Transportation Research Part C 17 (2009) 672–683

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems

Nagendra R. Velaga, Mohammed A. Quddus *, Abigail L. Bristow

Transport Studies Group, Department of Civil and Building Engineering, Loughborough University, Leicestershire LE11 3TU, United Kingdom

ARTICLE INFO

Article history:
Received 18 November 2008
Received in revised form 14 May 2009
Accepted 17 May 2009

Keywords:
Intelligent Transport System (ITS)
GPS, spatial road network
Optimisation
Topological map-matching

ABSTRACT

Map-matching (MM) algorithms integrate positioning data from a Global Positioning System (or a number of other positioning sensors) with a spatial road map with the aim of identifying the road segment on which a user (or a vehicle) is travelling and the location on that segment. Amongst the family of MM algorithms consisting of geometric, topological, probabilistic and advanced, topological MM (tMM) algorithms are relatively simple, easy and quick, enabling them to be implemented in real-time. Therefore, a tMM algorithm is used in many navigation devices manufactured by industry. However, existing tMM algorithms have a number of limitations which affect their performance relative to advanced MM algorithms. This paper demonstrates that it is possible by addressing these issues to significantly improve the performance of a tMM algorithm. This paper describes the development of an enhanced weight-based tMM algorithm in which the weights are determined from real-world field data using an optimisation technique. Two new weights for turn-restriction at junctions and link connectivity are introduced to improve the performance of matching, especially at junctions. A new procedure is developed for the initial map-matching process. Two consistency checks are introduced to minimise mismatches. The enhanced map-matching algorithm was tested using field data from dense urban areas and suburban areas. The algorithm identified 96.8% and 95.93% of the links correctly for positioning data collected in urban areas of central London and Washington, DC, respectively. In case of suburban area, in the west of London, the algorithm succeeded with 96.71% correct link identification with a horizontal accuracy of 9.81 m (2σ). This is superior to most existing topological MM algorithms and has the potential to support the navigation modules of many Intelligent Transport System (ITS) services.

1. Introduction

Map-matching (MM) techniques which integrate positioning data with spatial road network data have been developed in order to provide the real-time, accurate and reliable positioning information required by many ITS services such as route guidance, fleet management and accident and emergency response (Chen et al., 2003; Kim et al., 1996; Phuyal, 2002; Li and Chen, 2005; Li and Fu, 2003; Ochieng et al., 2004; White et al., 2000; Yin and Wolfson, 2004; Zhao et al., 2003). A range of MM techniques have emerged over the last decade categorised as geometric, topological, probabilistic and advanced. The earliest geometric MM (gMM) algorithms, developed in the 1990s, used geometric information, on the shape of the curve of the road segment (Kim et al., 1996; Quddus et al., 2007; White et al., 2000). These gMM algorithms are the simplest and fastest to implement as they require very little information, but they perform poorly especially when matching at junctions,

* Corresponding author. Tel.: +44 (0) 150 922 8545; fax: +44 (0) 1509 223981.
E-mail address: m.a.quddus@lboro.ac.uk (M.A. Quddus).

0968-090X/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.trc.2009.05.008

complex roundabouts and parallel roads. gMM algorithms may be improved by including historical data (such as the previously matched road segment), vehicle speed and topological information on the spatial road network (such as link connectivity). A MM algorithm that uses such additional information is called a topological MM (tMM) algorithm (Greenfeld, 2002; Li et al., 2005; Marchal et al., 2005; Quddus et al., 2003, 2007). Probabilistic MM (pMM) algorithms use probability theory to identify the set of candidate segments by taking into account the error sources associated with both navigation sensors and spatial road data. The MM algorithms classed as advanced MM (aMM) algorithms include applications of extended Kalman filter (EKF), belief theory, fuzzy logic (FL) and artificial neural network (ANN) techniques (Pyo et al., 2001; Quddus et al., 2006; Syed and Cannon, 2004; Yang et al., 2003). An aMM algorithm that uses these more refined approaches, outperforms other MM algorithms but requires more input data and is relatively slow and difficult to implement. Whereas a tMM algorithm is very fast, simple and easy to implement. For this reason, a tMM algorithm has more potential to be implemented in real-time applications by industry as its processor requires less memory. Once the limitations of existing tMM algorithms are addressed, the performance of a tMM algorithm is expected to be comparable to that of pMM or aMM algorithms.

Therefore, the main aim of this paper is to report the development of an enhanced tMM algorithm and to assess its performance using real-world field data. This process includes:

(1) the derivation of four weights (including two new weights) through an optimisation process,
(2) the introduction of two consistency checks to minimise mismatches at ambiguous situations,
(3) the development of a new procedure for the initial map-matching process to improve the overall performance of the algorithm.

The paper is organised as follows: The next section provides a discussion on the performance of existing tMM algorithms and identifies their limitations. This is followed by a description of an enhanced tMM algorithm: including the initial map-matching process, the optimisation technique and the consistency checks. The data collection process is then presented, followed by the results. The paper ends with conclusions and future research directions.

2. Performance of topological map-matching algorithms

In this paper, only the performance of topological map-matching (tMM) algorithms is considered. Readers are referred to Quddus et al. (2007) for a detailed review of MM algorithms. A tMM algorithm makes use of historical information, which might include the previously identified road segment and topological information such as link connectivity, road classification, turn restriction information, in addition to the basic geometric information. Different studies have used topological information at different levels. For example, using topological information to identify a set of candidate links or to check the map-matched positioning after geometric MM or in the process of correct link identification from a set of candidate links (Li and Fu, 2003; White et al., 2000). It has been established that the use of topological information in correct link identification can improve map-matching performance. Moreover, a weighting approach in selecting the correct road segment from the candidate segments improves the accuracy of correct road segment identification (Greenfeld, 2002; Quddus et al., 2003). An algorithm that assigns weights for all candidate links – using similarity in network geometry and topology information and positioning information from a GPS/DR integrated system – and selects the link with the highest weight score as the correct road segment is called a weight-based tMM algorithm.

