

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



**CC creative commons**  
COMMONS DEED

**Attribution-NonCommercial-NoDerivs 2.5**

**You are free:**

- to copy, distribute, display, and perform the work

**Under the following conditions:**

**BY:** **Attribution.** You must attribute the work in the manner specified by the author or licensor.

**Noncommercial.** You may not use this work for commercial purposes.

**No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

**Your fair use and other rights are in no way affected by the above.**

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

# **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  
Phone: + 44 (0) 150 922 8545  
Fax: + 44 (0)1509 223981

**Email:** [n.r.velaga@lboro.ac.uk](mailto:n.r.velaga@lboro.ac.uk)

**\*Email:** [m.a.quddus@lboro.ac.uk](mailto:m.a.quddus@lboro.ac.uk)

**Email:** [a.l.bristow@lboro.ac.uk](mailto:a.l.bristow@lboro.ac.uk)

12<sup>th</sup> April 2009.

**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 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 module was tested using field data from dense urban areas and suburban areas. The algorithm identified 96.8% of the links correctly in an urban area and 97.01% correct links with a horizontal accuracy of 8.9m (95%) in a suburban area. This is superior to most existing topological MM algorithms and has the potential to support the navigation modules of many Intelligent Transportation System (ITS) services.

---

**Keywords:** Intelligent Transportation Systems (ITS); GPS, Spatial road network; Optimisation; Topological Map-Matching

## 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; Li and Fu 2003; Li and Chen, 2005; 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, 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; Quddus et al., 2003; Quddus et al., 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 require more input data and are 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 develop an enhanced tMM algorithm and to validate it using real-world field data and to assess its performance. 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 highest weight score link as 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 et al. (2006) tested four of these algorithms (including their own) using suburban data (7200 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. Their results are shown in the last column of Table 1.

Table 1 Performance of Existing Topological MM Algorithms

Author 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 et al., (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							
Greenfled (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%	88.6%
							18.1 m (95%)

\* The same positioning data is used to check the developed algorithm performance (See data set 6 in table 2)

Table 1 suggests that the performance of tMM algorithms with respect to the correct link identification ranges from 85% to 98.5%; and the horizontal accuracy ranges 32m (95%) to 18.1m (95%). 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 dataset, 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.
- (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 connectivity, 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 such as Greenfeld (2002) or deriving empirically such as Quddus et al. (2003). 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/sec; 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) matching 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 to a junction*).

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 a weight scheme
- (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). Initially, the error ellipse concept was derived by Zhao (1997). The same concept, was used by Ochieng et al. (2004) and Quddus (2006a), is considered in this research. All the links that are either inside the error ellipse or crossing the error ellipse or tangent to the error ellipse 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 ellipse 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.

