Shortest path and vehicle trajectory aided map-matching for low frequency GPS data

PLEASE CITE THE PUBLISHED VERSION

http://dx.doi.org/10.1016/j.trc.2015.02.017

PUBLISHER

Crown Copyright. Published by Elsevier Ltd.

VERSION

VoR (Version of Record)

PUBLISHER STATEMENT

LICENCE

REPOSITORY RECORD

Shortest path and vehicle trajectory aided map-matching for low frequency GPS data

Mohammed Quddus [a],[*], Simon Washington [b]

[a] Transport Studies Group, Department of Civil and Building Engineering, Loughborough University, Leicestershire LE11 3TU, United Kingdom
[b] Queensland ASTRA Chair of Transport, School of Civil Engineering & Built Environment, Faculty of Science and Engineering, Centre for Accident Research and Road Safety (CARRS-Q), Faculty of Health, Queensland University of Technology, 2 George St, GPO Box 2434, Brisbane, Qld 4001, Australia

ARTICLE INFO

ABSTRACT

Map-matching algorithms that utilise road segment connectivity along with other data (i.e. position, speed and heading) in the process of map-matching are normally suitable for high frequency (1 Hz or higher) positioning data from GPS. While applying such map-matching algorithms to low frequency data (such as data from a fleet of private cars, buses or light duty vehicles or smartphones), the performance of these algorithms reduces to in the region of 70% in terms of correct link identification, especially in urban and sub-urban road networks. This level of performance may be insufficient for some real-time Intelligent Transport System (ITS) applications and services such as estimating link travel time and speed from low frequency GPS data. Therefore, this paper develops a new weight-based shortest path and vehicle trajectory aided map-matching (stMM) algorithm that enhances the map-matching of low frequency positioning data on a road map. The well-known A* search algorithm is employed to derive the shortest path between two points while taking into account both link connectivity and turn restrictions at junctions. In the developed stMM algorithm, two additional weights related to the shortest path and vehicle trajectory are considered: one shortest path-based weight is related to the distance along the shortest path and the distance along the vehicle trajectory, while the other is associated with the heading difference of the vehicle trajectory.

The developed stMM algorithm is tested using a series of real-world datasets of varying frequencies (i.e. 1 s, 5 s, 30 s, 60 s sampling intervals). A high-accuracy integrated navigation system (a high-grade inertial navigation system and a carrier-phase GPS receiver) is used to measure the accuracy of the developed algorithm. The results suggest that the algorithm identifies 98.9% of the links correctly for every 30 s GPS data. Omitting the information from the shortest path and vehicle trajectory, the accuracy of the algorithm reduces to about 73% in terms of correct link identification. The algorithm can process on average 50 positioning fixes per second making it suitable for real-time ITS applications and services.

1. Introduction

Map-matching algorithms play important roles in land vehicle navigation systems to enhance the accuracy of vehicle position solutions (Scott and Drane, 1994; Zhao, 1997; Scott, 1994). A map-matching process is a method of integrating

* Corresponding author.
   E-mail address: m.a.quddus@lboro.ac.uk (M. Quddus).

digital map data with the data from a positioning system (e.g. GPS, inertial navigation system, GPS integrated with dead-reckoning) to identify the correct link on which a vehicle is travelling and to determine the precise vehicle position on that link. Based on the available road geometry, topology and characteristics of positioning data, a range of mathematical methods have been utilised in these map-matching algorithms. According to Quddus (2006), map-matching algorithms are classified into three groups: geometric (Scott, 1994; Bernstein and Kornhauser, 1996, 1998; White et al., 2000; Krakiwsky et al., 1988; Jo et al., 1996), topological (Bouju et al., 2002; Greenfeld, 2002; Meng et al., 2003; Srinivasan and Cheu, 2003; Quddus et al., 2003; Velaga et al., 2009) and advanced (Fuchs et al., 1980; Kim et al., 2000; Najjar and Bonnifait, 2003; Syed and Cannon, 2004; Quddus et al., 2006).

A geometric map-matching algorithm uses only geometric information such as the "shape" of line segments (i.e. road centre-lines that define the road network and not the way in which they are connected). Different types of geometric map matching algorithms have emerged: point-to-point matching; point-to-curve matching; and curve-to-curve matching. A geometric map-matching algorithm can be applied to any frequency positioning data as the required input data include only position fixes (*x*- and *y*-coordinates) and a base road network map. The accuracies of geometric map-matching algorithms are, however, low ranging between 80% and 85% in correct link identification. A topological map-matching algorithm employs topological information such as connectivity of road segments, bearing of road segments, and attributes of the roads (e.g. turn restriction at junctions, road width). Greenfeld (2002) developed a weighted topological map-matching process based on a topological analysis of a road network and the coordinate information of observed GPS fixes. The weights are assigned to the proximity of a GPS fix relative to a road segment, the heading difference between the bearing of a road segment and the vehicle heading determined by a line joining two consecutive GPS fixes and the intersection angle between a road segment and a line formed by two consecutive GPS fixes when it is applicable. Heading and speed information obtained from GPS are not utilised in the Greenfeld's algorithm (Greenfeld, 2002). Since this method is mainly based on the GPS positioning fixes, the performance relies on the satellite geometry and visibility as well as road network complexity (Taylor et al., 2001; Quddus et al., 2003; Ochieng et al., 2004). The topological map-matching was further enhanced by introducing more topological information such as turn restriction at junctions, road curvature and road segment connectivity (e.g. Quddus et al., 2003; Meng et al., 2003; Velaga et al., 2009). The performance of topological map-matching algorithms is much better than geometric algorithms and the accuracy ranges from 80% to 96% in correct link identification (Velaga et al., 2009). One of the primary features of topological map-matching algorithms is that they are only effective in for matching high frequency positioning data, because the segment connectivity is not useful when matching low frequency positioning data.

Advanced map-matching algorithms use advanced statistical, mathematical and artificial intelligence techniques such as the Kalman Filter (Kim et al., 2000), Belief Theory (Najjar and Bonnifait, 2003), a fuzzy logic method (Syed and Cannon, 2004; Quddus et al., 2006; Fuchs et al., 1980) or the application of Bayesian interference (Pyo et al., 2001). Most of advanced map-matching algorithms are designed for use with high frequency (i.e. 1 Hz) positioning data. They utilise the additional information such as the positioning data quality, vehicle heading error and vehicle speed and therefore, the performance is better than other map-matching algorithms.