Few studies report on the performance of tMM algorithms. Those that have done so are shown in Table 1. Most did not assess algorithm performance with respect to 2-D horizontal accuracy due to a lack of higher accuracy reference (true) positioning trajectory. Quddus (2006) tested four of these algorithms using suburban data (2040 positioning fixes) obtained from GPS/DR and a digital map of scale 1:2500. Carrier-phase GPS observations were used to obtain the reference (true) trajectory. These results are shown in the last column of Table 1. This positioning data, used by Quddus (2006), is also used in this research to test the performance of the enhanced algorithm to enable a real test of comparative performance to be made.

Table 1 suggests that the performance of tMM algorithms with respect to correct link identification ranges from 85% to 98.5%; and the horizontal accuracy ranges from 32 m (2σ) to 18.1 m (2σ). Although, the MM algorithm developed by Srinivasan et al. (2003) identified 98.5% of the segments correctly, this was based on a small sample in a simple network. When tested on a larger, more representative, road network, the accuracy falls to 80.2%. The algorithm developed by Blazquez and Vonderohe (2005) is capable of identifying the correct road segment 94.8% of times while employing a sample size of 600 position fixes obtained from a DGPS. Their algorithm performance is reasonably good and this may be due to their use of a high accuracy DGPS (relative to a stand-alone GPS) to obtain position fixes and they also consider link connectivity and turn restriction information to verify map-matched positions after a point-to-curve map-matching approach.

Table 1 suggests that when tested on the same data set, weight-based algorithms perform better than non-weight-based algorithms. However, their performance is not sufficient to support many ITS services. Ways in which existing tMM algorithms may be improved include:

(1) The subsequent MM process of a weight-based tMM algorithm is heavily dependent on the performance of the initial MM process. Therefore, a more robust and reliable procedure for the initial MM process, should reduce mismatches.

Table 1
Performance of existing topological MM Algorithms.

Authors and Date	Navigation sensors	Test environment	Map scale	Sample size	Topological information used	% Correct link identification and horizontal accuracy (m) by authors	% Correct link identification and horizontal accuracy (m) by Quddus (2006)
Non-weight-based algorithms							
White et al. (2000)	GPS	Suburban	–	1.2 Km	Heading, proximity and link connectivity	85.80%	76.8% 32 m (95%)
Srinivasan et al. (2003)	GPS	University road network	–	242 GPS points	Heading and turn restriction	98.5%	80.2% 21.2 m (95%)
Blazquez and Vonderohe (2005)	DGPS	Urban and Suburban	1:2400	600 DGPS points	Link Connectivity and turn restrictions	94.8%	–
Weight-based approaches							
Greenfeld (2002)	GPS	Urban and Suburban	–	–	Heading, proximity and intersection weights	–	85.6% 18.3 m (95%)
Quddus et al. (2003)	GPS and DR	Urban	1:1250	–	Heading, proximity and position of point relative to link	88.6% 18.1 m (95%)	88.6% 18.1 m (95%)

(2) Weight-based algorithms primarily consider heading and proximity weights. These may be enhanced by including the performance of weights for turn-restriction at junctions, link connectively, roadway classification (e.g., one-way or two-way roads) and road infrastructure information (e.g., fly-overs and underpasses). The relatively good performance of the tMM algorithm developed by Blazquez and Vonderohe (2005) that used turn restriction and link connectivity would seem to support this.
(3) The relative importance of different weights may be derived using a robust method rather than assuming equal weights as Greenfeld (2002) did or deriving them empirically as Quddus et al. (2003) did. This can be done for different combinations of navigation sensors (such as GPS or GPS/DR or DGPS) by collecting data from different operational environments (such as dense urban, urban, suburban, rural and hilly areas). This will improve the transferability of the developed weighting scheme. Another approach would be to determine different weighting schemes for different operational environments. For instance, the weight for heading may be more important in a dense urban environment than in a rural context.

3. Description of the enhanced topological map-matching algorithm

The method by which the enhanced tMM algorithm was developed is outlined here, including the map-matching process, the optimisation of weights and consistency checks.

3.1. Input data

The data required for the improved tMM algorithm are: link data including a unique link ID, start node and end node; node data including unique node ID, easting and northing coordinates of the node; positioning and navigation data from a navigation sensor (either GPS or GPS/DR) including easting and northing coordinates of position fixes, vehicle heading, vehicle speed in m/s; and turn restriction data for junctions. The turn restriction information is stored in the form of a turn restriction matrix to consider all the possible turns at a junction point.

3.2. Map-matching process

A simple flowchart of the proposed tMM algorithm is shown in Fig. 1. The map-matching (MM) process is divided into three key stages: (a) initial MM, (b) MM on a link and (c) MM at a junction. The aim of the initial MM process is to identify the first correct link for the first positioning point. A robust and reliable method (discussed below) is introduced for the initial MM process. After assigning the first positioning fix to the correct link, the algorithm checks three criteria for matching the subsequent position fix:

(1) whether a vehicle is in a stationary condition (*matching on a link*),
(2) whether a vehicle is travelling on the previously matched link (*matching on a link*),
(3) whether a vehicle is near to a junction (*matching at a junction*).

