Development of a weight-based topological map-matching algorithm and an integrity method for location-based ITS services

Development of a weight-based topological map-matching algorithm and an integrity method for location-based ITS services

Nagendra R. Velaga

(A 684746)

A doctoral thesis submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University

July 2010

Dedicated to my *parents* and *god*

Abstract

The main objective of this research is to enhance navigation modules of location-based Intelligent Transport Systems (ITS) by developing a weight-based topological map-matching algorithm and a map-aided integrity monitoring process.

Map-matching (MM) algorithms integrate positioning data from positioning sensors with spatial road network data to identify firstly, the road link on which a vehicle is travelling from a set of candidate links; and secondly, to determine the vehicle's location on that segment. A weight-based topological MM algorithm assigns weights for all candidate links based on different criteria such as the similarity in vehicle movement direction and link direction and the nearness of the positioning point to a link. The candidate link with the highest total weighting score is selected as the correct link. This type of map-matching algorithm is very popular due to its simplicity and speediness in identifying the correct links. Existing topological map-matching algorithms however have a number of limitations: (1) employing a number of thresholds that may not be transferable, (2) assigning arbitrary weighting coefficients to different weights, (3) not distinguishing among different operational environments (i.e., urban, suburban and rural) when determining the relative importance of different weights and (4) not taking into account all available data that could enhance the performance of a topological MM algorithm. In this research a novel weight-based topological map-matching algorithm is developed by addressing all the above limitations. The unique features of this algorithm are: introducing two new weights on turn restrictions and connectivity at junctions to improve the performance of map-matching; developing a more robust and reliable procedure for the initial map-matching process; performing two consistency checks to minimise mismatches and determining the relative importance of different weights for specific operational environments using an optimisation technique.

Any error associated with either the raw positioning data (from positioning sensors) or spatial road network, or the MM process can lead to incorrect road link identification and inaccurate vehicle location estimation. Users should be notified when the navigation

system performance is not reliable. This is referred to as an integrity monitoring process. In this thesis, a user-level map-aided integrity method that takes into account all error sources associated with the three components of a navigation system is developed. Again, the complexity of the road network is also considered. Errors associated with a spatial road map are given special attention. Two knowledge-based fuzzy inference systems are employed to measure the integrity scale, which provides the level of confidence in map-matching results.

Performance of the new MM algorithm and the integrity method was examined using a real-world field data. The results suggest that both the algorithm and the integrity method have the potential to support a wide range of real-time location-based ITS services. The MM algorithm and integrity method developed in this research are simple, fast, efficient and easy to implement. In addition, the accuracy offered by the enhanced MM algorithm is found to be high; it is able to identify the correct links 97.8% of the time with an horizontal accuracy of 9.1 m $(\mu + 2\sigma)$. This implies that the developed algorithm has high potential to be implemented by industry for the purpose of supporting the navigation modules of location-based intelligent transport systems.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisors Professor Abigail Bristow and Dr. Mohammed A Quddus for their guidance, timely discussions, financial support, constructive comments and critical reviews and exceptional support. This research would not have been possible without the support from my supervisors.

I would like to thank Dr Yuheng Zheng of Loughborough University, for helping me to collect positioning data in Nottingham, UK and Dr Sabyasachee Mishra of Wayne State University, Detroit for helping in positioning data collection in USA. I would like to express my heartfelt gratitude to Professor Sunder L Dhingra, Professor K.V. Krishna Rao and Professor Tom V. Mathew of IIT Bombay whose teaching and supervision have provided fundamental of transportation and created research interest particularly in the field of GIS and GPS applications in transport.

I am also grateful to the staff and friends at the Civil and Building Engineering Department, in particular Udityasinh Gohil, Dr. Lisa Davison and Helen Newbold for their support. Finally, I would like to express my deepest appreciation to my family for their patience, continuous moral support and encouragement during the course of this research, especially to my parents, my fiancée Dr. SreeLalitha.