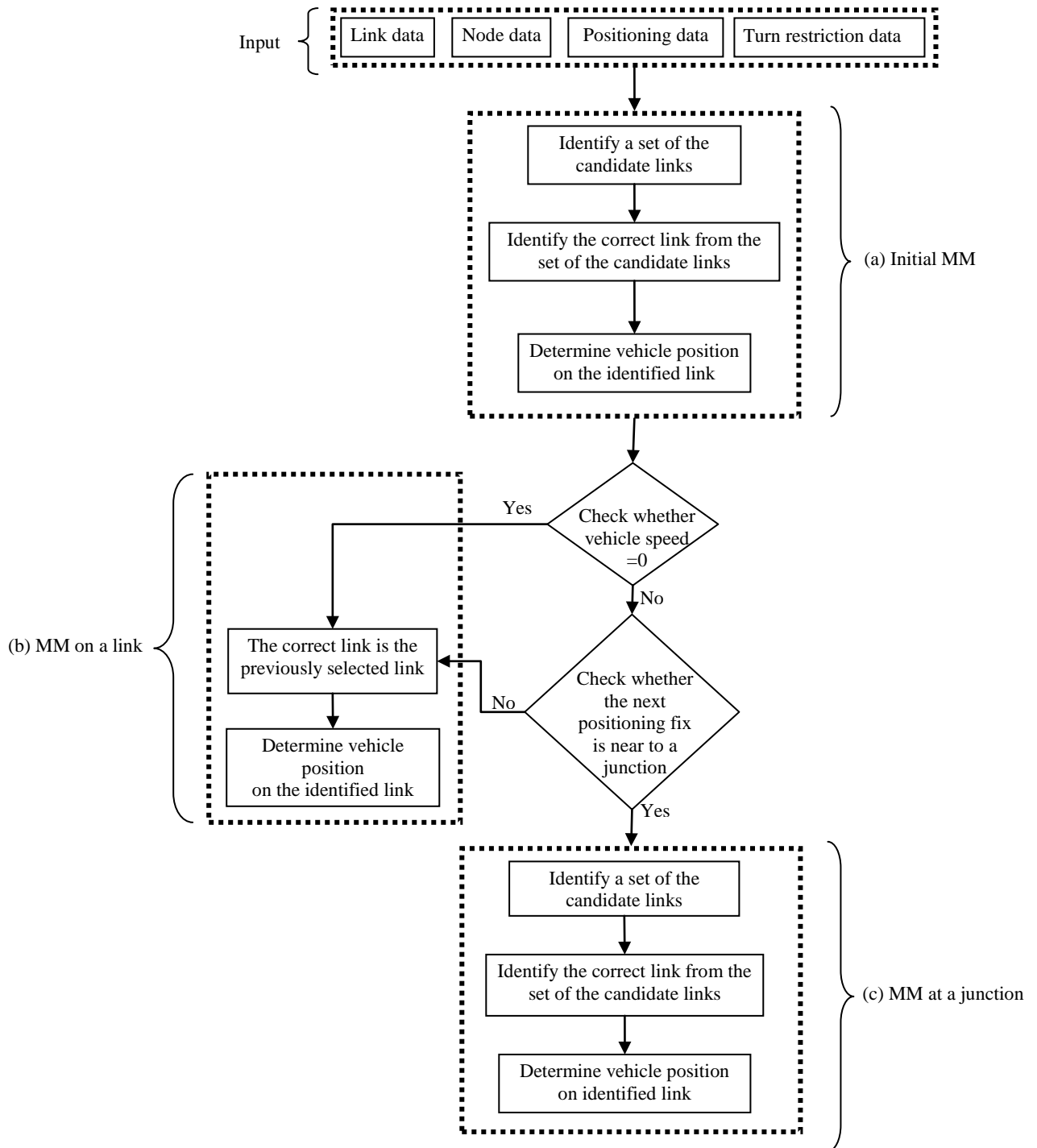


Fig. 1. A flow-chart representing the enhanced tMM algorithm.

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 equation 1.

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  $\Delta D$  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. If the positioning fix falls on the link (i.e.  $D=0$ ) the proximity weight parameter,  $f(D)$ , is 1; and if the distance between the positioning point and the link is more than 160m (i.e.  $D > 160m$ ), the proximity weight parameter,  $f(D)$ , is -1. This is because an empirical investigation suggests that if  $D$  is higher than 160m then the algorithm wrongly identified the link.

$$\begin{aligned} TWS &= W_h + W_p \\ TWS &= H_w f(\theta) + D_w f(D) \end{aligned} \quad (1)$$

where

$$\begin{aligned} f(\theta) &= \cos(\theta) \\ f(D) &= \left[ \frac{(80 - D)}{80} \right] \end{aligned}$$

$TWS$  is the total weight score

$W_h$  denotes the weight for heading

$W_p$  represents the weight for proximity

$H_w$  is the heading weight coefficient

$D_w$  is the proximity weight coefficient

$\theta$  denotes the angle difference between the vehicle heading and link direction with respect to the north.

$TWS$  is calculated for each candidate link and identifies the link with the highest  $TWS$  as the correct link. The vehicles 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. 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 these two checks is shown in equation 2.

$$\overbrace{d_1 \geq (d_2 + d_{threshold})}^{checkone} \text{ and } \overbrace{h_{RMS} \geq (\delta_i + h_{threshold})}^{Checktwo} \quad (2)$$

Where  $d_1$  is the distance between the previously map-matched positioning point to the downstream junction, and  $d_2$  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 parameter, a distance threshold ( $d_{threshold}$ ) as shown in equation (2) needs to be considered. Here, the  $d_{threshold}$  is considered as a positive value. GPS position fixes are less reliable when speed of the vehicle is less than 3 m/sec (Quddus et al., 2007; Taylor et al., 2001). To overcome this, a bearing threshold ( $h_{threshold}$ ) is added to  $\delta_i$ . This is to ensure that the map-matching process does not miss vehicles that may be at a junction.

