_i + w_{i+1})}{2} \tag{3}$$

where,

- W_{LC}: the weight coefficient for lik connectivity
- w_i, w_{i+1}: variables that represent the link connectivity between $P - i$, $P_(i - 1)$ and between P_i, $P_(i + 1)$:
  - w_i = 1 if the current GPS point P_i and preceding GPS point P_{i-1} are matched onto the same link or connected links
  - w_i = 0 if the current GPS point P_i and preceding GPS point P_{i-1} are matched onto unconnected links
  - w_{i+1} = 1 if the current GPS point P_i and subsequent GPS point P_{i+1} are matched onto the same link or connected links
  - w_{i+1} = 0 if the current GPS point P_i and subsequent GPS point P_{i+1} are matched onto unconnected links

4) Weighting for turns: The weighting for turns is based on the observation that paths in practice tend to be direct, rather than full of roundabouts [3].

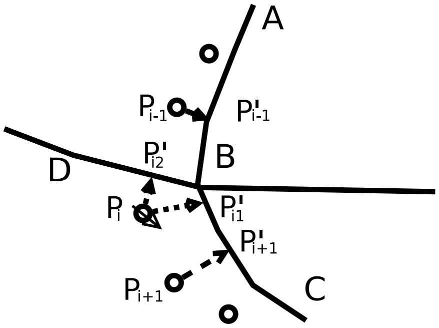


Fig. 10. An example using turn information

Figure 10 shows an example of using information about turns in weighing candidate links. P_{i-1}, P_i, P_{i+1} are three consecutive GPS points. P_{i-1} has been map matched onto link AB. P_i has two candidate links BC and BD, and both links are connected to link AB. $P_{i1}^{'}$ and $P_{i2}^{'}$ are the closest points to P_i on the candidate links correspondingly. Further, we consider the candidate link, BC, for P_{i+1}. There are two possible routes from P_{i-1} to P_{i+1}: (1) $P_{i-1}^{'}$, B, $P_{i2}^{'}$, B, $P_{i+1}^{'}$ and (2) $P_{i-1}^{'}$, $P_{i1}^{'}$, $P_{i+1}^{'}$. Since the first route has two turns whereas the second route has no turn, the second route has a higher probability of being the actual route. Thus, link BC should assigned a higher weight than BD by this criterion.

Formally, the weight for turns WS_T is given as follows:

$$WS_T = W_T * (1 - \frac{0.5 * (t_i + t_{i+1})}{MAX\_TURNS}) \tag{4}$$

where,

- W_T: the weight coefficient for turns
- MAX_TURNS: a predefined constant to adjust the weight for turns
- t_i, t_{i+1}: variables that represent the turns between $P - i$, $P_(i - 1)$ and between P_i, $P_(i + 1)$:
  - $t_i = 0$ if P_i and P_{i-1} are matched onto the same link or connected links with turn angle less than $45°$
  - $t_i = 1$ if P_i and P_{i-1} are matched onto connected links with turn angle ranging from $45°$ to $135°$
  - $t_i = 2$ if P_i and P_{i-1} are matched onto connected links with turn angle greater than $135°$
  - $t_i = Max\_Turns$ if P_i and P_{i-1} are matched onto unconnected links
  - $t_{i+1} = 0$ if P_i and P_{i+1} are matched onto the same link or connected links with turn angle less than $45°$
  - $t_{i+1} = 1$ if P_i and P_{i+1} are matched onto connected links with turn angle ranging from $45°$ to $135°$
  - $t_{i+1} = 2$ if P_i and P_{i+1} are matched onto connected links with turn angle greater than $135°$
  - $t_{i+1} = Max\_Turns$ if P_i and P_{i+1} are matched onto unconnected links