Most of the previously described map-matching (MM) algorithms (e.g. geometric MM, topological MM or fuzzy logic MM) are designed for use with high frequency positioning data (i.e. 1 Hz or 1 s sampling interval). They normally use information such as topology (i.e. road connectivity), turn restrictions, perpendicular distance to the candidate link, the bearing difference between the vehicle heading and the link direction to identify the correct link. Such information is available only with high frequency GPS data. There are many other ITS applications (e.g. fleet management, travel time estimation, bus arrival information as bus stops) in which the recording of low frequency (0.2 Hz or 5 s interval; 0.033 Hz or 30 s interval; 0.0166 Hz or 60 s interval) positioning data is the norm so as to reduce the cost of data storage and/or transmission. Current map-matching algorithms may not be able to achieve the required accuracy in terms of correct link identification and location determination to correctly match low frequency data. This is primarily due to the fact that two important factors–connectivity and turn restrictions—cannot be derived from low frequency data. This limitation is shown in Fig. 1 in which there are two vehicle position fixes denoted as P_1 and P_2. Assume that positioning data are updated every minute corresponding with a sampling frequency of 0.0167 Hz. P_1' implies the corrected match position for P_1. The map-matching task here is to match P_2 onto the correct link. d_i is the perpendicular distance from P_2 to a candidate link i. Assuming that there are three candidate links for P_2 which are denoted as links DA, AB and BA, the direct connectivity between the previous matched segment and the correct candidate segments (DA, AB and BA) does not exist. Additionally, turn restriction data at junction A are also not useful for this case since there is no evidence that the vehicle is travelling from link FA or GA. If the perpendicular distances from P_2 to link DA and AB are the same, then only the bearing difference between the vehicle direction and the candidate link direction would not be enough to identify the correct segment among these three segments.

This suggests that alternative data must be sought to correctly identify the road segment among the candidate segments for the low frequency data. In the low frequency positioning data (e.g. 0.2 Hz; 0.033 Hz; 0.0167 Hz), vehicles can travel up to 1.8 km distance between two consecutive position fixes (assuming a maximum speed of 110 km/hr with 0.0167 Hz frequency positioning data). Since no data are available within this distance travelled, it can be assumed that vehicles follow the shortest path and this information could be employed as additional variable in the map-matching process to enhance the matching of low frequency positioning data. It is however possible that vehicles do not follow the shortest path all the time, especially buses that follow pre-defined routes. Therefore, the traditional perpendicular distance (*d*) and the bearing difference between the link direction and the vehicle heading from GPS could still be used in the map matching procedure.
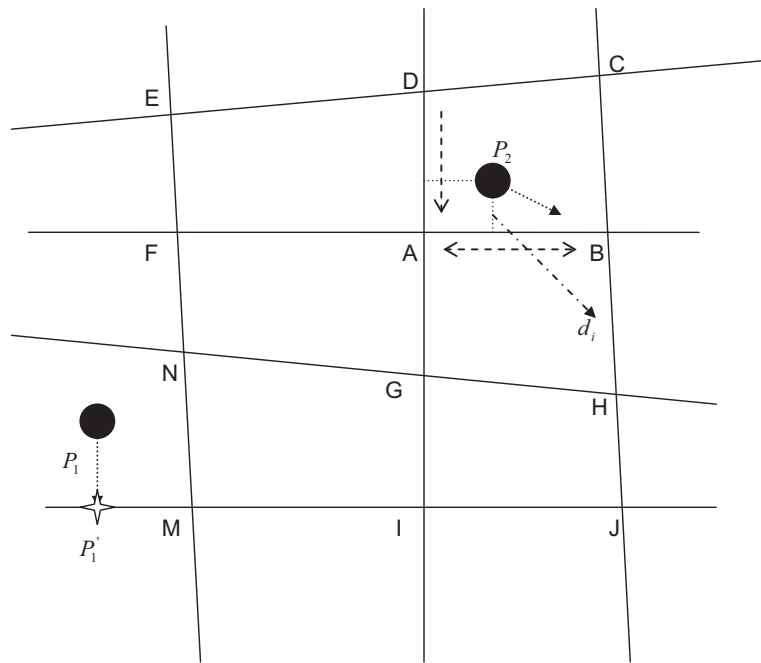
Fig. 1. The problem with a current map matching algorithm for the case of matching low frequency data.

The purpose of this paper is therefore to develop a new weight-based map-matching method by considering the perpendicular distance, the bearing difference and the extra information from the shortest path and vehicle trajectory. The developed algorithm is tested and validated across various low frequency positioning data collected by GPS.

The paper is organised as follows: the next section explains how the information from the shortest path and vehicle trajectory can aid in identifying the correct link. It is followed by a brief description of the A* shortest path search algorithm suitable for a map-matching process. The introduction of the weight-based shortest path and vehicle trajectory aided map-matching algorithm is then presented. This includes a description of both the initial map-matching process and the subsequent map-matching process. This is followed by map-matching testing and validation. Conclusion and future research directions are given at the end of the paper.

2. How does the shortest-path and vehicle trajectory aid a map-matching process?

The shortest path can be determined between the previous position fix and a point on the candidate link using the commonly employed A* shortest path search algorithm which is introduced in the following section. The distance travelled by a vehicle along the shortest path can then be determined between these two points. This distance is compared with the distance along the vehicle trajectory (i.e. the Euclidean distance between two consecutive points) (see Fig. 2). In this figure, there are two candidate links for P_2 denoted as link BE and CD. Points P_2^1 and P_2^2 are the points (candidate map-matched points) on link BE and CD respectively. The shortest path between P_1' and P_2^1 is $P_1' \rightarrow A \rightarrow B \rightarrow P_2^1$. The distance along this path is l_1. Again, the shortest path between P_1' and P_2^2 is $P_1' \rightarrow A \rightarrow B \rightarrow C \rightarrow P_2^2$. The distance along this path is l_2. The distance along the vehicle trajectory is therefore:

l = The Euclidean distance between P_1 and P_2.

It can be seen that $l_1 < l_2$ meaning that the difference between l_1 and l is smaller than that of between l_2 and l. This suggests that among the two candidates links (CD and BE), link BE is more likely to be the correct segment for the position fix P_2.