$h_{RMS}$  denotes the Root Mean Square (RMS) error value of all headings related to the positioning fixes mapped on the previously identified link.  $\delta_i$  is the absolute value of angle difference between the vehicle heading at the current position fix and the previously identified link direction. The distance threshold ( $d_{threshold}$ ) and the bearing threshold ( $h_{threshold}$ ) values were derived empirically from field data of 1800 GPS/DR fixes, and identified as 20m and  $5^\circ$  respectively. However, these threshold values depend on quality and scale of digital map, time interval of each positioning information, and quality of navigation data output from GPS/DR sensor.

If equations (2) is satisfied then the algorithm assumes that the vehicle is moving on the previously matched link, and the algorithm 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 TWS at junction is given below:

$$TWS = (H_w \cos(\theta)) + (D_w f(D)) + (C_w C_c) + (T_w C_t) \quad (3)$$

where

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

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

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

$C_c$  equals 1 if a candidate link (within the set of the candidate links) is directly connected to the previously identified link and -1 otherwise.  $C_t$  equals 1 if a vehicle can legally make a turn to a link and -1 otherwise.



$H_w$ ,  $D_w$ ,  $C_w$  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,  $\text{Cos}(\theta)$ , proximity,  $f(D)$ , connectivity,  $C_c$  and turn restrictions ( $C_t$ ) are specified such that their values lie between +1 to -1 for any possible values of the factors. This constraint allows to control the relative importance of weight coefficients. Although values of  $\theta$ ,  $D$ ,  $C_c$  and  $C_t$  in equation (3) 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.

Here 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 coefficients that minimise the total map-matching error in terms of identification of the correct links.

### *3.3 Optimisation of the weights*

The process starts with the map-matching of a positioning fix near to a junction and generates random values for the coefficients between 1 and 100 in such a way that the sum of all four coefficients equals 100. Using these selected values, the process then a map-matches at that junction and identifies the link on which the MM algorithm locates the vehicle. Since the actual link is known, it is possible to see whether the MM algorithm has identified the link correctly. If the algorithm fails to identify the true link among candidate links then the algorithm regenerates the random values and repeats the map-matching at that junction. This process continues until the algorithm selects the true 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, for 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 wrong link identification for each set. As the other variables,  $\theta$ ,  $D$ ,  $C_c$  and  $C_t$ , in equation (3) vary from 1 to -1 for any possible values, it is assumed that the map matching error with respect to correct link identification ( $MM_{error}$ ) is a function of the weights  $H_w$ ,  $D_w$ ,  $C_w$  and  $T_w$  only.

$$MM_{error} = f(H_w, D_w, C_w, T_w) \quad (4)$$

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 ( $H_w D_w$ ,  $H_w C_w$ ,  $H_w T_w$ ,  $D_w C_w$ ,  $D_w T_w$  and  $C_w T_w$ ), a functional relationship can be written as:

$$\ln(MM_{error}) = \alpha + [\beta_{h1} H_w + \dots + \beta_{t1} T_w] + [\beta_{h2} H_w^2 + \dots + \beta_{t2} T_w^2] + \left[ \frac{\beta_{h3}}{H_w} + \dots + \frac{\beta_{t3}}{T_w} \right] + [\beta_{hd} (H_w D_w) + \dots + \beta_{ct} (C_w T_w)] + \varepsilon_i \quad (5)$$

where

$\alpha$  is an intercept term.

$\beta_{h1}, \beta_{h2}, \dots, \beta_{t1}, \beta_{t2}, \beta_{t3}$  are the regression coefficients for heading, proximity, connectivity and turn restriction weights.

$\varepsilon_i$  is the error term.

Initially, all 18 variables are considered in the regression analysis. The regression analysis is carried using a step-by-step backward elimination process, at each step one statistically insignificant parameter, based on t-stat (which is a measure of the relative influence of the independent variable on the dependent variable), is removed. 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.