5) Total weight: The total weight is obtained by summing up the four individual weights:

$$TWS = WS_D + WS_H + WS_{LC} + WS_T \tag{5}$$

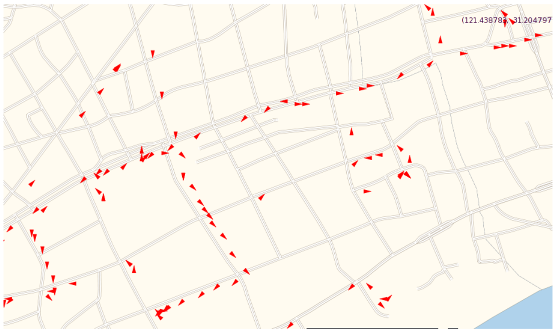


Fig. 11. An original GPS trajectory derived from GPS data

By selecting different values for the weighting coefficients (i.e., W_D, W_H, W_{LC}, W_T), the weight for each criterion can be adjusted. In previous research, the weight coefficients for the criteria were assumed to be equal [14] or determined empirically [23]. In this work, the relative importance of each criterion is determined empirically.

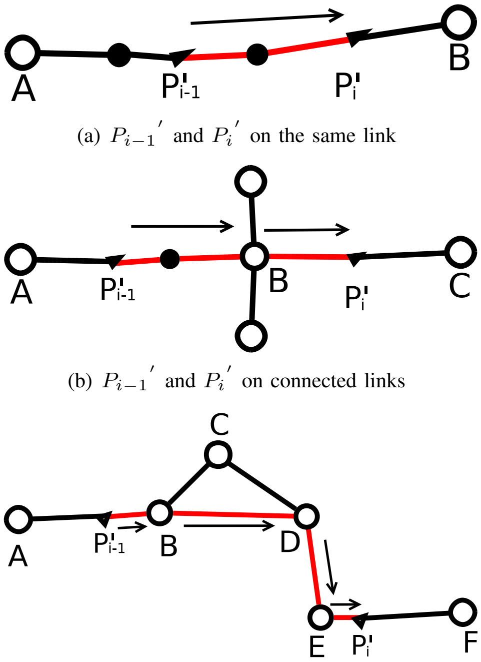
Figure 11 shows an original GPS trajectory derived from GPS points. The vehicle heading always has a great similarity with the link heading, and most GPS points are near to the actual link (98.7% in our data sets). Therefore, heading and proximity are assigned more importance than link connectivity and turns.

6) Determination of actual link: The total weight is calculated for each candidate route from the preceding GPS point P_{i-1} to the following GPS point P_{i+1}. Since the preceding GPS point has been map matched to its actual link, the number of candidate routes is decided by the number of candidate links for P_i and P_{i+1}. As the example shown in Figure 6, both P_i and P_{i+1} have two candidate links, so P_i has four candidate routes (refer to Figure 7). Each of P_i's candidate links is contained by at least one route. For each link, we only store the route that gives the highest score and the score is taken as the weight of the link. The link that has the highest score is selected to be the actual link and the closet point on that link to P_i is taken as the vehicle's location on that link.

IV. Interpolation method

The GPS trajectory data set used in this work has relatively long sampling intervals and the vehicle status during the interval are unknown. After map matching, two consecutive map-matched points may be on different links that are not even connected. To address this problem, we propose an interpolation method to determine the path between two consecutive GPS points and to generate the trajectory between the two points.

The interpolation process goes in two steps: path determination and trajectory interpolation. Path determination is to identify the path taken by the vehicle between consecutive map-matched GPS points. Trajectory interpolation is to generate the trajectory between consecutive map-matched points based on the path.



(a) P_{i-1}' and P_i' on the same link

(b) P_{i-1}' and P_i' on connected links

(c) P_{i-1}' and P_i' on unconnected links

Fig. 12. Three cases in path identification

A. Assumptions of path choosing

In this work, we make the assumption that the paths tend to be direct and drivers prefer the shortest path.

There are three cases to consider. P_{i-1}' and P_i' are two consecutive map-matched points.