The heading between two consecutive matched points (i.e. the map-matched point for the previous fix P_1' and the current candidate map-matched point P_2^1) can be obtained. This heading can then be compared with the heading along the vehicle trajectory to identify the correct link. Two simple examples are given to illustrate the concept in Fig. 3a and b. P_1 and P_2 are the two consecutive position fixes in both examples. Assuming that P_1 is correctly matched to P_1', the task here is to identify the correct link for position fix P_2. There are two candidate links for P_2 denoted as BE and BC (see Fig. 3a). Candidate map-matched positions on link BE and BC are P_2^1 and P_2^2 respectively. If perpendicular distances from P_2 to P_2^1 and P_2^2 are the same (see Fig. 3b), the shortest path between P_1' and P_2^1 and the shortest path between P_1' and P_2^2 would be the same and therefore, the shortest path may not aid in identifying the correct link among candidate links BE and BC. Therefore, the heading along

Fig. 2. The concept of the shortest path with respect to a map-matching algorithm.

Fig. 3. Additional information from the vehicle trajectory that can aid the map-matching process.

lines $P_1' \rightarrow P_2^1$ and $P_1' \rightarrow P_2^2$ can be derived and compared with the vehicle trajectory heading along $P_1 \rightarrow P_2$. In Fig. 3, the heading along the line $P_1' \rightarrow P_2^1$ is more closer to the heading along the line $P_1 \rightarrow P_2$ suggesting that link BE is more likely to be the correct link among these two candidate links for position fix P_2.

In summary, it can be said that both the distance along the shortest path and the heading difference from the vehicle trajectory can be used in the map-matching process. This is fully explained in the following sections.

3. A* shortest path and map-matching

A* (also known as A star) searching algorithm uses the best-first search process to find the least-cost path from a given initial node to the goal node (Dechter and Pearl, 1985). It has been widely used in path finding and graph traversal. It uses a distance-plus-cost heuristic function (i.e. $f(x)$) to find the lowest $f(x)$ value of the path). In the case for a road network, the $f(x)$ is the sum of two functions:

1. The path-cost function $g(x)$ which is the distance from the starting node to the current node.
2. The heuristic estimation function $h(x)$ which is the distance from the current node to the goal node.

It chooses the lower-cost path segment (i.e. the smallest value of $f(x)$) as the shortest path and is processed continuously until the goal node is reached. For the purpose of map-matching, the connectivity among the road segments along with the turn restriction information at junctions need to be utilised in the A* shortest path searching. Fig. 4 explains the processing of A* searching algorithm in the case of a map-matching process. In Fig. 4, a vehicle is travelling from node A to node F in a hypothetical network in which link BA and FC are the one-way segments. The processing is shown as follows:

Step one:

(1) by considering the connectivity:

$$f(AC) = 10 + 22, \quad f(AB) = 18 + 12, \quad f(AE) = 40 + 30$$

(2) by considering the turn restriction:

$$f(AC) = 10 + 20 \quad \text{and } f(AE) = 40 + 30$$

$f(AB)$ is removed from the candidate set even it has the lowest value of $f(x)$. This is because AB is a one-way segment and the vehicle cannot travel from node A to node B.

Step two:

(1) by considering the connectivity:

$$f(CF) = 32 + 0, \quad f(CB) = 20 + 12, \quad f(AE) = 40 + 30$$

(2) by considering the turn restriction, $f(CF)$ is removed as CF is a one-way segment and the vehicle cannot travel from node C to node F.

Step three:

(1) by considering the connectivity:

$$f(BF) = 32 + 0, \quad f(BD) = 30 + 10, \quad f(AE) = 40 + 30.$$

(2) by considering the turn restriction, none of them is deleted.

Fig. 4. An example of A* shortest path searching method.

Therefore, the shortest path for a vehicle travelling from node A to node F is found to be AC → CB → BF. For the case of a map-matching algorithm, the shortest path algorithm also helps to reduce the number of candidate links and hence increases efficiency, accuracy and reliability. If no shortest route between a candidate link and the previous matched link is found, the candidate link should be removed from the set of candidate links.

4. Weight-based shortest path and vehicle trajectory aided map-matching algorithm

In this paper, a shortest path and vehicle trajectory based Map-Matching (stMM) algorithm suitable for low frequency data is developed. The algorithm uses the shortest path and heading of vehicle trajectory as additional information to aid in identifying the correct link. The A* searching algorithm is used to find the shortest path which considers both the connectivity between the road segments and turn restrictions at junctions. The stMM algorithm developed in this paper has two key stages: (1) Initial Map-Matching Process (IMP) and (2) Subsequent Map-Matching process (SMP). Each of these processes is briefly discussed below:

4.1. The initial map matching process

The input to the initial map-matching process is:

1. an initial position fix (x, y),
2. the distance difference between the shortest path (from the first positioning fix to one of the candidate map-matched points corresponding to the second positioning fix) and the vehicle trajectory formed by the first two epochs,
3. the heading difference between the heading of the first two consecutive positioning fixes (i.e. the heading of the vehicle trajectory) and the heading from the first position fix to one of the candidate map-matched points,
4. the perpendicular distance from a positioning fix to a candidate link,
5. the bearing difference between the vehicle direction and the candidate link direction.

Since the shortest path using A* searching algorithm needs to be computed between a start node and an end node, the positioning data from two epochs are needed to execute the IMP. For example, as shown in Fig. 5, there are two consecutive positioning fixes denoted as P_1 and P_2 in which vehicle position P_i ($i = 1, 2$) includes a number of attributes such as the east coordinate x_i, the north coordinate y_i, the speed of vehicle v_i and the vehicle bearing a_i. This can be expressed as:

$$P_i = \{x_i, y_i, v_i, a_i\}$$

The goal of the IMP is then to match the second position P_2 to the correct road segment rather than the identification of the correct link at the first position P_1. This is to avoid the possibility of wrong matching at P_1 with limited information to perform map-matching. There are four candidate links within the confidence region at P_2 as shown in Fig. 5.
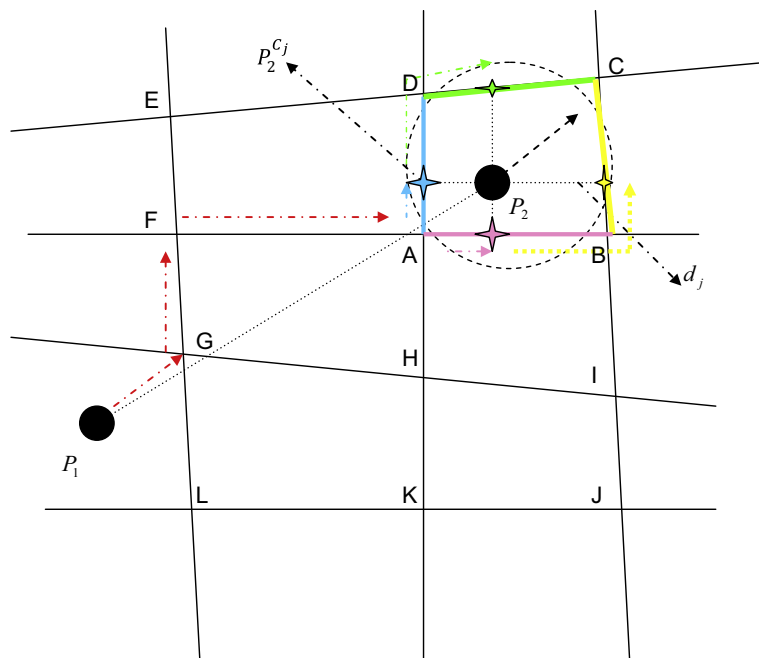

Fig. 5. Initial map-matching process.

The set of candidate links can be expressed as:

$$C_j = \{AB, BC, CD, DA\}$$

in which j is the number of the candidate segments. Four candidate map-matched positions can be found on the candidate links C_j which are:

$$P_2^{C_j} = \{P_2^{AB}, P_2^{BC}, P_2^{CD}, P_2^{DA}\}$$

The confidence region is a circle decided by the GPS positioning quality at P_2. From these candidate map-matched positions $P_2^{C_j}$ to the first position fix P_1, four shortest paths can be determined using the A* searching algorithm. These four paths may share some common links as shown by the red arrows in Fig. 5 or may be completely independent to each other.

This stMM algorithm subsequently uses the same concept of weight-based map-matching processes proposed by Greenfeld (2002), Quddus et al. (2003), and Velaga et al. (2009). Two new weights related to the shortest path and vehicle trajectory are introduced. Once the shortest paths are calculated, four different weights can be assigned to each of the candidate links for the purpose of identifying the correct link. These are: (1) weight for the perpendicular distance (PD), (2) weight for the bearing difference, (3) weight for the shortest-path distance (SD) and (4) weight for the vehicle trajectory heading difference (HD). Each of the weights is expressed as a functional form:

$$
\begin{aligned}
PD &= W_d f(d_j) \\
BD &= W_b f(\Delta\alpha) \\
SD &= W_s f(\Delta l) \\
HD &= W_h f(\Delta\theta)
\end{aligned}
\tag{1}
$$

W_i are the weight coefficients indicating the relative importance of different weights. The weight for the perpendicular distance (PD) is a function of d_j that is the perpendicular distance from P_2 to the candidate link C_j and the smaller the d_j, the higher the weight. Therefore, $f(d_j)$ is a positive function (ranges from 0 to 1) of the perpendicular distance d_j that directly contributes to the weighting score for PD. It is defined as:

$$f(d_j) = \begin{cases} 1 - d_j/200 & \text{if } d_j \leqslant 200 \text{ m} \\ 0 & \text{if } d_j > 200 \text{ m} \end{cases} \tag{2}$$
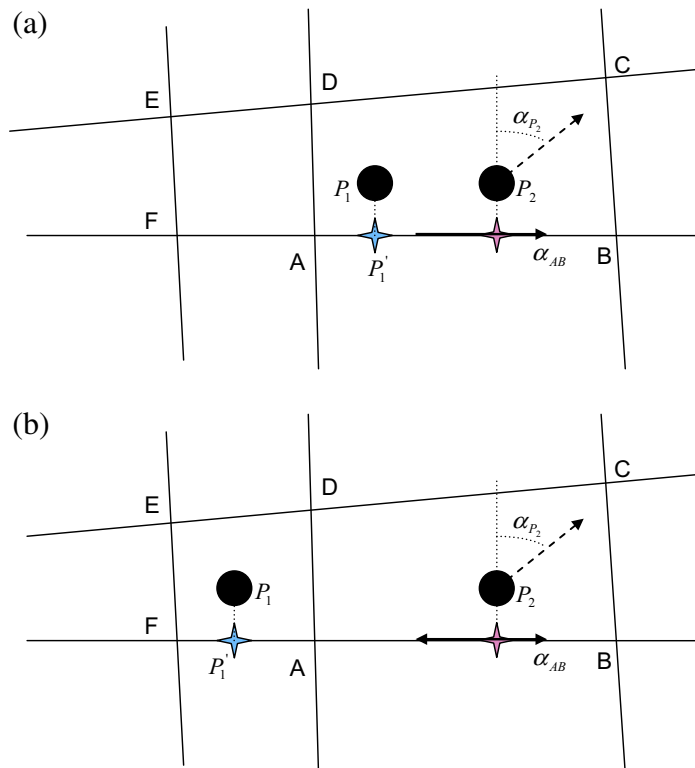

Fig. 6. The example of bearing difference.

The smaller the perpendicular distance the higher the weighting score for PD i.e. if point P_2 is closer to a candidate link, this link has a high possibility to be the correct link. As suggested by Quddus (2006), a candidate link with the perpendicular distance higher than 200 m produces a weighting score of 'zero', indicating that the positioning fix is far away from the candidate segment and therefore, this large distance becomes less important in identifying the correct segment.