Nagendra R Velaga
Transport Studies Group
Loughborough University

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Around the world transport and traffic related problems have been increasing over the recent decades. It is increasingly recognised that simply attempting to provide additional infrastructure to meet such demand is not sustainable (Chowdhury and Sadek, 2003). Among a range of possible supply and demand based measures, Intelligent Transport Systems (ITS) have the potential to improve safety, mobility and efficiency, and reduce congestion and environmental impacts through the use of advanced computer, communication, navigation, information and vehicle sensing technologies (Sussman, 2005). Current ITS applications include traveller information services, route guidance, adaptive signal control, variable message signs, pre-trip and en-route travel information, travel demand management, electronic payment systems, and incident management. Most ITS services including navigation and route guidance, bus arrival information at bus stops (*countdown* in London, i.e. iBus), fleet management (a GPS-based taxi dispatching service in Singapore) and distance-based insurance premiums require vehicle positioning data in real-time.

There are three key elements involved in locating a vehicle's position on a road. They are: (1) positioning systems/sensors (2) a Geographic Information System (GIS)-based road map and (3) a map-matching (MM) algorithm (Taylor and

Blewitt, 2006). A number of positioning systems/sensors are normally used to obtain vehicle positioning data. These are:

(a) dead reckoning (DR) systems which use a relative positioning technique to determine the location of a moving vehicle requiring vehicle speed and its direction of movement (White et al., 2000; Zhao et al., 2003; Li and Fu, 2003);

(b) ground based beacon systems (Bernstein and Kornhauser, 1996; White et al., 2000), and

(c) Global Navigation Satellite System (GNSS) such as GPS (White et al., 2000; Greenfeld, 2002; Zhao et al., 2003; Quddus et al., 2003).

Recently, GPS has become more popular as it offers a fast and convenient method of obtaining positioning data that is well-suited for viewing in a Geographic Information System (GIS) (Chen et al., 2003).

A map-matching (MM) algorithm is used to augment positioning data from a navigation system with spatial road network data. A MM algorithm makes use of a range of navigational data including vehicle position, heading, speed and road network topology to identify the road segment on which a vehicle is travelling and the vehicle location on that road segment. The key task for a MM algorithm is to identify the correct road segment from a pool of candidate road segments.

Various map-matching techniques (e.g. geometric, topological, probabilistic and advanced) have been developed for land vehicle navigation (Bernstein, 1996; Kim et al., 1996; White et al., 2000; Taylor et al., 2001; Phuyal, 2002; Srinivasan et al., 2003; Li and Fu 2003; Yin and Wolfson, 2004; Blazquez and Vonderohe, 2005; Quddus et al., 2007; Sohn, 2009). The earliest algorithms, developed in the 1990s, used only geometric information, the shape of the curve of the road segment (Kim et al., 1996; Bernstein, 1996; White et al., 2000; Quddus et al., 2007). A MM algorithm that uses only geometric information is known as a geometric MM algorithm (gMM). These gMM algorithms are the simplest and

fastest to implement as they require very little information. However, gMM algorithms perform poorly, especially when matching at junctions, complex roundabouts and parallel roads. These gMM algorithms may be improved by including historical data (such as the previously matched road segment), vehicle speed and topological information from a spatial road map (such as link connectivity). A MM algorithm that uses such additional information is called a topological MM (tMM) algorithm (Greenfeld, 2002; Quddus et al., 2003; Li et al., 2005; Quddus et al., 2007). Obviously, the performance of a tMM is much better than that of a gMM (White et al., 2000; Greenfeld, 2002). A probabilistic MM (pMM) algorithm uses 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, bayesian multiple hypothesis technique, fuzzy logic (FL) and artificial neural network (ANN) techniques (Pyo et al., 2001; Yang et al., 2003; Syed and Cannon, 2004; Fu et al., 2004; Haibin et al., 2006; Quddus et al., 2007a; Smaili et al., 2008).

The raw positioning data from a navigation system contains errors due to satellite orbit and clock bias, atmospheric (ionosphere and troposphere) effects, receiver measurement error and multipath error (Kaplan and Hegarty, 2006). GIS-based road maps include errors which can be geometric (e.g. displacement and rotation of map features) and/or topological (e.g. missing road features) (Goodwin and Lau, 1993; Kim et al., 2000). Even when raw positioning data and map quality are good, MM techniques sometimes fail to identify the correct road segment especially at roundabouts, level-crossings, Y junctions, dense urban roads and parallel roads (White et al., 2000; Quddus et al., 2007a). Any error associated with either the raw positioning points, the digital map used, or the MM process employed can lead to a mismatch. This could mislead the users and make the ITS service ineffective. It is therefore important to enhance the GPS/GIS/MM system, so as to enable the further improvement of vehicle navigation system.