- If P_{i-1}' and P_i' are on the same link, then the vehicle is assumed to move from P_{i-1}' to P_i' along this link.
- If P_{i-1}' and P_i' are on two connected links, the vehicle is assumed to move along the two connected links from P_{i-1}' to P_i'.
- If P_{i-1}' and P_i' are on unconnected links, the vehicle is assumed to move first along the link containing P_{i-1}', then along the shortest path from the end point of the link containing P_{i-1}' to the start point of the link containing P_i', and finally along the link containing P_i'.

Figure 12 shows examples of the three cases. In Figure 12(a), the vehicle moves from P_{i-1}' to P_i' by traveling along the link AB. In Figure 12(b), the vehicle first travels from P_{i-1}' to B along link AB and then from B to P_i' along link BC. In Figure 12(c), the path between P_{i-1}' and P_i' is comprised of three parts: (1) the path from P_{i-1}' to B along link AB, (2) the shortest path from B (the end point of link AB) to E (the start point of link EF), and (3) the path from E to P_i' along link EF.

B. Path determination

The path determination process is to find the path between consecutive GPS points. The path is described as a sequence of points which include nodes and shape points and two map-matched GPS points. Based on our assumption, if two

Algorithm 1 ADAPTED DIJKSTRA'S ALGORITHM

```
S := empty sequence
for each vertex v in G do
    dist[v] := $\infty$
    previous[v] := NULL
end for
dist[source] := 0
Q:= V
while Q is not empty && iter $<$ total number of vehicles do
    iter :=iter+1
    u := vertex in Q with smallest dist
    if  dist[u] = $\infty$ then
        break
    end if
    if  u = target then
        break
    end if
    remove u from Q
    for each neighbor v of u do
        dist := dist[u]+dist_between(u,v)
        if dist $<$ dist[v] then
            dist[v] := dist
            previous[v] := u
        end if
    end for
end while
u:= target
while previous[u] != NULL do
    insert u at the beginning of S
    u := previous[u]
end while
```

consecutive map-matched points are on the same link or connected links, the path between them can be easily obtained by recording the points (including nodes and shape points) between them along the links. Otherwise, the shortest path should be identified.

Given a weighted graph G = (V, E, w), |V|= n, |E| = e, the shortest path between two vertices can be efficiently computed by Dijkstra's algorithm [24]. To utilize Dijkstra's algorithm, the road network should be transformed to a weighted graph G = (V, E, w). This is achieved by replacing each link with a straight line segment. In the weighted graph G, vertices represent the nodes from the road network whereas edges the links. The link length is taken as the weight associated with the edge. A link can be completely characterized by a sequence of points (endpoints and shape points), so the length is calculated by summing up the distances between each two consecutive points on the link. This is described by the following equations:

$$link\ length = \sum_{i=1}^{n} dist(A_{i-1}, A_i) \tag{6}$$

where (A_0, A_1,...,A_n) is the sequence of points on the link, and $dist(A_{i-1}, A_i)$ is the surface distance in meters between A_{i-1} and A_i.

Then we adapt Dijkstra's algorithm to find the shortest path between the source and the target. The source is the end point of the link that contains the preceding map-matched GPS point and the target is the start point of the link that contains the current map-matched GPS point. The original Dijkstra's algorithm finds the shortest distance from the source to all other nodes. Since we are only interested in the shortest path between source and target, the search can be terminated if the shortest path to the target has been identified. By identifying all the nodes and shape points along the path, we can transform the path returned by Algorithm 1 (Adapted Dijkstra's Algorithm) to the path in the road network.