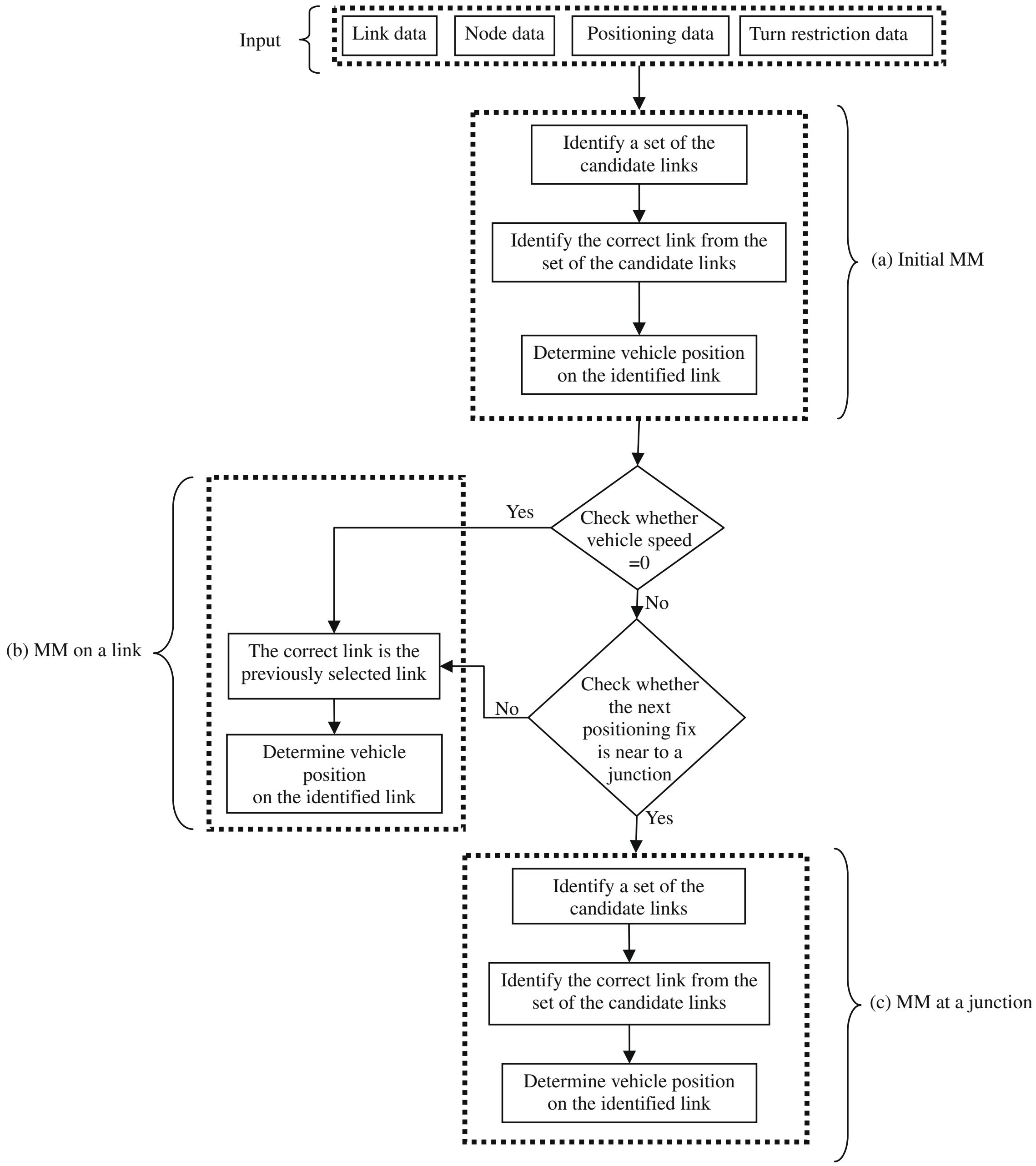

Fig. 1. A flow-chart representing the enhanced tMM algorithm.

If the speed of the vehicle for a positioning fix is zero then the vehicle is stationary; in this case the vehicles position is assigned to the previously map-matched link. If the vehicle is not stationary, then the algorithm examines whether the positioning fix is near to the downstream junction or not. If the vehicle is some distance from the junction then this positioning fix is also assigned to the previously map-matched road segment. On the other hand, if the vehicle is near to a junction, the algorithm re-identifies the correct road segment from a set of candidate segments which is known as *matching at a junction*. The above three criteria are further described in the following sections.

In all cases, once the correct link is identified for a positioning fix, a perpendicular projection from the positioning fix to the link gives the location of the vehicle on that link.

3.2.1. Initial map-matching

The purpose of the initial MM process is to identify the first correct road segment for the first positioning point. After the initial MM process, the subsequent matching (either on a link or at a junction) may commence. Since any error in the initial matching process will lead to a mismatching of the subsequent positioning points, a robust and reliable approach is introduced which has three major stages:

(1) the identification of a set of candidate links,
(2) the identification of the correct link among the candidate links using heading weight (W_h) and proximity weight (W_p),
(3) the estimation of vehicle position on the correct link.

Firstly, the algorithm creates an error bubble around the first positioning fix. The radius of the error bubble is primarily based on quality of positioning data (i.e. variance and covariance of easting and northing) at that instant (for that positioning point). The error bubble used in this research was suggested by Zhao (1997). The same concept was used by Ochieng et al. (2004) and Quddus (2006). All the links that are either inside the error bubble or crossing the error bubble or tangent to the error bubble are considered as the candidate links for the first positioning fix. In previous studies of MM algorithms only the links that had a node (either its starting node or its end node) within the error bubble were considered, and this could lead to a potential mismatch (Quddus et al., 2007). The approach introduced here should eliminate the possibility of such a mismatch. Then the task is to select the correct link among these candidate links. For the first positioning fix, topological information (link connectivity and turn restrictions) is not available. Therefore, only heading and proximity weights are considered. A GPS receiver provides heading data for the first positioning fix based on the last stored position fix.

Among the candidate links, clearly higher weight should be given to a link that is in-line with the vehicle's direction of movement. Therefore, the heading weight is considered as a cosine function of angle between the vehicle movement direction and link direction (as suggested by Greenfeld, 2002) and shown in Eq. (1).

$$W_h = H_w f(\theta) \tag{1}$$

where $f(\theta) = \cos(\theta)$, W_h denotes the weight for heading, H_w denotes the weight coefficient for heading. The value of H_w will be determined by an optimisation technique presented in Section 3.3. θ denotes the angle difference between the vehicle heading and link direction with respect to the north.

The weight for proximity is based on the perpendicular distance (D) from the positioning point to the link. If a link is nearer to the positioning point, then this link should be given more weight than a link which is further away. If the perpendicular line from the positioning fix to the link does not physically intersect then D is increased by ΔD which represents the distance between the intersection point and the closest node of the link. The weight for proximity (W_p) varies linearly with distance.

$$W_p = D_w f(D) \tag{2}$$

where $f(D)$ is a function of distance that contributes to the relative importance of proximity in identifying the correct link at a junction and can be given by:

$$f(D) = \left[\frac{(80 - D)}{80}\right]$$

W_p denotes the weight for proximity, D_w denotes the weight coefficient for proximity. The value of D_w will be determined by an optimisation technique presented in Section 3.3.