Vehicle navigation systems may be enhanced by reducing the errors in the positioning sensors (e.g. GPS) or improving the quality of the spatial road network map or further enhancement of map-matching algorithms. For instance, the Russian Global Orbit Navigation Satellite System (GLONASS) and the upcoming European Galileo system, along with a DR system, may enhance the performance of existing vehicle positioning systems. However, a good map-matching algorithm will still be needed to physically locate a vehicle on a road network. Current map-matching algorithms have many constraints and limitations, especially in typical operational environments (such as dense urban areas) in which highly accurate positioning data are essential (Quddus et. al., 2007a). Enhancement in vehicle positioning can also be obtained by improving the quality of spatial network data. This is however costly and time consuming at a national level. Therefore, further improvement in map-matching algorithms is a viable alternative in order to enhance the vehicle navigation system.

Among the four different map-matching algorithms (e.g. geometric, topological, probabilistic and advanced) identified in the literature, an advanced MM (aMM) algorithm that uses these more refined approaches, outperforms other MM algorithms (Quddus et al., 2006a). These aMM algorithms require more input data 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 greater potential to be implemented in real-time applications by industry as its processor would require less memory. There are a number of constraints and limitations in existing tMM algorithms. Once such limitations are addressed, it is expected that the performance of a tMM algorithm could be comparable to that of a pMM algorithm or an aMM algorithm.

As mentioned before, any error associated with the three main components (raw positioning systems/sensors, digital map, MM algorithms) may lead to a wrong location identification. Users should be notified when the system performance is

not reliable. This is vital not only for safety critical ITS applications such as emergency vehicle management and vehicle-based collision avoidance but also liability critical applications including electronic toll collection systems and distance-based pay-as-you-drive road pricing. For example, wrong vehicle location identification, due to errors associated with a navigation system, in an emergency vehicle routing ITS service may delay ambulances arrival at an accident site. If a driver is informed when the system performance is not reliable then the driver will not blindly depend on the output of the vehicle navigation system.

1.2 Problem statement and intention of this thesis

A MM algorithm that uses historical data (such as the previously matched road segment), vehicle speed and topological information on the spatial road network (such as link connectivity) is called a topological MM (tMM) algorithm (Greenfeld, 2002; Li et al., 2005; Quddus et al., 2003). An algorithm which further assigns weights for all candidate links based on different criteria such as the similarity in vehicle movement direction and link direction, the nearness of the positioning point to a link, and the connectivity of a candidate road link to the previously travelled road link is known as a weight-based topological MM algorithm (Quddus et al., 2003). 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). According to the current literature, the generic limitations of such weight based topological map-matching algorithms are:

1. Assuming equal importance (or derived empirically) of the weights considered in the weight-based tMM algorithm;

2. operational environments (i.e., urban, sub-urban and rural) are not considered while determining the relative importance of different weights and

3. Employing a number of arbitrary thresholds which may not be transferable.

Moreover, in many algorithms the horizontal accuracy (2D) has not been identified due to a lack of reference (true) vehicle positioning data. In order to enhance the vehicle navigation module of an intelligent transport system a more robust tMM algorithm, which can overcome the limitations of existing algorithms, needs to be developed.

The concept of user-level integrity monitoring has been successfully applied to air transport navigation systems. Very little research has been devoted to land vehicle navigation system integrity monitoring. The primary difference is that in addition to the errors associated with the space segment, one needs to consider the errors associated with the digital map and the map-matching process when monitoring the integrity of a land vehicle navigation system. Research on land vehicle integrity monitoring has concentrated on either the integrity of raw positioning data obtained from GNSS (Philipp and Zunker, 2005; Feng and Ochieng, 2007; Lee et al., 2007) or the integrity of the map-matching process (Yu et al., 2006; Quddus et al., 2006c; Jabbour et al., 2008). Clearly, considering both sources of error concurrently, in identifying the goodness (trustability) of final positioning fixes, should lead to a better outcome.