C. Trajectory interpolation

Given any two consecutive map-matched points $P_{i-1}^{'}$ and $P_i^{'}$ with time stamps t_{i-1} and t_i, the GPS trajectory between them is generated by estimating the vehicle location at time $t \in [t_{i-1}, t_i]$. To estimate the vehicle location at time t, we should know the distance the vehicle travels from $P_{i-1}^{'}$ to P_t. However, this information is not available from the GPS data set. So we make the assumption that the vehicle travels at a constant speed from $P_{i-1}^{'}$ to $P_i^{'}$. Then the distance can be calculated by (speed * traveling time). Based on the distance from $P_{i-1}^{'}$ to P_t, the vehicle location at time t on the path can be determined. The distance calculation process and location determination process are as follows.

1) Distance calculation: To get the distance from $P_{i-1}^{'}$ to P_t, we calculate the vehicle speed between $P_{i-1}^{'}$ and $P_i^{'}$. Since the vehicle travels at a constant speed from $P_{i-1}^{'}$ to $P_i^{'}$, the speed can be calculated by:

$$speed = \frac{length\ of\ path}{t_i - t_{i-1}} \tag{7}$$

where:

- t_{i-1}, t_i: the time stamps of $P_{i-1}^{'}$ and $P_i^{'}$
- length of path: the distance that the vehicle travels from $P_{i-1}^{'}$ to $P_i^{'}$, which is given by:

$$length\ of\ path = \sum_{i=1}^{n} dist(A_{i-1}, A_i) \tag{8}$$

where (A_0, A_1,...,A_n) is the sequence of points on the path, and $dist(A_{i-1}, A_i)$ is the surface distance in meters between A_{i-1} and A_i.

Given any time t $\in [t_{i-1}, t_i]$, the distance the vehicle travels from $P_{i-1}^{'}$ to P_t is given by $speed\ *\ (t - t_{i-1})$.

2) Location determination: After calculating the distance the vehicle travels from $P_{i-1}^{'}$ to P_t, we need to determine the location of P_t on the path. This involves two steps: (1) identification of the segment on which the vehicle travels at time t, and (2) determination of the vehicle location on that segment.

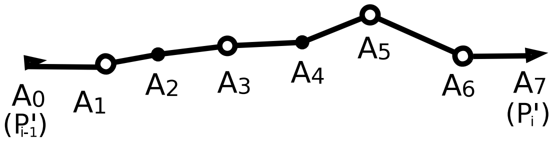


Fig. 13. Path between $P_{i-1}{}'$ and $P_i{}'$

The path from $P_{i-1}{}'$ to $P_i{}'$ is described as a sequence of points $(A_0, A_1, \ldots, A_n)$ as shown in Figure 13. $dist(t_{i-1}, t)$ denotes the distance the vehicle travels from t_{i-1} to t.

If $dist(t_{i-1}, t) \leq dist(A_0, A_1)$, then P_t is on segment (A_0, A_1). Otherwise, find the segment $A_{j-1}A_j$ which satisfies that

(1) $\sum_{i=1}^{j-1} dist(A_{i-1}, A_i) < dist(t_{i-1}, t)$, and

(2) $\sum_{i=1}^{j-1} dist(A_{i-1}, A_i) \geq dist(t_{i-1}, t)$.

This segment is the one on which the vehicle travels at time t.

After identifying the segment $A_{j-1}A_j$ which contains P_t, we need to determine the location of P_t on that segment.

The distance from P_t to A_{j-1} is given by

$$dist(A_{j-1}, P_t) = dist(t_{i-1}, t_i) - \sum_{i=1}^{j-1} A_{i-1}A_i \tag{9}$$

Coordinates of P_t (x_t, y_t) can be obtained from the endpoints of the segment as follows:

$$x_t = \alpha * (x_j - x_{j-1}) + x_{j-1} \tag{10}$$

$$y_t = \alpha * (y_j - y_{j-1}) + y_{j-1} \tag{11}$$

where:

$$\alpha = \frac{dist(A_{j-1}, P_t)}{dist(A_{j-1}, A_j)} \tag{12}$$

With the vehicle location estimated, we generate a complete GPS trajectory between consecutive map-matched points.