The second term in Eq. (1) is the bearing difference between the vehicle heading and the candidate link direction in which α is denoted the bearing of the vehicle (relative to the North), a_j is denoted the bearing of the candidate link C_j (relative to the North). Therefore, $f(\Delta\alpha)$ is a positive function (ranges from 0 to 1) of bearing difference that directly contributes to the weighting score for BD:

$$f(\Delta\alpha) = |\cos(\Delta\alpha)| \tag{3}$$

In which $\Delta\alpha = |\alpha - \alpha_j|$. The absolute value is used here for $f(\Delta\alpha)$ to reduce the angle difference from (0–180°) to (0–90°). This is different from the existing map matching algorithms (Greenfeld, 2002; Quddus et al., 2003). The reason is simply explained with the aid of Fig. 6. As mentioned, existing map-matching algorithms are normally designed for 1 Hz frequency positioning data. In Fig. 6a, if P_1 and P_2 are obtained from a 1 Hz positioning dataset, P_1 is matched on link AB (i.e. P_1') the vehicle should travel from node A to node B. Therefore, the bearing of link AB should be $\overrightarrow{AB}$, which is $\alpha_{AB} = 90°$ (related to the North). Assumed that the vehicle bearing is $\alpha_{P_2} = 30^0$ (related to the North) at P_2, the bearing difference between the vehicle heading and link AB is 60°. The weight of $\cos(60°) = 0.5$ is then assigned to link AB according to an existing map-matching algorithm. However, if P_1 and P_2 are obtained from a low frequency positioning dataset (as shown in Fig. 6b), there is prior indication whether the vehicle is travelling from node A to node B or vice versa. The vehicle could travel through either of the two nodes of link AB. The bearing of link AB could be either $\overrightarrow{AB}$ or $\overrightarrow{BA}$. Therefore, the bearing differences could be either 60° (i.e. between $\alpha_{P_2} = 30^0$ and $\alpha_{AB} = 90°$) or 240° (i.e. between $\alpha_{P_2} = 30^0$ and $\alpha_{BA} = 270°$). The weight for link AB could be either $\cos(60°) = 0.5$ or $\cos(240°) = -0.5$. Considering that the vehicle could travel through either of the two nodes of link AB in the case for low frequency data, the stMM algorithm gives the higher weight to the low bearing difference for link AB, which is $|\cos(60°)| = |\cos(240°)| = 0.5$. Therefore, the range of angle difference is reduced to (0–90°).

The third term in Eq. (1) is the distance difference between the distance of two consecutive position fixes and the shortest path distance from the first position fix to the current map-matched point on the candidate link. l denotes the Euclidean distance between the two position fixes. l_j is the distance along the shortest path from $P_2^{C_j}$ to P_1. The function for the distance difference in the SD weighting score is therefore proposed as:

$$f(\Delta l) = \begin{cases} 1 - \Delta l/1000 & \text{if } \Delta l \leqslant 1000 \text{ m} \\ 0 & \text{if } \Delta l > 1000 \text{ m} \end{cases} \tag{4}$$

where $\Delta l = |l - l_j|$ and the functional value of $f(\Delta l)$ ranges from 0 to 1. The higher weight should be given to the smaller difference in distances. 1000 meter[1] is used as a threshold for this function and also for the shortest path search process. If the distance-plus-cost heuristic function (i.e. $f(x)$) in the shortest path search algorithm exceeds 1000 m plus the distance between two consecutive position fixes, the shortest path algorithm stops the search process and the functional value of the weighting score for SD, $f(\Delta l)$, is assigned to zero for the candidate link. This threshold is extremely useful for the shortest path searching as this improves the processing time and consequently the stMM algorithm becomes suitable for real-time applications. This threshold can be dynamic based on the frequency of the positioning data to further improve the processing time of the algorithm (e.g. the smaller threshold can be assigned to the higher frequency data).

The last term in Eq. (1) is the weighting score for the difference of directions between θ and θ_j, where θ is the vehicle trajectory heading between P_1 and P_2 (relative to the North) and θ_j is the direction between P_1 and $P_2^{C_j}$ (relative to the North). Therefore, $f(\Delta\theta)$ is a positive function (ranges from 0 and 1) and takes the following form:

$$f(\Delta\theta) = |\cos(\Delta\theta)| \tag{5}$$

where $\Delta\theta = |\theta - \theta_j|$. The high weight should be given to the smaller. Similar to the bearing difference, the range of angle differences is also reduced to (0–90°).

In Eq. (1), four weight coefficients (i.e. W_d, W_b, W_s, W_h) are used to provide the relative importance of different weighting scores and as such, the sum of these four weight coefficients should be equal to a fixed value say 100. How do we optimally estimate these weighting parameters? The values of these weights can be obtained by either an optimisation analysis (Velaga et al., 2009) or an empirical analysis (Quddus, 2006). Therefore, the total weighting score (TWS) is given as:

$$\text{TWS} = W_d f(d) + W_b f(\Delta\alpha) + W_s f(\Delta l) + W_h f(\Delta\theta) \tag{6}$$

In this paper, a Genetic Algorithm (GA) based optimisation technique is employed. Error in the map-matching process in identifying the correct links largely depends on the values of these four weight coefficients. If we have access to a reference dataset with true input and output then a set of coefficient values would result in a specific percentage of error in correct link

---

[1] This has been derived empirically from a reference input–output map-matching dataset consisting of both urban and rural road configurations and GPS data. A range of sensitivity analyses suggest that a value of 1000 m provides the lowest error in identifying the correct links. 1000 m can also significantly improve the processing time of map-matching and also guarantees the shortest path searching by considering the average length of road segments.
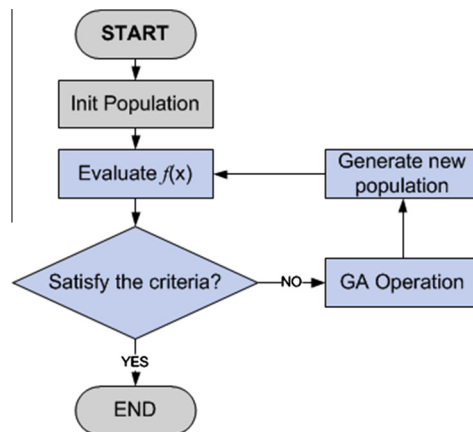
Fig. 7. A GA process to minimise map-matching errors.

identification. A functional relationship between the percentages of map matching error and the corresponding weight coefficients can be formed (see Velaga et al., 2009). Therefore, a GA algorithm can be employed to identify the values of weight coefficients that minimise the percentage of map matching error. This becomes an error minimisation problem in which two boundary conditions can be imposed.