Moreover, taking the complexity of the road network (i.e. operational environment) into account may further improve the integrity process. This is because, although the raw positioning points from GPS contain some errors, a good map-matching algorithm can identify the correct road segment if the road network in which the vehicle is travelling is not complex. None of the existing research on land vehicle navigation integrity methods has considered the complexity of the road network in the integrity monitoring process.

1.3 Research Aim and Objectives

The primary aim of the this research is to enhance navigation modules of location-based Intelligent Transport Systems (ITS) by developing a weight-based topological map-matching algorithm and a map-aided integrity monitoring process.

To achieve the above aim, the following objectives are formulated:

1. To identify positioning requirements expressed as Required Navigation Performance parameters for a range of location-based ITS.

2. To critically assess existing topological map-matching algorithms and integrity methods.

3. To develop an improved weight-based topological map-matching algorithm and introduce an optimisation technique to identify the relative importance of weight scores in different operational environments.

4. To explore the transferability of tMM to different contexts; and systematically identify the sources of error in the tMM algorithm to further enhance it.

5. To develop an improved integrity method by taking into account errors associated with the positioning data, GIS map and map-matching process concurrently.

6. To identify location based intelligent transport systems that may be supported by the enhanced tMM algorithm and the integrity method.

Extensive positioning data is obtained from different operational environments, using a low-cost GPS receiver and a carrier-phase GPS receiver integrated with a high-grade Inertial Navigation System (INS) to examine the relative importance of weight scores in different operational environments and to evaluate the performance of the tMM algorithm and the integrity method. As most ITS services need continuous vehicle location information, here the positioning data is recorded every second. The aim is to develop a generic tMM algorithm that does not assume an O-D pair in the map-matching process.

1.4 Outline of the Thesis

This thesis is organised into ten chapters. This chapter has described the research background, current research issues, aim and objectives and the structure of the thesis:

Chapter Two provides a brief introduction to ITS services and ITS taxonomy showing various ITS services that require vehicle positioning information and services that may use the vehicle positioning information. This is followed by explanation of Required Navigation Performance (RNP) parameters in the context of land vehicle navigation. Then, a review of RNP for ITS, aiming to focus on how these RNP parameters are derived and the strategy to identify each RNP parameter for a range of ITS services, is provided. The chapter ends with evidence of RNP parameters for a range of ITS services and the target ITS services in this thesis.

Chapter Three critically reviews geometric, topological, probabilistic and advanced map-matching algorithms. This is followed by a detailed literature review of topological map matching (tMM) algorithms by classifying then into: (1) non-weight based on-line algorithms, (2) weight based on-line algorithms and (3) off-line algorithms. Existing tMM algorithms performance is reported. The limitations of existing tMM algorithms are identified.

Chapter Four introduces the role of integrity methods in the context of land vehicle navigation and provides a detailed review of integrity methods. Performance and limitations of existing integrity methods are examined. Chapter Five specifies all the positioning data sets collected or used to identify the optimal weight scores in the tMM algorithm and to test both the enhanced tMM algorithm and the integrity method.

In Chapter Six, a weight-based topological MM algorithm is described by dividing it into: initial map-matching, matching on a link and map-matching at a junction. Then an optimisation technique to identify the relative importance of weight scores is explained. The optimisation technique includes development of relation between map-matching errors and the weight scores and identification of optimal weight scores using the Matlab optimisation toolbox. Then, the performance of the tMM algorithm is assessed using positioning data collected in London, UK and Washington, USA. The positioning accuracy of the algorithm (i.e., horizontal accuracy, along track error and cross track error) is examined using highly accurate positioning data obtained from carrier-phase GPS integrated with INS.

In Chapter Seven further enhancement of the tMM algorithm developed in the previous chapter is carried out using a sequential process of map-matching error detection, mitigation and performance re-evaluation. Firstly, all the mis-matching cases are identified by carefully analysing three positioning data sets from UK, USA and India. Here three different GIS maps are used. The mismatches due to positioning data errors, GIS map errors and map-matching process errors are measured. Secondly, a number of strategies are identified to avoid these mismatches enabling enhancement of the topological map-matching algorithm Thirdly, the tMM algorithm developed in Chapter Six is further modified/improved. Finally the enhanced algorithm performance is re-evaluated before and after improvement.