V. Empirical experiments

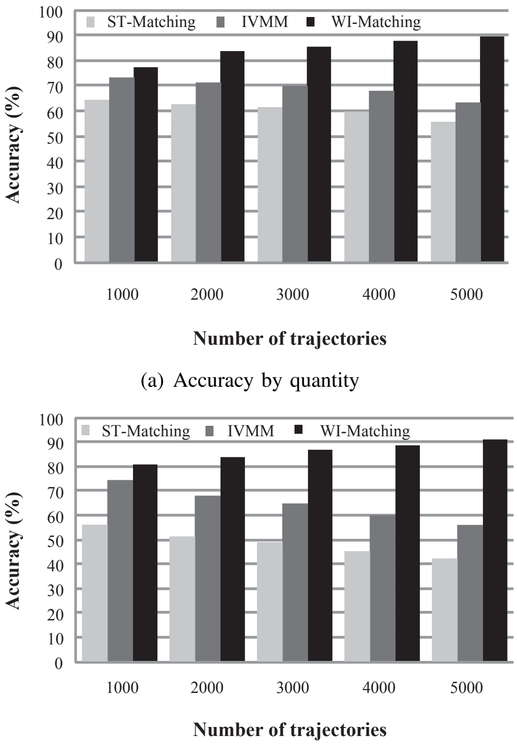
In this section, we report the empirical experiments we conducted to evaluate our method. We evaluate both efficiency and accuracy, and compare our method with two other state-of-the-art methods.

A. Experiment setup

Experiment data sets: The test data set is originated from Shanghai City Traffic Bureau, containing 5631 taxis' GPS sampling reports, and the time length is a year. We also obtained 5000 taxis' actual trajectories as the ground truth. In the city, there are 22,413 intersections connecting 33,290 road segments. The data scale we investigated is more than 200 GB.

Experiment environment: The experiments are run on a PC having an Intel Dual-Core CPU at 1.8 GHz and 4 GB main memory. The operating system is Linux Fedora.

Baseline method: We select two most recent methods, ST-Matching and VIMM, as the baseline method [3], [20], because the two methods are much more efficient and accurate than the current other methods, such as Incremental algorithm and AFD-based global algorithm [5], [14]. We take the same parameter setting as these two [3], [20] for a fair comparison.



(a) Accuracy by quantity

(b) Accuracy by length

Fig. 14. Accuracy evaluation

B. Accuracy evaluation

The accuracy of the methods is measured by two criteria as follows.

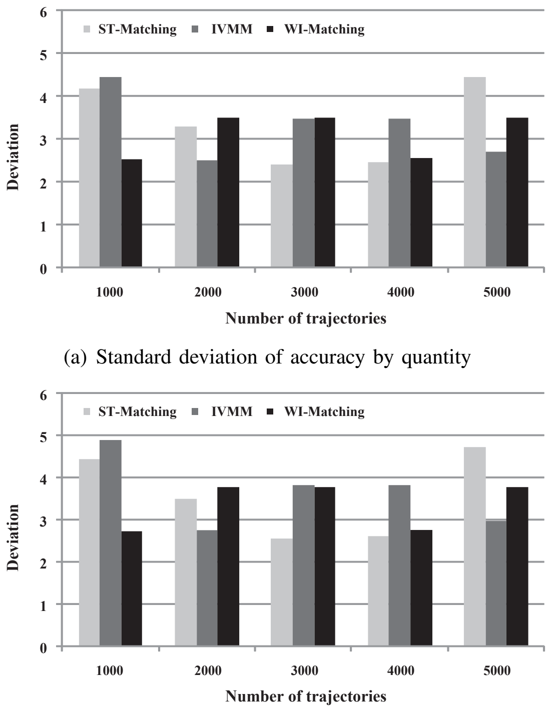
Accuracy by quantity is denoted as A_q, and calculated by the following equation.