Minimization:

$$MM_{error} = \exp \left\{ \alpha + [\beta_{h1}H_w + \dots + \beta_{t1}T_w] + [\beta_{h2}H_w^2 + \dots + \beta_{t2}T_w^2] + \left[ \frac{\beta_{h3}}{H_w} + \dots + \frac{\beta_{t3}}{T_w} \right] + [\beta_{hd}(H_w D_w) + \dots + \beta_{ct}(C_w T_w)] + \varepsilon_i \right\} \quad (6)$$

subject to:

$$\begin{aligned} H_w + D_w + C_w + T_w &= 100 \\ 1 \leq (H_w, D_w, C_w, T_w) &\leq 100 \end{aligned}$$

Optimisation of equation (6) 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 minimisation of map-matching error ( $MM_{error}$ ); here 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 previously matched point to 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 empirical analysis, this threshold fixed at 40m in this case. However, a variable threshold may be defined based on the road width (if the data are available) and quality of navigation sensors.

#### 4. Data collection

Five data sets, with 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 datasets provided positioning data every second. A 1:2500 scale digital map was used in the map-matching process. Data set 1, 2 and 5 were obtained from Quddus et al. (2006) and Quddus (2006a), whilst data sets 3 and 4 were generated as part of this study. The actual links on which the vehicle was driven were known for all datasets.

Table 2 Positioning Datasets

Data set	Operational environment	Data collection date (month and year)	Equipment used	Sample size: data points	Location characteristics
1	<b>Urban:</b> (Central London)	Jun-02	GPS/DR	1280	Urban characteristics such as tall buildings, bridges, flyovers, and dense road network
2	<b>Suburban:</b> (South London)	Nov-05	GPS/DR	1812	Suburban area
3	<b>Rural:</b> (Loughborough, East Midlands, UK)	Mar-08	GPS/DR	1200	University roads, and other rural roads
4	<b>Urban:</b> (Central London)	May-08	GPS/DR	2814	Dense urban road network, tall buildings, bridges, flyovers, tunnels.
5	<b>Urban:</b> (Washington, DC, USA)	Jan-09	GPS	3600	Urban characteristics such as bridges, flyovers,
6*	<b>Suburban:</b> (West of London - near Reading)	Aug-05	GPS/DR and Carrier-phase GPS	1228	Suburban area

\* The dataset six was also used to examine the existing tMM algorithms performance by Quddus et al. (2006) (for details see column 8 in table 1)

#### *4.1 Data for Optimisation*

Datasets 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, datasets 4, 5 and 6 were used. For the collection of the fourth dataset, 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 26<sup>th</sup> May 2008. The data set five was collected on a pre-planned route in urban areas of Washington, DC, USA, using a 16-channel single frequency high sensitivity GPS receiver, on 13<sup>th</sup> Jan 2009. For both the 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 set 4 and 5 was about 18 km and 17 Km respectively. The test trajectory for dataset 4 (in central London) and data set 5 (in Washington, DC) are shown in Fig. 2 and Fig. 3 respectively. But, no reference (actual) trajectory in terms of true vehicle positions was available for the both fourth and fifth datasets 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 dataset 6 obtained from Quddus et al. (2006) and Quddus (2006a). This allows the performance to be assessed in terms of both link identification and horizontal accuracy. It should be noted that the dataset 6 was also used to examine the existing tMM algorithms performance by Quddus et al. (2006) (for details see column 8 in table 1).



Fig. 2. Test route in central London.