Chapter Eight starts with description of how to consider the integrity of raw positioning points, digital map errors and integrity of MM process concurrently. This is followed by development of enhanced integrity monitoring process for land vehicle navigation systems, which also includes two fuzzy inference systems. The chapter ends with performance evaluation of the developed integrity method.

Chapter Nine compares the performance of the developed tMM algorithm and the integrity method with existing algorithms' performance; and examines the extent to which the enhanced algorithm and the integrity method can support the navigation requirements of ITS services. Chapter Ten summarises the research contribution in the area of map-matching process and integrity methods. Research directions for future improvement of the MM algorithms and the integrity methods, which can support the limitations of this work, are provided.

Chapter 2

Location-based ITS services and their RNP requirements

2.1 Introduction

This chapter contains a brief description of location-based transport services, and their positioning and navigation requirements, which are referred to as Required Navigation Performance (RNP) parameters. The RNP parameters are: accuracy, integrity, continuity, and availability. The basic factors that influence the RNP requirements for location-based ITS services are described. The RNP parameter requirements for a range of location-based ITS services are then discussed. The chapter ends with conclusion describing the ITS services that are targeted in this thesis.

A composite taxonomy of user services derived from Chadwick (1994), FHA (2003), Christos and Anagnostopoulos (2006) and Quddus (2006) is shown in Table 2.1. A location-based ITS service is a service that requires knowledge about vehicle position (X, Y, and Z coordinates or latitude longitude and height) continuously during its operation. In this table the user services with star mark (*) require navigation and positioning capabilities and the services marked with plus mark (+) are additional services which can be assisted by navigation and positioning technologies. These ITS services are under various levels of development, testing, and deployment. Further details of potential applications of ITS can be found in Mashrur and Sadek (2003), Nijkamp et al. (1996), Klein (2001), Quddus (2006), Kaplan and Hegarty (2006).

Table 2.1: Taxonomy for ITS

ITS User Groups	User services
Advanced Traffic Management System	Pre-trip travel information
	En-route driver information*
	Route guidance*
	Ride matching and reservation
	Traveller services information*
	Traffic control+
	Incident management*
	Travel demand management
	Traffic data quality management
	Highway-rail intersection (HRI)
	On-board emissions monitoring (OEM)+
Advanced Public Transport and Operations	Public Transport Management*
	En-Route Transit Information*
	Demand Responsive Transit
	Public Travel Security
	Automatic bus arrival announcements*
Electronic Payment System	Electronic Toll Collection+
	Electronic Parking Payment+
	GPS based Variable Road User Charging *
Commercial Vehicle Operations	Commercial Vehicle Electronic Clearance*
	Automated Roadside Safety Inspection
	On-Board Safety Monitoring+
	Commercial Vehicle Administrative Processes*
	Hazardous Material Transport+
	Commercial Fleet Management*
Emergency Management Systems	Emergency Vehicle Priority at Junction
	Emergency Notification and Personal Security*
	Disaster Response and Management
	Emergency Vehicle Management and Routing*
Advanced Vehicle Safety System	Vehicle based Collision Notification and Warning System*
	Vehicle-Based Collision Avoidance*
	Infrastructure-Based Collision Avoidance
	Sensor-Based Driving Safety Enhancement
	Safety Readiness
	Pre-Collision Restraint Deployment
	On-board safety monitoring*
Other Services	Information Management System
	Parking facility management+
	Maintenance and Construction Management System +
	Road Weather Management+
	Inter-modal Freight+
	Stolen Vehicles Recovery+
	Accident Survey+

* Services that require navigation and positioning capabilities

+ Additional services which can be assisted by navigation and positioning technologies

The navigation function of ITS is responsible for providing the physical location of a vehicle travelling on a road network. For example, accident and emergency (A&E) response vehicles under a safety-of-life accident notification service are equipped with a navigation system supported by GPS and a digital road map and a mobile communication system supported by Global System for Mobile Communication (GSM), General Packet Radio Service (GPRS) or Wireless Local Area Network (WLAN) or a combination of these technologies. In the event of an emergency such as an accident, the telemetric information from the navigation module is transmitted to the emergency service agencies through a wireless communication system. This information is not only useful in navigating an emergency vehicle to an accident, but also useful for road congestion relief. From the traffic control centre, vehicles receive real-time traffic updates on the state of the road network and the user is updated with the optimal route to reach a destination. This has the effect of re-distributing traffic around the network, thereby reducing congestion and its environmental impacts.