$$A_q = \frac{\#matched\ road\ segments}{\#all\ road\ segments\ of\ a\ trajectory} \tag{13}$$

Accuracy by length is denoted as A_l, and calculated by the following equation.

$$A_l = \frac{\sum the\ length\ of\ matched\ road\ segments}{Length\ of\ a\ trajectory} \tag{14}$$

In Figure 14, we report the accuracy evaluation. In the figure, x-axis is the number of trajectories, and y-axis is the accuracy, which is measured by Eq. (13) and Eq. (14) respectively. In Figure 14(a), we report the accuracy evaluation by quantity. Our method is much better than both ST-matching and VIMM, especially with the increase in number of trajectories. The reason for this result is that in our method, we consider the road heading, and turn information. In Figure 14(b), we report the accuracy evaluation by length, and our method is also better than both ST-matching and VIMM, especially with the increase in number of trajectories. In Figure 15, we report the standard deviation of the above accuracy



(a) Standard deviation of accuracy by quantity

(b) Standard deviation of accuracy by length

Fig. 15. Standard deviation of accuracy evaluation

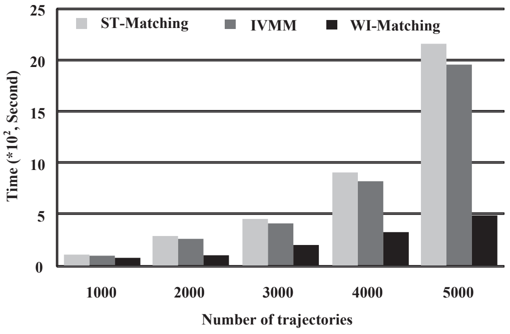


Fig. 16. Time cost in map-matching

evaluation results. In the figure, x-axis is the number of trajectories, and y-axis is the standard deviation of accuracy. Note that in the two figures, the standard deviation is between 2 and 5, which indicates that the data points tend to be very close to the mean, and the results in Figure 14 are reliable.

C. Efficiency evaluation

The efficiency evaluation is measured by the time cost to calibrate a set of vehicle GPS records. First, we evaluate the efficiency of map-matching, second, we evaluate the efficiency of interpolation, and last, the total time cost is reported.

In Figure 16, we report the time cost in map-matching. In the figure, x-axis is the number of input trajectories, and y-axis is the time cost in seconds. Our method (WI-matching) performs much faster than both ST-matching and VIMM, especially when the data set scales up.