$$MM_{error} = f(W_d, W_b, W_s, W_h)$$

Objective function:

$$\min f(x)$$

where
$$x = \exp\{[\beta_{d1}W_d + \beta_{b1}W_b + \beta_{s1}W_s + \beta_{h1}W_h] + [\beta_{d2}W_d^2 + \beta_{b2}W_b^2 + \beta_{s2}W_s^2 + \beta_{h2}W_h^2] + [\beta_{d3}\frac{1}{W_d}$$

$$+ \beta_{b3}\frac{1}{W_b} + \beta_{s3}\frac{1}{W_s} + \beta_{h3}\frac{1}{W_h}] + [\beta_{db}W_dW_b + \beta_{sh}W_sW_h + \beta_{ds}W_dW_s + \ldots\ldots\ldots.]\}$$

Subject to : $W_d + W_b + W_s + W_h = 100; 1 \le \{W_d, W_b, W_s, W_h\} \ge 97$

By employing a reference ("true") input–output map-matching dataset consisting of 2130 GPS epochs and the following error minimisation function was obtained (see Velaga et al., 2009 for details):

$$MM_{error} = 0.2W_d + 0.23W_b + 0.3W_s + 0.08W_h + 0.006W_s^2 - 0.0007W_dW_b \qquad (7)$$

In the GA optimisation process, a cross-over rate of 40% with a mutation rate of 60% along with a population size of 50 (with uniformly distributed values between 1 and 100) were employed. After 1230 generations the function value (fitness value) achieves the global optimal values. The process is shown in Fig. 7.

The optimal weight coefficient values were found to be $W_d = 32, W_b = 21, W_s = 35$ and $W_h = 12$. As can be seen, a slightly more than 50% of the total weights are assigned to the traditional PD and BD weights implying that these two weights are very useful for all frequency data. Considering that the GPS heading is poor when the vehicle speed is very low (less than 3 m/s), the higher weight was found to be associated with the PD. This finding is consistent with that reported by Quddus et al. (2006). The reason for assigning only 35% of the total weight to the shortest path distance is that a vehicle may not always travel on the shortest path. In such cases, other weights are capable of rectifying the error that may be caused by the SD weight.

The link which gives the highest weight is taken as the correct link among the candidate links and the map-matched position on the candidate link is considered as the current map-matched position on the road.

4.2. The Subsequent Map-Matching Process (SMP)

After successfully implementing the IMP, the Subsequent Map-Matching Process (SMP) is used to continuously match the following position fixes. The processing of the SMP is quite similar to that of the IMP. It also needs two position fixes: the previous position fix P_{i-1} and the current position fix P_i. Since the previous position fix P_{i-1} has been correctly matched on a candidate link by the IMP (or the SMP), the shortest path search algorithm searches the shortest path from the previous map-matched position to the candidate map-matched position on a candidate link. An example of the processing of SMP is shown in Fig. 8. To match current vehicle position fix P_i, the previous map-matched position P_{i-1} is used to find the shortest path to the candidate map-matched positions $P_i^{C_j}$. Similar to Fig. 5, there are also four candidate links within the confidence
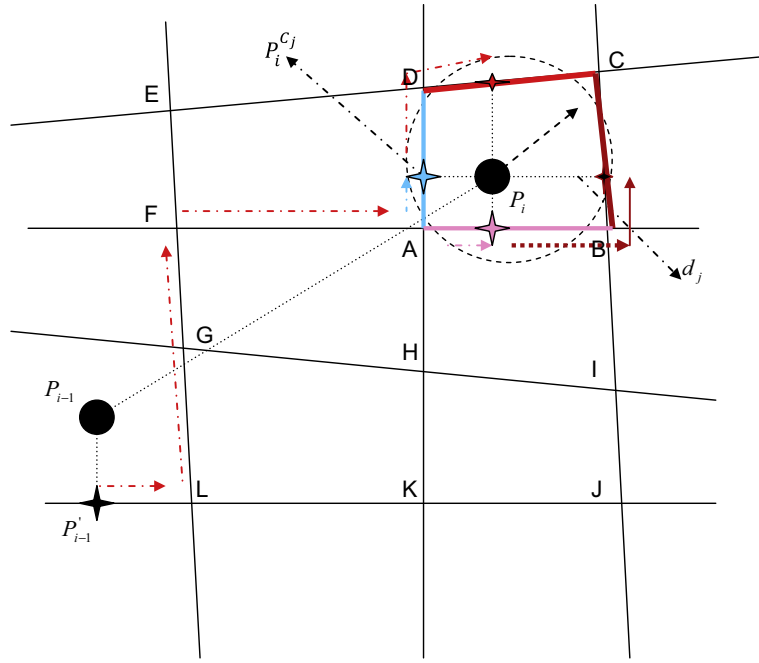
Fig. 8. Subsequent map-matching process.

region of P_i in Fig. 8 (C_j = {AB, BC, CD, DA}, and j is the number of the candidate segments). Four candidate map-matched positions can be found on the candidate link C_j which are

$$P_i^{C_j} = \{P_i^{AB}, P_i^{BC}, P_i^{CD}, P_i^{DA}\}, \quad \text{(in which } i \text{ is the the current epoch)}.$$

Then the A* search algorithm is used to search the shortest paths from the previous matched position P_{i-1} to the candidate map-matched positions $P_i^{C_j}$.

After the shortest paths are found, Eqs. (1)–(6) are used to calculate the weight for each of the candidate links. The link which gives the highest weight is taken as the correct link among the candidate links, and the matched position on the candidate link is considered as the current vehicle map-matched position on the road.

5. Testing and validation

The data were collected from urban, suburban and rural areas in Nottingham and Birmingham, UK to evaluate the performance of the shortest path based MM algorithm developed in this paper. A standard low-cost GPS receiver (i.e. a Ublox 4P receiver) was installed on a vehicle and used to collect the 1 Hz positioning data. For the correct link identification, a high accurate integrated navigation system consisting of a high-grade Inertial Navigation System (INS) and GPS was also used to collect high accuracy (i.e. within 5 centimetres at the 95% of the time) reference vehicle position fixes. The test vehicle was travelling randomly on the chosen road networks with varying complexities including fly-overs and the correct travel routes are identified by utilising the reference vehicle trajectory and the digital base map in a post-processing mode.