The positioning system must satisfy a wide range of requirements such as performance of the system with respect to horizontal positioning accuracy, the total outage and the maximum continuous outage of the system, warning users when the system performance falls below the expected level. These quantities are usually referred to Required Navigation Performance (RNP) parameters (*accuracy, integrity, continuity and availability*). The following section provides background and definitions of the RNP parameters.

2.2 Required Navigation Performance (RNP) parameters

RNP parameters were originally developed for aviation, by the International Civil Aviation Organization (ICAO), in 1983, to provide the specification of airspace, based on demonstrated levels of navigation performance and certain functional capabilities, to improve the system efficiency and safety (Cassell and Smith, 1995; Sang and Kubic, 1998). The same concept has been extended to marine and

land transport. The main parameters to measure the performance of a navigation system are *Accuracy, Integrity, Continuity, and Availability*. These parameters are required to improve the overall system efficiency. The navigation system performance parameters defined by FRP (1999), Ochieng and Sauer (2002), Kibe (2003), Ochieng et al. (2003), DoT (2004) and Quddus (2006) in the context of land navigation are summarised below.

2.2.1 Accuracy

Accuracy can be defined as the degree of conformance between the estimated or measured position of a point at a given time and its true or standard position. The true or standard position can be obtained from an independent source of high accuracy measurement such as GPS carrier phase observables (i.e., observations with accurate GPS) or high-resolution satellite imagery). Accuracy is generally presented as a statistical measure of system error; for example, the accuracy requirement of a GPS is specified at a 95% (2σ), i.e. for any estimated position at a specific location the probability that the position error is within the accuracy requirement should be at least 95 percent.

Accuracy is generally specified as:

Predictable - The accuracy of a position with respect to charted solution (geographic or geodetic coordinates of the earth).

Repeatable - The accuracy with which a user can return to a position whose coordinates have been measured at a previous time with the same navigation system.

Relative - The accuracy with which a user can measure position relative to that of another user of the same navigation system at the same time.

2.2.2 Integrity

Integrity is the ability of system to provide timely warnings to users when the system should not be used for navigation/positioning. Integrity has three components: Alarm limit, Time-to-alarm and Integrity risk.

Alarm limit (AL) is the error tolerance not to be exceeded without issuing an alert to the user. It represents the largest error that results in the safe operation.

Time to alarm (TTA) is the duration between the onset of a failure (i.e. the error greater than the allowable error (AL) limit) and an alert being issued by the user's receiver.

Integrity risk (IR) is the probability that an error exceeds the alarm limit without the user being informed within the time to alert.

For example, integrity of a system with AL: 10 m; TTA: 6 sec; IR: 10^{-2} per 1 hour means the probability that the system gives an alarm to the user on or before 6 seconds when the measured position error is equal to or more than 10 m is 99% in one hour.

2.2.3 Continuity

Continuity is the ability of a total system to perform its function without interruptions during the intended period of operation (POP). The continuity of the system is addressed by the Continuity Risk (CR). The CR can be defined as the probability that the system will be interrupted and will not provide guidance information for the intended period of operation. CR is the measure of system uncertainty.

For example, the CR requirement of a particular system is defined as 10^{-3} per hour then this means that the probability that the system will be interrupted and

will not provide guidance due to a loss of either system accuracy or system integrity is 0.1 % for an hour duration.

2.2.4 Availability

Availability of a navigation system is the percentage of time that the services of the system are usable within a specified coverage area. The service is available if the system accuracy, integrity and continuity requirements are satisfied.