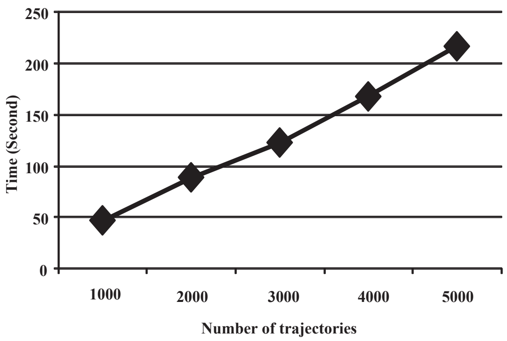


Fig. 17. Time cost in interpolation

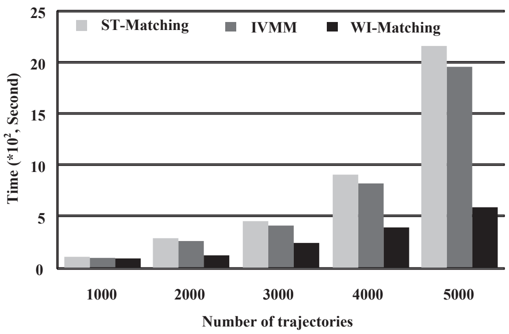


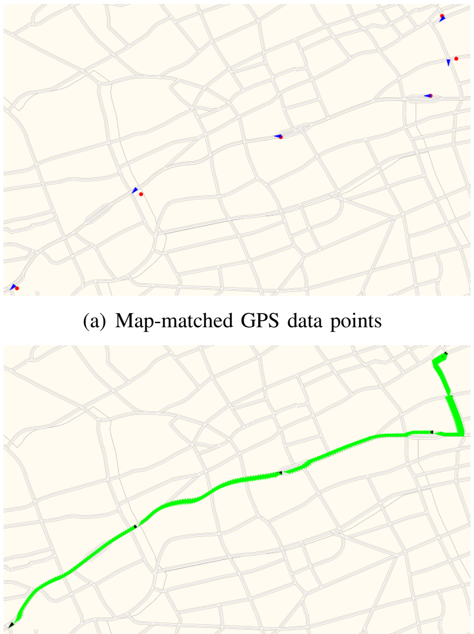
Fig. 18. Efficiency evaluation

For the interpolation, there is no such a process in ST-matching and VIMM, hence, we report the time cost of WI-matching in Figure 17. Note that the interpolation costs a linear-increasing time along with the number of trajectory scaling up. The time cost of interpolation is relative small comparing with the time cost in map-matching, while the interpolation benefits us a much more sufficient and accurate trajectory.

In Figure 18, we report the efficiency evaluation. In the figure, x-axis is the number of input trajectories, and y-axis is the time cost in seconds. Our method, WI-matching, is much faster than both ST-matching and VIMM, especially when the data set is large.

D. Large scale calibrated data set

In Figure 19, we illustrate a set of map-matched GPS points and their interpolated trajectory. Note that after the calibration, the low-sampling-rate, erroneous data is not only mapped on actual roads, but is also amended with sufficient data points to become a complete trajectory. Base on the calibrated trajectory data, we can analyze the vehicles' location information, e.g., the distribution of taxis in the city. Figure 20 shows the calibrated trajectories for the entire data set of 5361 taxis. Base on our work in [1], [2], we can find that the calibrated trajectory data can help us give insightful research on taxi's behaviors.



(a) Map-matched GPS data points

(b) Interpolated GPS trajectory

Fig. 19. Trajectory calibration results

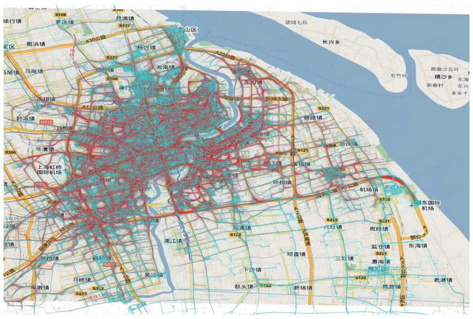


Fig. 20. Calibrated trajectory data set

VI. Conclusion and Future Work

Utilizing the geometric and topological information of the road networks and taking advantage of the impacts of the historical information on the vehicle trajectories, we propose a novel weighting-based map matching algorithm and an interpolation algorithm to calibrate the erroneous low-sampling-rate GPS trajectory data sets in this paper. The map matching algorithm matches GPS sampling points to the actual roads and the interpolation algorithm fills the gaps between consecutive map-matched points and generates a sufficient trajectory. Real world large scale data sets are utilized in our empirical experiments and the results confirm the great accuracy and efficiency of our solution.

In the future work, we are studying alternative weighting schemes as well as making the weighting schemes automatically adapt to the data set. At the same time, we will try to utilize the geometric and topological information of the road networks to tackle more vehicle mobile data related issues. Another interesting direction is that how much impact will the historical information make to the vehicle related applications and research.

VII. Acknowledgment

This research/project is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

References

[1] S. Liu, Y. Liu, L. M. Ni, J. Fan, and M. Li, “Towards mobility-based clustering,” in *Proc. of the 16th ACM SIGKDD*, 2010.