If a positioning fix falls on the link (i.e. $D = 0$) then $f(D)$ in Eq. (2) is 1; and if the distance between a positioning point and the link is more than or equal to 160 m (i.e. $D \geqslant 160$ m), $f(D)$ is -1. Between 0 and 160 m (that is $0 < D < 160$ m), $f(D)$ decreases linearly with the distance. This is because an empirical investigation suggests that if D is higher than 160 m then the algorithm has wrongly identified the link.

The total weight score (*TWS*) for the first positioning point is the sum of both heading and proximity weights.

$$TWS = W_h + W_p \tag{3}$$

TWS is calculated for each candidate link and the link with the highest *TWS* is identified as the correct link. The vehicle location on that selected link is then estimated. This is achieved by a perpendicular projection of the positioning point onto the link.

3.2.2. Map-matching on a link

After successful completion of the initial MM process, the second stage of the tMM algorithm starts which is the MM on a link. The algorithm checks the speed of the vehicle. If the vehicle speed is zero, the algorithm assigns the vehicle to the previously map-matched road segment. If the vehicle is moving (i.e. speed is greater than zero), the algorithm checks whether the vehicle is near a junction using two criteria:

(1) distance from the previously map-matched vehicle position to the downstream junction
(2) the vehicle heading with respect to the previously matched link direction.

For the first check, to examine whether the vehicle is near to a junction or not, the algorithm compares the remaining distance on the previously map-matched road segment with the distance travelled by the vehicle within the last time interval, i.e.,

$$d_1 \geq (d_2 + d_{threshold}) \tag{4}$$

where d_1 is the distance between the previously map-matched positioning point to the downstream junction, d_2 is the distance travelled by the vehicle during last time interval.

If $d_1 \cong d_2$, it is considered that, for the current positioning fix, the vehicle is at a junction. However, due to errors associated with the previous map-matched positioning, errors with digital map, and ignorance of road width, a distance threshold ($d_{threshold}$) needs to be considered. This is to ensure that the map-matching process does not miss vehicles that may be at a junction. Here, $d_{threshold}$ is considered as a positive value.

For the second check, if the vehicle direction changes significantly with respect to the previously selected road link, it is considered to have turned. The mathematical representation of this check is shown in Eq. (5).

$$h_{RMS} \geq (\delta_i + h_{threshold}) \tag{5}$$

where h_{RMS} denotes the Root Mean Square (RMS) error value of all headings related to the positioning fixes mapped on the previously identified link. δ_i is the absolute value of angle difference between the vehicle heading at the current position fix and the previously identified link direction.

GPS position fixes are less reliable when speed of the vehicle is less than 3 m/s (Quddus et al., 2007; Taylor et al., 2001). To overcome this, a bearing threshold ($h_{threshold}$) is added to δ_i.

The distance threshold ($d_{threshold}$) and the bearing threshold ($h_{threshold}$) values were derived empirically from an independent field data set of 1800 GPS/DR fixes, and identified as 20 m and 5°, respectively. However, these threshold values depend on the quality and scale of digital map, sampling frequency, and the quality of navigation output from a positioning sensor (either GPS or GPS/DR).

If the two checks are satisfied then the algorithm assumes that the vehicle is moving on the previously matched link, and the algorithm then snaps the current positioning fix to the previously selected road segment.

3.2.3. Map-matching at a junction

When the vehicle is at a junction, a road segment is identified among the set of candidate segments. The procedure for the identification of the set of candidate segments for a positioning point at a junction is the same as that of the initial MM process. The correct link is selected based on the total weight score (*TWS*). At this stage, two additional weights are introduced on turn restrictions at junctions and link connectivity. If a vehicle approaches a junction and is not legally permitted to turn (either a left-turn, a right-turn or a U-turn) on to a link connected to the junction, then the link is given less weight relative to the other links on to which the vehicle can turn. With respect to link connectivity, a link is given more weight if it is directly connected to the previously identified link for the previous epoch. The link connectivity weight (W_c) and turn restriction weight (W_t) are given below:

$$W_c = C_w X \tag{6}$$

$$W_t = T_w Y \tag{7}$$

where X is a variable that contributes to the relative importance of link connectivity in identifying the correct link at a junction. This variable takes only two values 1 and −1, i.e.,

$$X = \{1, -1\}$$

Y is a variable that contributes to the relative importance of turn restriction in identifying the correct link at a junction. This variable takes only two values 1 and −1, i.e.,

$$Y = \{1, -1\}$$

C_w and T_w are the weight coefficients for link connectivity and turn restriction respectively. The values of C_w and T_w will be determined by an optimisation technique presented in Section 3.3.

X equals 1 if a candidate link (within the set of the candidate links) is directly connected to the previously identified link and −1 otherwise. Y equals 1 if a vehicle can legally make a turn to a link and −1 otherwise.

The *TWS* for a link at a junction is the sum of four weights as given below:

$$TWS = H_w \cos(\theta) + D_w \left(\frac{80 - D}{80}\right) + C_w X + T_w Y \tag{8}$$

The H_w, D_w, C_w and T_w are the weight coefficients for heading, proximity, link connectivity and turn restriction respectively. These coefficients represent the relative importance of different factors in calculating the *TWS*.

The functions representing heading, $f(\theta)$, proximity, $f(D)$, connectivity, X and turn restrictions, Y are specified such that their values lie between +1 to −1 for any possible values of the factors. This constraint allows some control over the relative importance of weight coefficients. Although values of θ, D, X and Y in Eq. (8) are available for a positioning fix, the values of the coefficients H_w, D_w, C_w and T_w are unknown. In previous research, these values were assumed to be equal (Greenfeld, 2002) or determined empirically (Quddus et al., 2003). This raises the issue of transferability to different operational environments.

In this research, an optimisation technique is developed to determine the values of H_w, D_w, C_w and T_w. The aim is to identify the values of these weight coefficients that minimise the total map-matching error in terms of identification of the correct links.