For the purpose of testing the developed stMM algorithm for low frequency positioning data, three additional datasets were artificially generated from the 1 Hz data: the first one is for every 5 s data (0.2 Hz), the second one is for every 30-s data (0.033 Hz) and the third one is for every 60-s data (0.0167 Hz). The optimal weight coefficient values were estimated using a Genetic Algorithm (GA). There are: $W_d = 32$, $W_b = 21$, $W_s = 35$ and $W_h = 12$. The snapshot of the map-matching results from all four datasets is shown in Fig. 9 for the case of a complex roundabout in Nottingham. The results are presented in Table 1.

As can be seen in Table 1, for one second data, the weighted stMM algorithm finds the correct links for 7829 epochs out of a total 7940 epochs resulting in a 98.6% correct link identification which is similar to the performance of most existing map-matching algorithms (see Quddus, 2006; Velaga et al., 2009). This is because that the shortest path provides the information that is similar to the connectivity information in the traditional map-matching algorithms, and the heading difference of vehicle trajectory gives the turn restriction as in the traditional map-matching algorithms. Reducing the data frequency to every five seconds, the stMM algorithm still provides a 98.4% correct link identification (i.e. 1563 correct epochs out of a total 1588 epochs). For every 30-s and 60-s data, the success rates of the algorithm are 98.9% (i.e. 263 correct epochs out of a total 266 epochs) and 98.5% (i.e. 133 correct epochs out of a total 135 epochs). The horizontal accuracy in Table 1 is obtained by comparing the map-matching positions to the reference vehicle positions obtained by high-a grade GPS/INS system. An examination of the mismatches in identifying the correct links reveals that the majority of the incorrect matches are due to the
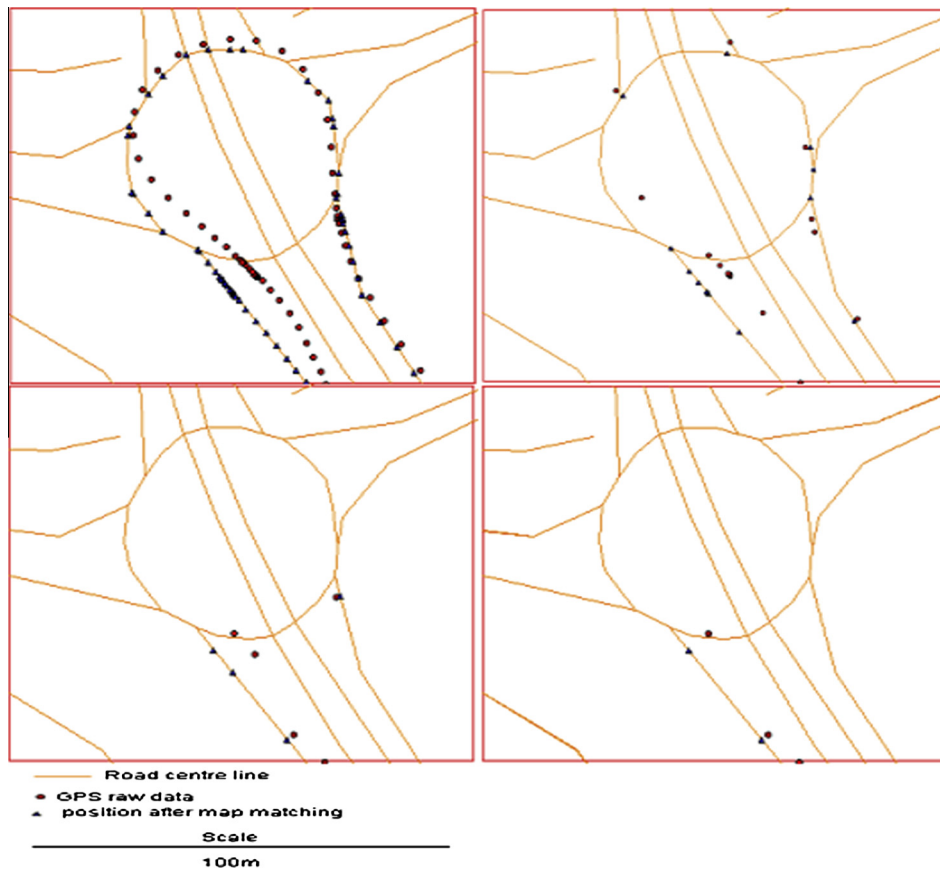
Fig. 9. Map-matching results for all four frequency data at a junction within the test route.

Table 1
The performance of the developed map-matching algorithm for different frequency GPS data.

Variables	Frequency of positioning data from GPS			
	1 s	5 s	30 s	60 s
Sample size	7940	1588	266	135
Corrected link identification (%)	98.6	98.4	98.9	98.5
Horizontal accuracy (2σ, m)	7.35	8.61	3.42	2.93
Processing time	<100 s	<30 s	<15 s	<10 s

assumption that a vehicle travels along a shortest path. This assumption sometimes led to the erroneous weighting scores due to inaccurate the functional values associated with the shortest path $W_s f(\Delta l)$ and the vehicle trajectory $W_h f(\Delta\theta)$.

In summary, if only the perpendicular distance (PD) and the bearing difference (BD) are used in identifying the correct links as this is done in the traditional map-matching algorithms, the correct identification rate is around 73%. But with the proposed four variables, this rate increases to over 98%. Therefore, the stMM algorithm provides a significant improvement in the map-matching of low frequency positioning data. The processing time is a key issue for any real-time applications. The threshold which is given to the shortest path searching reduces the processing time significantly and avoids the dead loop which could be happened in the shortest path searching. The stMM algorithm can process around 50 positioning fixes per second with the threshold of 1000 m. If no threshold is considered then the stMM becomes computationally less efficient by ten-fold.