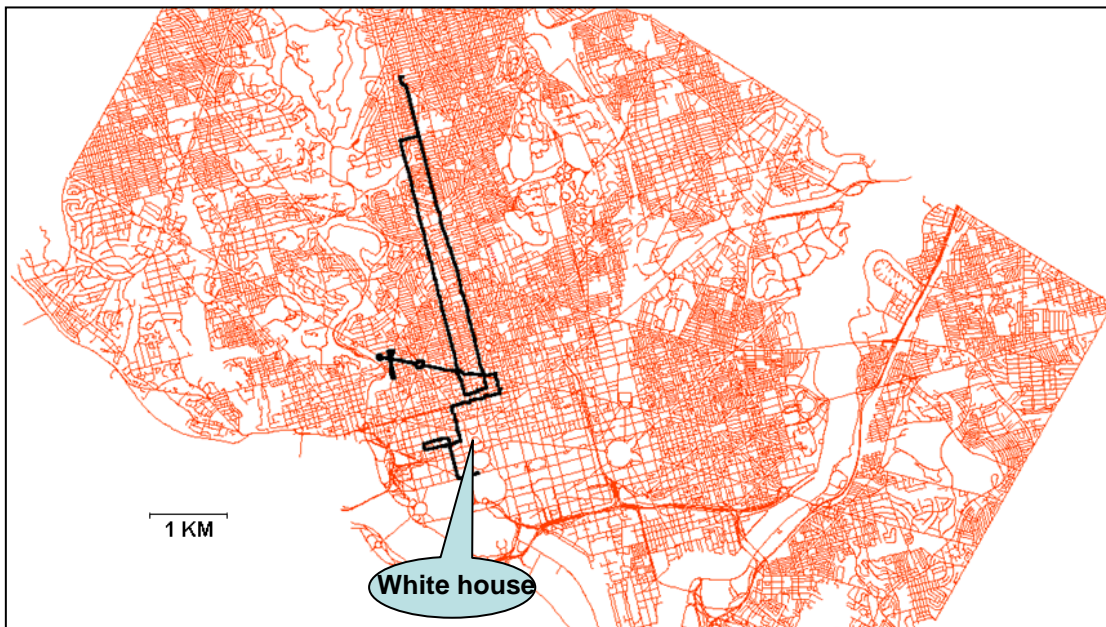


Fig. 3. Test route in Washington, DC, USA.

## 5. Optimisation results

Table 3 shows the best fitting regression models. Corresponding adjusted  $R^2$  for each model is also illustrated in the table. The adjusted  $R^2$  provides what percentage of behaviour of dependant variable (i.e. percentage of wrong link identification) is explained by the independent variables. Because of the regression through the origin, the adjusted  $R^2$  for all models was found to be high. As mentioned, the sum of the four weight is 100. If an intercept (i.e.,  $\alpha$ ) 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. This is the reason why the regression line is forced through the origin. It can be seen that different specifications are achieved for different operational environments suggesting that the use of one specification for all environments can be misleading.

Table 3 Regression Models for Urban, Suburban and Rural Area

Weights	Urban		Suburban		Rural	
	<i>Coefficient</i>	<i>T-stat</i>	<i>Coefficient</i>	<i>T-stat</i>	<i>Coefficient</i>	<i>T-stat</i>
$H_w$	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.



The optimal values of weights for the three operational environments are illustrated in Table 4. In suburban and rural areas the weight for connectivity ( $C_w$ ) and weight for turn restriction ( $T_w$ ) are not very important; whereas, weights for heading and proximity ( $H_w, D_w$ ) are almost equally important, probably because in suburban areas and rural areas the quality of GPS/DR positioning fixes is good and the road network is less dense. In dense urban areas, heading and connectivity weights ( $H_w, C_w$ ) are almost equal and the weight for proximity ( $D_w$ ) is less important, this is because in dense urban areas roads are in close proximity and the quality of positioning information is bad compared to open areas.

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

## 6. Algorithm performance

The developed algorithm identifies the operational environment on which vehicle is travelling, and selects the corresponding weight score, suitable for that environment, from the weight matrix (table 4). The identification of operational environment (whether the vehicle is in an urban area or a suburban area or a rural area) could be determined based on the complexity of road network, landuse data, building height data etc. In urban areas, roads are proximity to each other compared to suburban or rural areas. The total length of the road network (i.e. sum length of individual road links) per a given area in an urban area will be greater than a suburban area; and similarly, in a suburban area it will be greater than a rural area. The proposed algorithm identifies the operational environment based on complexity of

road network (i.e. total length of roads per a given area). However, one can further improve the identification of operation environment using building height data and landuse data; if such data is available.

The values of the weights from Table 4 are applied for algorithm testing. For dataset 4 (urban area in central London) the enhanced tMM algorithm identified 96.8% of the road segments correctly. In case of dataset 5 (urban areas in Washington, DC) the success rate is 95.93%. From the above two data sets, the percentage of correct link identification in urban areas is considered as 96.36 (average value of 96.8% and 95.93%). In terms of computational speed, the algorithm carried out the map-matching of 180 positioning fixes per sec (with a laptop of 1GB 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 along with raw positioning fixes (with star symbol) and map-matched fixes (with round symbols).