3.3. Optimisation of the weights

The values of the four weight coefficients (H_w, D_w, C_w and T_w) can be estimated from a series of controlled experiments in a post-processing way (i.e., an off-line mode). For a specific data set, the optimisation process starts with the map-matching of a positioning fix near to a junction and generates random values for the four coefficients between 1 and 100 in such a way that the sum of all coefficients equals 100. Using these selected values, the process then calculates the *TWS* (see Eq. (8)) for all links connected to that junction and identifies the correct link based on the highest *TWS* value. Since the actual link is known, it is possible to see whether the MM algorithm has identified the link correctly for the selected values of the four coefficients. If the algorithm fails to identify the correct link among the candidate links then the algorithm regenerates another set of random values of the coefficients and repeats the map-matching at that junction. This process continues until the algorithm selects the correct link. This produces a set of weight coefficients (H_w, D_w, C_w and T_w) that identify the link correctly at that junction. These weights are then applied to all positioning fixes. The percentage of wrong link identification is then calculated for these specific values of the coefficients. The same procedure is repeated for all positioning fixes near to junctions. This process generates a set of values for the weight coefficients and the corresponding percentage of error associated with the wrong link identification for each set. As the other variables, $f(\theta)$, $f(D)$, X and Y, in Eq. (8) vary from 1 to -1 for any possible values, it is assumed that the map-matching error with respect to the correct link identification (MM_{error}) is a function of the weights H_w, D_w, C_w and T_w only, i.e.,

$$MM_{error} = f(H_w, D_w, C_w, T_w) \tag{9}$$

This simulated data is then used to develop a relationship between percentage of wrong link identification and the weight coefficients (H_w, D_w, C_w and T_w) using a regression analysis. Since the error associated with wrong link identification is always a positive value, a log-linear model is used. The functional relationship between the weights and the MM error is unknown and therefore, various specifications are considered. Assuming that the map-matching error (MM_{error}) depends on the individual weights (H_w, D_w, C_w and T_w), their square terms (H_w^2, D_w^2, C_w^2 and T_w^2), inverse terms ($1/H_w$, $1/D_w$, $1/C_w$ and $1/T_w$) and interaction terms (H_wD_w, H_wC_w, H_wT_w, D_wC_w, D_wT_w and C_wT_w), a functional relationship can be written as:

$$\ln(MM_{error}) = \alpha + [\beta_{h1}H_w + \cdots + \beta_{t1}T_w] + [\beta_{h2}H_w^2 + \cdots + \beta_{t2}T_w^2] + \left[\frac{\beta_{h3}}{H_w} + \cdots + \frac{\beta_{t3}}{T_w}\right] + [\beta_{hd}(H_wD_w) + \cdots + \beta_{ct}(C_wT_w)] + \varepsilon_i \tag{10}$$

where α is an intercept term. $\beta_{h1}, \beta_{h2}, \ldots, \beta_{t1}, \beta_{t2}, \beta_{t3}$ are the regression coefficients to be estimated. ε_i is the error term assumed to be independently and identically distributed with zero mean and a constant variance.

Initially, all 18 variables are considered in the regression analysis. The regression analysis is carried out using a step-by-step backward elimination process, at each step one statistically insignificant parameter based on the corresponding t-stat, is removed. Variables with a t-stat of 1.96 or higher are retained (considering a 95% confidence interval). The final regression model with all statistically significant variables is the optimisation function.

The objective is to minimise the error. In order to perform this minimisation, some restrictions have to be imposed. As discussed, the sum of all weight coefficients is set to be 100 and the minimum and maximum values of each weight coefficient set at 1 and 100, respectively. The optimisation function, obtained from above regression analysis, and the associated constraints is given below.

$$\begin{aligned} \text{Minimisation} \quad MM_{error} = \exp\Big\{&\alpha + \left[\hat{\beta}_{h1}H_w + \ldots + \hat{\beta}_{t1}T_w\right] + \left[\hat{\beta}_{h2}H_w^2 + \ldots + \hat{\beta}_{t2}T_w^2\right] \\ &+ \left[\frac{\hat{\beta}_{h3}}{H_w} + \ldots + \frac{\hat{\beta}_{t3}}{T_w}\right] + \left[\hat{\beta}_{hd}(H_wD_w) + \ldots + \hat{\beta}_{ct}(C_wT_w)\right]\Big\} \end{aligned} \tag{11}$$

$$\text{subject to}: \quad \begin{aligned} &H_w + D_w + C_w + T_w = 100 \\ &1 \leqslant \{H_w, D_w, C_w, T_w\} \leqslant 100 \end{aligned}$$

Optimisation of Eq. (11) was carried out in MATLAB using the constrained nonlinear minimization method (Michael et al., 2007). The values of four weight coefficients (H_w, D_w, C_w and T_w) were calculated by identifying the global minimum of map-matching error (MM_{error}). It is a convex optimisation problem. The process was applied to real-world positioning data obtained from different operational environments including: dense urban, suburban and rural areas.

3.4. Consistency checks to minimise mismatches

Two consistency checks are carried out before finalising the selection of the correct link among the candidate links. These are:

(a) whether the *TWS* for two or more links are close to each other and
(b) whether the distance between the raw position fix and the map-matched position on the link is large.

For the first check, if the *TWS* for two (or more) links are found to be within 1% then the algorithm identifies this as an ambiguous situation. This is because an investigation of our data suggests that 1% difference in the *TWS* values correctly picks all ambiguous situations. The algorithm then uses some external information such as the distance from the last map-matched position to the current map-matched position and compares this with the distance (speed $\times$ time) travelled by the vehicle within the last time interval. If these two distances agree for a particular link then it is assumed that this is the current link.

After matching a positioning fix to the identified link, the second consistency check estimates the distance from the positioning fix to the map-matched location on the link. If the distance exceeds the pre-defined threshold, then it is assumed that the identified link is not the correct link. In such a case, the algorithm carries out the first check (i.e., comparing distance between the previously matched point to the current map-matched position with distance travelled by vehicle within the last time interval, which is one second in our case), and identifies the road segment on which vehicle is travelling. The pre-defined threshold is based on the error ellipse, the quality of spatial road data and sampling frequency of positioning data. From an analysis of an independent data set consisting of 1800 observations, a fixed threshold, 40 m, has been selected.