2.3 Review of RNP for ITS

RNP parameters for ITS services are under development (DOT, 2004). Partial information on RNP parameters (alarm limit, time to alarm, integrity risk and continuity risk) is available in the literature (Ochieng and Sauer, 2002; DOT, 2004; Feng and Ochieng, 2007). This evidence is discussed here.

The criteria to derive RNP parameters for each ITS service vary significantly as service functions vary widely. For example, *accident and emergency (A&E) response vehicles* safety issues are predominant parameter to derive RNP. In the case of a *distance based road user charging* ITS service, where road users are charged based on distance travelled on a chargeable road, any error or outage of navigation system may result in errors in charges to users. In this case, commercial issues are major criteria to derive RNP. The following five parameters are identified as influencing issues or criteria in deriving RNP parameters for ITS services:

1) System performance requirements (SPR)
2) Commercial issues(CI)
3) Operational environments (OE)
4) Safety issues (SI)
5) Type of operation (ToO)

2.3.1. System performance requirements (SPR)

If an ITS service is intended to improve the efficiency of a transport system then the RNP parameters are derived from the operational aspects of the service. For instance, *automated vehicle identification* ITS system can support *bus arrival time information* (BATI) at bus stops and *bus priority at junction* (BPJ) sub-services. The RNP requirements for BPJ are more than BATI at bus stops because large error or outage of positioning system for BPJ service provides wrong signal timing at the junction, which causes more junction delay and congestion. On the other hand, any error in positioning system for BATI at bus stops leads to show bus arrival information with few seconds error which causes only little discomfort to passengers.

2.3.2 Commercial issues (CI)

For some ITS services, the ability to accurately charge users is the first priority while developing the RNP parameters. Examples include *electronic payment systems*, *variable road user charging* (VRUC) and *distance-based pay-as-you-go insurance schemes*. There could be financial consequences if the system accuracy is set too low or the system fails to alert the user and operator when the performance is below the expected levels. For example, a GPS based VRUC scheme may charge users erroneously especially in situations where different prices apply to two adjacent roads (i.e., charge for a motorway is £1.50 per mile and charge for a minor road parallel to the motorway is £0.25 per mile); the navigation system needs to identify the correct road segment on which a vehicle is travelling. Therefore, the required accuracy, integrity, continuity, and availability for such a service will be relatively high (Quddus et al., 2007b). The RNP parameters are driven by the needs to charge users accurately according to use.

2.3.3 Operational environments (OE)

The operation environments (urban, suburban, and rural) also influence the RNP parameters for ITS services. The RNP parameters for ITS services operating within an urban area should be high relative to sub-urban and rural areas. In urban areas complex road network and the impact of physical obstructions imposed due to tall buildings, narrow streets, flyovers, bridges and tunnels demand a more accurate positioning system. For instance, for navigation and route guidance ITS services in dense urban area, any small error or outage associated with position system can easily keep the vehicle on a wrong road, which misguides and confuses the driver. This is less likely to occur in simple rural and suburban networks. Therefore, the complexity of road network (i.e., operational environment: urban, suburban and rural) and surrounding land use and infrastructure will also influence the RNP parameters for ITS services.

2.3.4 Safety issues (SI)

For many ITS applications and services, safety is given the highest priority; these are 'safety-of-life' (SOL) critical applications. People are willing to take a risk if it can be considered to be as long as reasonable practicable (ALARP). Such an acceptable level of risk is referred to a 'target level of safety' (TLS). The TLS varies greatly from one ITS service to the other. For example, the TLS for *collision avoidance* (CA) service (a safety-of-life application) should be higher than that of for *emergency response* (ER) service (DOT, 2004; FRP, 1999). Therefore, the positioning requirements for CA are high compared with those for ER. Normally, highly accurate positioning information is essential for SOL ITS services so that the precise location of the vehicle can be determined to avoid any collision (DOT, 2004). Moreover, the values of alert limit and time-to-alert should also be set low (i.e., high requirements) so that drivers can take appropriate actions to avoid a possible collision.