[2] H. Zhu, Y. Zhu, M. Li, and L. Ni, “SEER: Metropolitan-scale Traffic Perception Based on Lossy Sensory Data,” in *Proc. of the 28th IEEE INFOCOM*, 2009.

[3] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, “Map-matching for low-sampling-rate GPS trajectories,” in *Proc. of the 17th ACM SIGSPATIAL*, 2009.

[4] M. Quddus, “High integrity map matching algorithms for advanced transport telematics applications,” Ph.D. dissertation, Citeseer, 2006.

[5] I. Skog and P. Handel, “In-Car Positioning and Navigation Technologies: A Survey,” *IEEE Transactions on Intelligent Transportation Systems*, 2009.

[6] A. Tsui and A. Shalaby, “An enhanced system for link and mode identifications for GPS-based personal travel surveys,” *Transportation Research Record*, 2006.

[7] M. Quddus, W. Ochieng, and R. Noland, “Current map-matching algorithms for transport applications: State-of-the art and future research directions,” *Transportation Research Part C: Emerging Technologies*, 2007.

[8] G. Taylor, G. Blewitt, D. Steup, S. Corbett, and A. Car, “Road reduction filtering for GPS-GIS navigation,” *Transactions in GIS*, 2001.

[9] C. Wenk, R. Salas, and D. Pfoser, “Addressing the need for map-matching speed: Localizing globalb curve-matching algorithms,” in *Proc. of the* 18th *SSDBM*, 2006.

[10] W. Chen, Z. Li, M. Yu, and Y. Chen, “Effects of sensor errors on the performance of map matching,” *The Journal of Navigation*, 2005.

[11] Y. Meng, W. Chen, Z. Li, Y. Chen, and J. Chao, “A simplified map-matching algorithm for in-vehicle navigation unit,” *Annals of GIS*, 2002.

[12] M. Weber, L. Liu, K. Jones, M. J. Covington, L. Nachman, and P. Pesti, “On map matching of wireless positioning data: a selective look-ahead approach,” in *Proc. of the* 18th *SIGSPATIAL GIS*, 2010.

[13] A. Brilingaitė and C. S. Jensen, “Enabling routes of road network constrained movements as mobile service context,” *Geoinformatica*, vol. 11, March 2007.

[14] J. Greenfeld, “Matching GPS observations to locations on a digital map,” in *Prof. of the 81st Annual Meeting of TRB*, 2002.

[15] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, “On map-matching vehicle tracking data,” in *Proc. of the* 31st *VLDB*, 2005.

[16] W. Kim, G. Jee, and J. Lee, “Efficient use of digital road map in various positioning for ITS,” in *Proc. of IEEE PLANS*, 2000.

[17] M. Fu, J. Li, and M. Wang, “A hybrid map matching algorithm based on fuzzy comprehensive Judgment,” in *Proc. of 7th IEEE ITSC*, 2004.

[18] H. Yin and O. Wolfson, “A weight-based map matching method in moving objects databases,” in *Proc. of the 16th SSDBM*, 2004.

[19] M. Nanni and R. Trasarti, “K-bestmatch reconstruction and comparison of trajectory data,” in *Proc. of* 9th *IEEE ICDM*, 2009.

[20] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G.-Z. Sun, “An interactive-voting based map matching algorithm,” in *Proc. of the* 11th *MDM*, 2010.

[21] M. Quddus, R. Noland, and W. Ochieng, “Validation of map matching algorithms using high precision positioning with GPS,” *The Journal of Navigation*, 2005.

[22] D. Pfoser and C. S. Jensen, “Capturing the uncertainty of moving-object representations,” in *Proc. of the* 6th *SSD*, 1999.

[23] M. Quddus, W. Ochieng, L. Zhao, and R. Noland, “A general map matching algorithm for transport telematics applications,” *GPS solutions*, 2003.

[24] E. W. Dijkstra., “A note on two problems in connection with graphs,” *Numerische Mathematik*, 1959.