4. Positioning data

Six data sets, obtained from different GPS/DR equipment in different time periods, and in various operational environments – dense urban, suburban and rural areas – were used in this research (see Table 2). All these GPS/DR data sets provided positioning data every second. A 1:2500 scale digital map was used in the map-matching process. Data set 1, 2 and 6 were obtained from Quddus (2006), whilst data sets 3, 4 and 5 were collected as part of this study. The actual links on which the vehicle was driven were known for all data sets.

4.1. Data for optimisation

Data sets 1, 2 and 3 collected in urban, suburban and rural areas, respectively, were used for the optimisation process.

4.2. Data for algorithm performance checking

In order to evaluate the performance of the enhanced tMM algorithm, data sets 4, 5 and 6 were used. For the collection of the fourth data set, a test vehicle equipped with a single frequency high sensitivity GPS receiver and a low-cost gyroscope were used. The test vehicle travelled on a pre-defined route in central London on the 26th May 2008. Data set 5 was collected on a pre-planned route from the downtown area of Washington, DC, USA using a 16-channel single frequency high sensitivity GPS receiver, on the 13th January 2009. For both data sets (4 and 5) the test route was selected carefully to ensure that the vehicle travelled through a good mix of urban characteristics. The total trip length of data sets 4 and 5 were about 18 km and 17 km, respectively. The test trajectory for data set 4 (in central London) and data set 5 (in Washington, DC) are shown in Figs. 2 and 3, respectively. But, no reference (actual) trajectory in terms of true vehicle positions was available for the data sets and therefore, the algorithm's performance can be tested only with respect to correct link identification. However, the reference trajectory of the vehicle was available for data set 6 obtained from Quddus (2006). This allows the performance to be assessed in terms of both link identification and horizontal accuracy. It should be noted that the data set 6 was also used to examine the existing tMM algorithms performance by Quddus (2006) and the results are directly comparable.

Table 2
Positioning data sets.

Data set	Operational environment	Data collection date (month and year)	Equipment used	Sample size: data points	Location characteristics
1	*Urban*: (central London)	Jun-02	GPS/DR	1280	Urban characteristics such as tall buildings, bridges, flyovers, and dense road network
2	*Suburban*: (South London)	Nov-05	GPS/DR	1812	Suburban area
3	*Rural*: (Loughborough, East Midlands, UK)	Mar-08	GPS	1200	University roads, and other rural roads
4	*Urban*: (central London)	May-08	GPS	2814	Dense urban road network, tall buildings, bridges, flyovers, tunnels
5	*Urban*: (Washington, DC, USA)	Jan-09	GPS	3600	Urban characteristics such as bridges, flyovers
6*	*Suburban*: (West of London – near Reading)	Aug-05	GPS/DR and carrier-phase GPS	2040	Suburban area

* Data set 6 was also used to examine existing tMM algorithms performance by Quddus (2006) (as shown in Table 1).

5. Optimisation results

Table 3 shows the best fitting regression models; the adjusted R^2 estimates the percentage of behaviour of dependent variable (i.e., percentage of wrong link identification) is explained by the independent variables. As mentioned, the sum of the four weights is 100. If an intercept (i.e., α) term is considered in the regression, this term is then directly correlated with weight coefficients and subsequently, one of the weight coefficients is automatically dropped from the regression model. However, the inclusion of individual weights is important as our objective is to find the relative importance of these four weight coefficients (H_w, D_w, C_w and T_w) in reducing the error in map-matching process. Therefore, the regression does not

Fig. 2. Test route in central London.

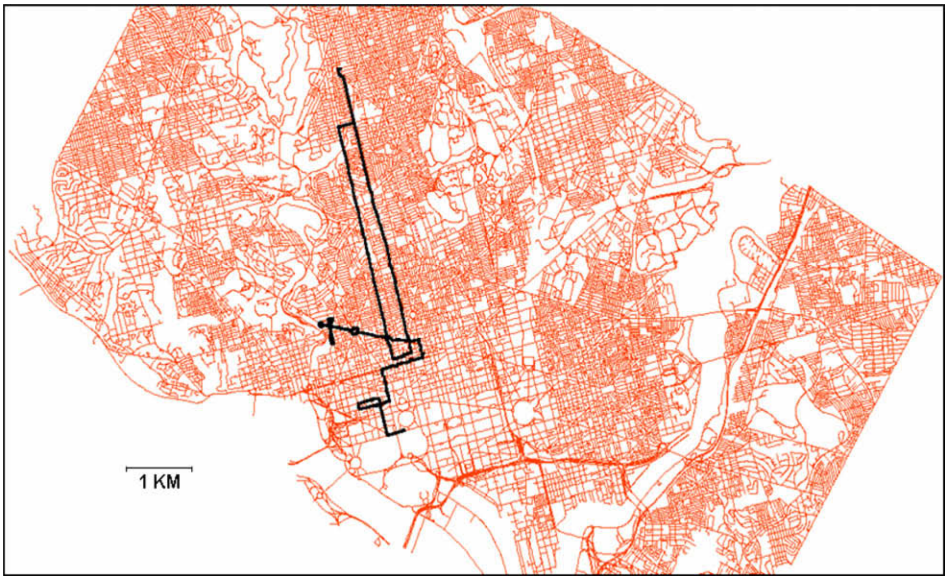

Fig. 3. Test route in Washington, DC, USA.

have an intercept term 'α'. As the regression is forced through the origin, the adjusted R^2 is high in all the cases. The model specifications vary by operational environment suggesting that the use of one specification for all environments can be misleading.

The values of regression coefficients presented in Table 3 are then put in Eq. (11) to obtain three optimisation functions representing three different operational environments. Subsequently, the optimisation process discussed in Section 3.3 is applied to each of the three functions and the optimal values of the four weight coefficients for each of the three operational environments are illustrated in Table 4. As speculated earlier, the relative importance of the four factors (heading, proximity, connectivity and turn restriction) considered in identifying the correct link varies by operational environment. This suggests that a map-matching algorithm that uses the same weight coefficients for all operational environments may create incorrect results. It is interesting to note that all existing weight-based topological map-matching algorithms assume a single set of weight coefficients for all environments.