6. Conclusions

Matching low frequency GPS positioning data on a road map have always been a big challenge in implementing real-time ITS applications and services. In this paper, a new shortest path and vehicle trajectory aided map-matching (stMM) algorithm was developed to match low frequency GPS data on a road map. An efficient A* shortest path search algorithm was employed to derive the shortest path between two consecutive GPS fixes. Both link features (i.e. connectivity) and turn

restrictions at junctions were taken into account while deriving the shortest path using the A* algorithm. In addition to the input currently used in existing map-matching algorithms such as the proximity and the bearing difference, two additional weights related to the shortest path distance and the heading difference of vehicle trajectory were introduced to enhance map-matching of low frequency positioning data. One weight is related to the distance along the shortest path and the distance along the vehicle trajectory. The other weight is associated with the heading difference derived from the vehicle trajectory and map-matched trajectory. Since the A* search algorithm used in the stMM algorithm was required to utilise in each of the epochs, the computational time was an issue as the stMM develops to support real-time ITS applications. In order to increase the efficiency of the A* algorithm, a threshold (1000 m) value was used so that the search for the shortest path stops when the distance along the candidate shortest path is more than the sum of 1000 m and the distance between the two consecutive fixes. This restriction significantly increases the computational efficiency of the map-matching process without introducing deficiency in the correct link identification. This is significant for a real-time map-matching procedure. In addition, the novelty of the developed stMM algorithm lies in the use of a Genetic Algorithm (GA) in optimising the four weight parameters in identifying the correct segment from a set of candidate segments. This optimisation addressed the uncertainty and ambiguity inherent to matching GPS fixes within complex road configurations where multiple segments have the potential to be the correct segment (e.g. fly-overs and roundabouts).

The stMM algorithm was tested in urban, suburban and rural areas of Nottingham and Birmingham, UK. Four different frequency data (i.e. 1 Hz, 0.2 Hz, 0.033 Hz and 0.0167 Hz) were tested. It was found that the performance of the algorithm in correct link identification was 98.6%. The performance reduces to about 73% if neither information from the shortest path nor vehicle trajectory are used. This developed algorithm could be used to match heterogeneous GPS data from multiple sources (e.g. vehicles, smartphones) on a road map in real-time.

Acknowledgement

References

Bernstein, D., Kornhauser, A., 1996. An Introduction to Map Matching for Personal Navigation Assistants. <http://www.njtude.org/reports/mapmatchintro.pdf> (accessed 18.11.10).

Bernstein, D., Kornhauser, A., 1998. Map matching for personal navigation assistants. In: Proceedings of the 77th Annual Meeting of the Transportation Research Board, 11–15 January, Washington, DC.

Bouju, A., Stockus, A., Bertrand, F., Boursier, P., 2002. Location-based spatial data management in navigation systems. IEEE Symposium on Intelligent Vehicle, vol. 1, pp. 172–177.

Dechter, R., Pearl, J., 1985. Generalized best-first search strategies and the optimality of A*. J. ACM 32 (3), 505–536.

Fuchs, H., Kedem, Z.M., Naylor, B.F., 1980. On visible surface generation by a priori tree structures. Comput. Graph. 14, 124–133.

Greenfeld, J.S., 2002. Matching GPS observations to locations on a digital map. In: Proceedings of the 81st Annual Meeting of the Transportation Research Board, January, Washington, DC.

Jo, T., Haseyamai, M., Kitajima, H., 1996. A map matching method with the innovation of the Kalman filtering. IEICE Trans. Fund. Electron. Comm. Comput. Sci. E79-A, 1853–1855.

Kim, W., Jee, G., Lee, J., 2000. Efficient use of digital road map in various positioning for ITS. In: IEEE Symposium on Position Location and Navigation, San Deigo, CA.

Krakiwsky, E.J., Harris, C.B., Wong, R.V.C., 1988. A Kalman filter for intrgrating dead reckoning, map matching and GPS positioning. In: Proceedings of IEEE Position Location and Navigation Symposium, pp. 39–46.

Meng, Y., Chen, W., Chen, Y., Chao, J.C.H., 2003. A simplified map-matching algorithm for in-vehicle navigation unit. Research Report, Department of Land Surveying and Geoinformatics, Hong Kong Polytechnic University.

Najjar, M.E., Bonnifait, P., 2003. A roadmap matching method for precise vehicle Localization using belief theory and Kalman filtering. In: The 11th International Conference in Advanced Robotics, Coimbra, Portugal, June 30–July 3.

Ochieng, W.Y., Quddus, M.A., Noland, R.B., 2004. Integrated positioning algorithms for transport telematics applications. In: Proceedings of the Institute of Navigation (ION) Annual Conference, 20–24 September, California, USA.

Pyo, J., Shin, D., Sung, T., 2001. Development of a map matching method using the multiple hypothesis technique. In: IEEE Proceedings on Intelligent Transportation Systems, pp. 23–27.

Quddus, M.A., Ochieng, W.Y., Lin, Z., Noland, R.B., 2003. A general map matching algorithm for transport telematics applications. GPS Solutions 73, 157–167 (1st December 2003).

Quddus, M.A., 2006. High Integrity Map Matching Algorithms for Advanced Transport Telematics Applications. Ph.D. Thesis, Centre for Transport Studies, Imperial College London, UK.

Quddus, M.A., Ochieng, W.Y., Noland, R.B., 2006. A high accuracy fuzzy logic-based map matching algorithm for road transport. J. Intell. Transport. Syst. Technol. Plann. Operat. 10 (3), 103–115.

Scott, C.A., Drane, C.R., 1994. Increased accuracy of motor vehicle position estimation by utilizing map data, vehicle dynamics and other information sources. In: Proceedings of the Vehicle Navigation and Information Systems Conferences, pp. 585–590.

Scott, C.A., 1994. Improved GPS positioning for motor vehicles through map matching. In: Proceedings of the Institute of Navigation (ION) Annual Conference, 20–23 September, Salt Lake City, Utah.

Srinivasan, D., Cheu, R.L., 2003. Development of an improved ERP system using GPS and AI techniques. IEEE Proc. Intell. Transport. Syst., 554–559

Syed, S., Cannon, M.E., 2004. Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons. In: Proceedings of the Institute of Navigation (ION) National Technical Meeting, 26–28 January, California, USA.

Taylor, G., Blewitt, G., Steup, D., Corbett, S., Car, A., 2001. Road reduction filtering for GPS–GIS navigation. Transact. GIS 5 (3), 193–207 (ISSN 1361-1682).

Velaga, N.R., Quddus, M.A., Bristow, A.L., 2009. Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems. Transp. Res. Part C 17 (6), 672–683.

White, C.E., Bernstein, D., Kornhauser, A.L., 2000. Some map matching algorithms for personal navigation assistants. Transp. Res. Part C 8, 91–108.

Zhao, Y., 1997. Vehicle Location and Navigation System. Artech House Inc., MA.