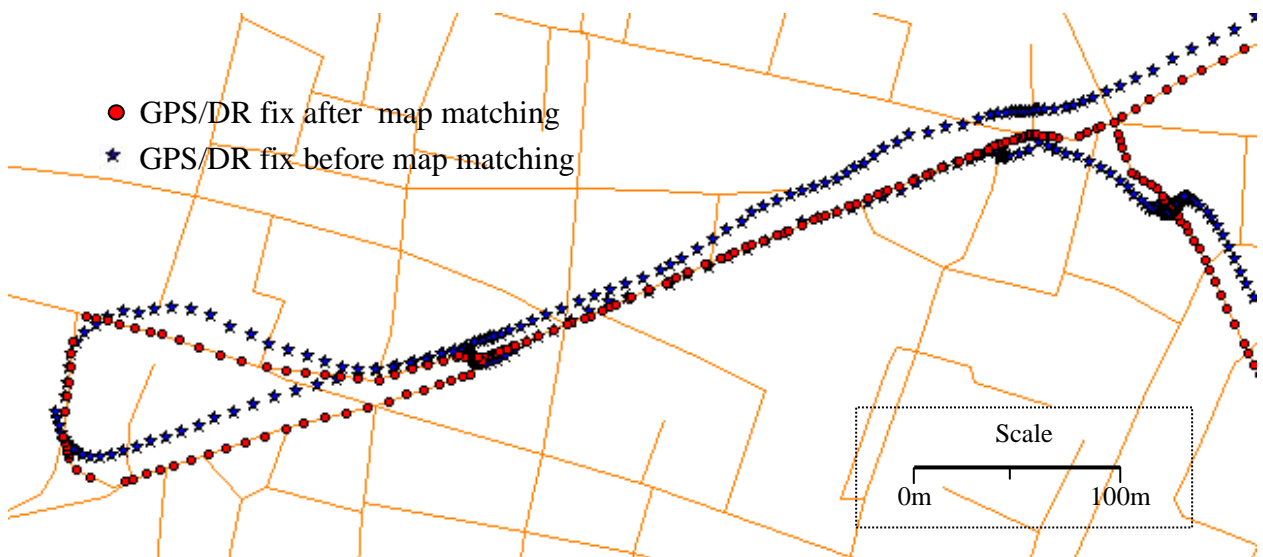


Fig. 4. A part of test road with map-matched positions.

The enhanced tMM algorithm was then applied to the sixth dataset (suburban area). The performance of the algorithm was evaluated in terms of both correct link identification and position determination. The algorithm identified 97.01% of the segments correctly with a

horizontal (2D) accuracy of 8.9m (95%). The along-track and cross-track errors were found to be 8.6m (95%) and 2.4m (95%) respectively.

A fuzzy logic-based MM algorithm developed by Quddus et al. (2006) is capable of identifying 99.2% of the links correctly with the horizontal positioning accuracy of 5.5 m (95%) for suburban road network; and in case of urban road network, the algorithm can identify 98.5% of the links correctly . According to the results summarised by Quddus et al. (2006), the enhanced tMM algorithm developed in this study outperforms most existing tMM algorithms and its performance approaches to that of a pMM or 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 considering some 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 improved performance of the algorithm. The enhanced topological MM algorithm identified 96.36% of the road segments correctly in an urban area; and 97.01% of the road segments with a horizontal accuracy of 8.9m (95%) 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 datasets collected from various road environments. This requires further testing.

This topological MM algorithm is fast, simple and very efficient and therefore, has a 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. 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.

Further research will investigate the optimisation of weights using more positioning data from each operational environment to ensure that these optimal values are transferable. The performance of a tMM algorithm can further be improved by investigating the causes that are responsible for 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, pp. 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, pp.15-22 (April 22-24).
- 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, D.C.
- 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 29 th International Symposium on Automotive Technology and Automation (ISATA)*, Florence, Italy, vol. 10. pp. 121–126.
- 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), pp. 69-76.

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.

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, pp. 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), pp. 1-18.

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.

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.

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), pp.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), pp. 103-115.

Quddus, M.A., 2006a. High integrity map matching algorithms for advanced transport telematics applications. PhD Thesis, Centre for Transport Studies, Imperial College London, UK.

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, pp. 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), pp. 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), pp. 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, pp. 257–275.



### List of Figures

Figure 1: A flow-chart representing the enhanced tMM algorithm

Figure 2: Test route in central London.

Figure 3: Test route in Washington, DC, USA

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

### List of Tables

Table 1: Performance of Existing Topological MM Algorithms

Table 2: Positioning Datasets

Table 3: Regression Models for Urban, Suburban and Rural Area

Table 4: Optimisation Result