As shown in Table 4, the two new weights (i.e., weights for connectivity and turn restriction) considered in this study are found to be contribute to about 52% of the *TWS* for the case of urban data. This is one of the significant findings of this research as this indicates that the consideration of link connectivity and turn-restriction at a junction is more important for an urban road network compared to a suburban or a rural network. This is realistic as roads in an urban area are in close proximity and the quality of positioning data are bad (due to the effect of multipath) compared to an open area. In suburban and rural areas the weight for connectivity (C_w) and weight for turn restriction (T_w) are not so important as the quality of GPS/DR positioning fixes is good and the road network is less dense.

6. Algorithm performance

The enhanced algorithm developed in this research first identifies the operational environment in which the vehicle is travelling, and then selects the corresponding weights from the weight matrix (Table 4) suitable for that environment. The identification of an operational environment should be based on the density of road network, surrounding land-use information (i.e., residential, commercial, open spaces, etc.) and building height data from a digital terrain model (DTM). Here, only the density of road network is used as other data are not available. In urban areas, roads are in closer proximity to each other compared to suburban or rural areas and therefore, the total length of all roads for a given area in an urban environment will be greater than that of a suburban environment and similarly, the total length of all roads for a given area within a suburban environment will be greater than that of a rural area.

Table 3
Regression models for urban, suburban and rural area.

Weights	Urban		Suburban		Rural	
	Coefficient	*t*-Stat	Coefficient	*t*-Stat	Coefficient	*t*-Stat
H_w	0.0231	8.35	0.0287	16.82	0.0285	6.48
D_w	0.0266	8.81	0.0233	17.99	0.0235	9.51
C_w	0.0352	4.78	0.00347	4.97	0.0311	14.83
T_w	0.0132	4.88	0.00467	6.72	0.0302	19.2
H_w^2	–	–	−0.000115	−5.34	–	–
D_w^2	–	–	−0.0000476	−2.56	–	–
$1/(H_w)$	2.542	9.44	1.266	34.3	–	–
$1/(D_w)$	0.551	2.55	1.137	25.36	–	–
$1/(C_w)$	0.957	4.47	0.197	6.12	–	–
$1/(T_w)$	–	–	0.260	5.94	–	–
$(H_w * D_w)$	–	–	−0.000539	−18.16	−0.00056	−3.62
$(H_w * C_w)$	−0.00064	−4.14	–	–	–	–
$(H_w * T_w)$	–	–	−0.000069	−2.97	–	–
$(D_w * C_w)$	−0.000552	−2.99	–		–	–
$(C_w * T_w)$	−0.000406	−2.29	0.00013	5.91	–	–
Adjusted R^2	0.984		0.997		0.997	
Observations	175		450		40	

Where H_w = heading weight coefficient; D_w = proximity weight coefficient; C_w = connectivity weight coefficient; and T_w = turn restriction weight coefficient.

Table 4
Optimisation result.

Weights	Operational areas		
	Urban	Suburban	Rural
H_w	39.99	46.24	44.48
D_w	8.13	44.99	53.52
C_w	36.40	4.46	1
T_w	15.48	4.31	1

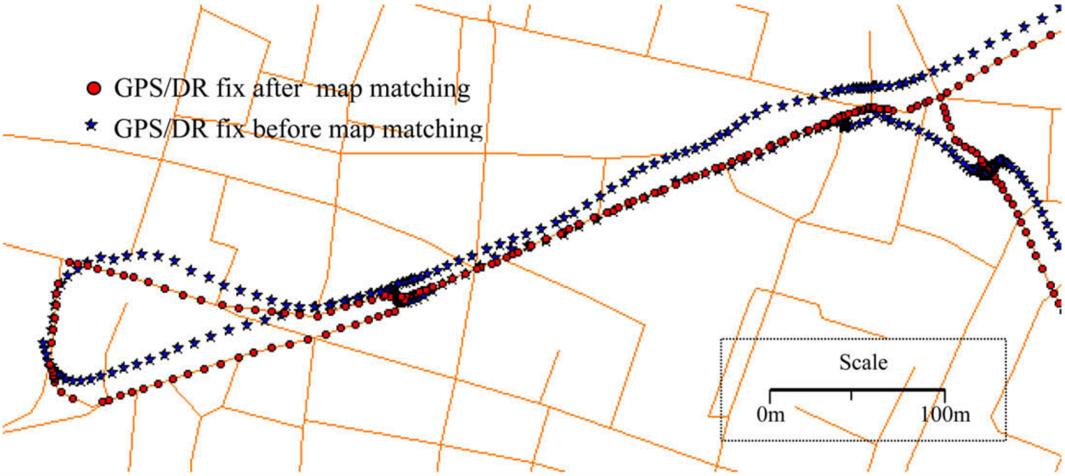

Fig. 4. A part of test road with map-matched positions.

The values of the weights from Table 4 are then applied for algorithm testing. For data set 4 (urban areas in central London) the enhanced tMM algorithm identified 96.8% of the road segments correctly. In case of data set 5 (urban areas in Washington, DC) the success rate is 95.93%. The test result, with data sets from two metropolitan cities, suggests that the enhanced algorithm is transferable to a certain extent. In terms of computational speed, the algorithm carried out the map-matching of 180 positioning fixes per sec (with a laptop of 1 GB RAM and 1.46 processor speed). This suggests that the algorithm is suitable for real-time implementation. Fig. 4 shows a part of the test trajectory (from data set 4) along with raw positioning fixes (with star symbol) and map-matched fixes (with round symbols).

The enhanced tMM algorithm was then applied to the sixth data set (suburban area), which was also used for performance evaluation in terms of both the correct link identification and 2-D horizontal accuracy of existing MM algorithms by Quddus (2006). The algorithm identified 96.71% of the road segments correctly with a horizontal (2D) accuracy of 9.81 m (2σ). The along-track and cross-track errors were found to be 7.36 m (2σ) and 9 m (2σ), respectively. This outperforms all the tMM algorithms previously tested by Quddus (2006) using the same data set.

Again, based on the same positioning data, a fuzzy logic-based MM algorithm developed by Quddus (2006) was capable of identifying 99.2% of the links correctly with a horizontal positioning accuracy of 5.5 m (2σ) for suburban road network. These results indicate that the performance of the enhanced tMM algorithm developed here approaches that of an aMM algorithm. This enhanced tMM algorithm has the potential to support a range of ITS services.

7. Conclusions and future research

In this paper, a real-time, weight-based topological MM algorithm has been developed by addressing some of the limitations of existing topological MM algorithms. The algorithm has been tested using real-world field data collected in different operational environments. The key features of the enhanced topological MM algorithm are:

(a) the selection of candidate links in the initial map-matching process and the map-matching at junctions,
(b) the introduction of two additional weight parameters, connectivity and turn restriction,
(c) use of an optimisation process to derive the relative importance of weights using data collected in different operational environments and
(d) the implementation of two consistency checks to reduce mismatches.

These new features have all contributed to the improve performance of the algorithm. The two new weights (i.e., link connectivity and turn restriction weights) have contributed significantly in the identification of the correct link at a junction for the case of urban data. The enhanced topological MM algorithm identified 96.8% of the road segments correctly for data set from central London; and 95.93% correct road segments for positioning data from urban areas in Washington, DC; and 96.71% of the road segments with a horizontal accuracy of 9.81 m (2σ) in a suburban area. The optimal weights for different factors such as heading, proximity, connectivity and turn-restriction may be transferable as these values were estimated from a range of data sets collected from various road environments. This requires further testing.

This algorithm performs better than most existing topological MM algorithms reported in the literature and its performance is comparable with that of advanced MM algorithms. This topological MM algorithm is fast, simple and very efficient and therefore, has the good potential to be implemented by industry. This algorithm is capable of supporting navigation modules of many location-based ITS services operating in urban areas.

Further research will investigate the optimisation of weights using more positioning data from each operational environment to ensure that the values are transferable. The performance of a tMM algorithm can further be improved by investigating the causes of the mismatches and modifying the algorithm accordingly.

References

Blazquez, C.A., Vonderohe, A.P., 2005. Simple map-matching algorithm applied to intelligent winter maintenance vehicle data. Transportation Research Records 1935, 68–76.

Chen, W., Yu, M., Li, Z.L., Chen, Y.Q., 2003. Integrated vehicle navigation system for urban applications. In: Proceedings of International Conference on Global Navigation Satellite Systems (GNSS), Graz, Austria, April 22–24, pp. 15–22.

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

Kim, J.S., Lee, J.H., Kang, T.H., Lee, W.Y., Kim, Y.G., 1996. Node based map-matching algorithm for car navigation system. In: Proceedings of 29th International Symposium on Automotive Technology and Automation (ISATA), Florence, Italy, vol. 10, pp. 121–126.

Li, Z., Chen, W., 2005. A new approach to map-matching and parameter correcting for vehicle navigation system in the area of shadow of GPS signal. In: Proceedings of IEEE Conference on Intelligent Transportation Systems, pp. 425–430.

Li, J., Fu, M., 2003. Research on route planning and map-matching in vehicle GPS/dead-reckoning/electronic map integrated navigation system. In: Proceedings of IEEE Conference on Intelligent Transportation Systems, pp. 1639–1643.

Li, X., Lin, H., Zhao, Y., 2005. A connectivity based map-matching algorithm. Asian Journal of Geoinformatics 5 (3), 69–76.

Marchal, F., Hackney, J., Axhausen, W., 2005. Efficient map matching of large Global Positioning System data sets tests on speed-monitoring experiment in Zurich. Transportation Research Records 1935, 93–100.

Michael, C.F., Mangasarian, O.L., Wright, S.J., 2007. Linear Programming with MATLAB. MPS-SIAM Series on Optimization, Inc., Philadelphia. ISBN: 9780898716436.

Ochieng, W.Y., Quddus, M.A., Noland, R.B., 2004. Map matching in complex urban road networks. Brazilian Journal of Cartography 55 (2), 1–18.

Phuyal, B.P., 2002. Method and use of aggregated dead reckoning sensor and GPS data for map matching. In: Proceedings of Institute of Navigation-GPS (ION-GPS) Annual Conference, Portland, pp. 430–437.

Pyo, J.S., Dong-Ho, S., Sung, T.K., 2001. Development of a map matching method using the multiple hypothesis technique. In: Proceedings of IEEE Intelligent Transportation Systems Conference, Oakland, Calif., pp. 23–27.

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., Zhao, L., Noland, R.B., 2003. A general map matching algorithm for transportation telematics applications. GPS Solutions 7 (3), 157–167.

Quddus, M.A., Ochieng, W.Y., Noland, R.B., 2006. A high accuracy fuzzy logic-based map matching algorithm for road transport. Journal of Intelligent Transportation Systems: Technology, Planning, and Operations 10 (3), 103–115.

Quddus, M.A., Ochieng, W.Y., Noland, R.B., 2007. Current map matching algorithm for transport applications: state-of-the art and future research direction. Transportation Research Part C: Emerging Technologies 15, 312–328.

Srinivasan, D., Chue, R.L., Tan, C.W., 2003. Development of an improved ERP system using GPS and AI technique. In: Proceedings of IEEE Conference on Intelligent Transportation Systems, pp. 554–559.

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

Taylor, G., Blewitt, G., Steup, D., Corbett, S., Car, A., 2001. Road reduction filtering for GPS–GIS navigation. Transactions in GIS 5 (3), 193–207.

White, C.E., Bernstein, D., Kornhauser, A.L., 2000. Some map matching algorithms for personal navigation assistants. Transportation Research Part C: Emerging Technologies 8 (1), 91–108.

Yang, D., Cai, B., Yuan, Y., 2003. An improved map-matching algorithm used in vehicle navigation system. In: Proceedings of IEEE Conference on Intelligent Transportation Systems, pp.1246–1250.

Yin, H., Wolfson, O., 2004. A weight-based map matching method in moving objects databases. In: Proceedings of IEEE International Conference on Scientific and Statistical Database Management (SSDBM'04), pp. 437–438.

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

Zhao, L., Ochieng, W.Y., Quddus, M.A., Noland, R.B., 2003. An extended Kalman filter algorithm for integrating GPS and low cost dead-reckoning system data for vehicle performance and emissions monitoring. Journal of Navigation 56, 257–275.