2.3.5 Type of operation (ToO)

For land transport, type of operation can generally be categorised into fixed route operation and variable route operation. Normally, transit and rail operations fall under the fixed route category. The examples of ITS services for transit and rail include *bus arrival information at bus stops*, *bus priority at junctions*, *public transport maintenance and management and transit emergency response*. For the fixed route operation, users position is located on a predefined route. Any small error in positioning data may not affect the system performance severely. The RNP requirements for fixed route services are normally low. On the other hand, ITS services that perform on roadways which are non fixed route services - examples include *navigation and route guidance*, *variable road user charging*, *commercial fleet management* and *vehicle based collision avoidance* - require high accuracy, integrity, continuity and availability of positioning information compared to fixed route operations.

As mentioned before, each of the ITS services has different RNP requirements. The criteria to determine the navigation performance parameters vary across ITS services. For a few ITS services such as *emergency vehicle management*, *vehicle based collision notification and warning system*, *vehicle-based collision avoidance*, safety issues are main criteria. For few other services such as GPS based VRUC and electronic parking payment, commercial issues are predominant factors for deriving RNP parameters. The criteria to derive RNP parameters for most of the existing location-based transport services, which require navigation and positioning information, are depicted in Table 2.2.

Table 2.2: Criteria to Derive RNP Parameters for location-based ITS

ITS User Groups	User services	Criteria to Derive RNP
Advanced Traffic Management System	Navigation and Route guidance	SPR, OE
	En-route driver information	SPR, OE
	Traveller services information	SPR, OE
	Incident management	OE,
	Traffic control	OE,
	On-board emissions monitoring (OEM)	SPR
Advanced Public Transport and Operations	Public transport management	ToO, SPR
	En-route transit information	ToO, SPR
	Demand responsive transit	ToO
Electronic Payment System	Variable road user charging	CI, OE
	Electronic parking payment	CI
	Electronic toll collection	CI
Commercial Vehicle Operations	Commercial fleet management	SPR, OE
	Hazardous material incident response	OE
	On-board safety monitoring	SI
Emergency Management Systems	Emergency vehicle management	SPR, SI, OE
	Emergency notification / personal security	SPR, SI, OE
	Disaster response and management	OE, SI
Advanced Vehicle Safety System	Vehicle based collision notification and warning system	SI, OE
	Vehicle-based collision avoidance	SI, OE
Other Services	Accident survey	SPR
	Maintenance and management system	SPR
	Road weather management	OE

For example, RNP requirements for *navigation and route guidance* are primarily driven by system performance requirements (SPR) and operational environments (OE).

2.4 RNP values in the literature

The level of accuracy required depends upon the quantity, type and quality of guidance information provided to users. If a route guidance system is required to provide information such as roadway signing (e.g. stop signs, sharp curve, wet

pavement, warning messages whenever driver exceeds safe speed, etc.) along with the usual turn-to-turn instructions on how to reach a destination, then the required positioning accuracy should be high. The accuracy requirement for the *navigation and route guidance* service varies from 1m to 20m (95%) (DOT, 2004; Quddus, 2006; FRP, 1999; Ochieng and Sauer, 2002).

AL and TTA for navigation and route guidance vary from 2m to 20m and from 1sec to 15 sec respectively (DOT, 2004; Quddus, 2006). These parameters are set in such a way that there should not be any adverse effect if the system fails to alert the user, and if the performance goes below the expected level. Moreover, these RNP requirements for *navigation and route guidance* systems also depend on operational environments; if it is a critical operational environment such as dense urban area it demands high requirements. Because in complex road networks, where roads are in close proximity to each other, small positioning error or failure to alert users keeps the users in wrong road which creates more confusion.

The recommended availability requirements (by DOT, 2004; Quddus, 2006; FRP, 1999; Ochieng and Sauer, 2002) for *navigation and route guidance* vary from 95% to 99.7% meaning that the total outage of the system is from about 4 minutes to 70 minutes in a day.

RNP parameters accuracy, integrity, continuity and availability for various ITS services discussed in the literature are illustrated in Table 2.3. It is noticeable from Table 2.3 that the requirements for only two RNP parameters (*accuracy* and *availability*) are fully reported in the literature for the case of land transport. There is partial information on system integrity and continuity. This suggests that the impact of *integrity* (i.e., warning users when the system is not trustable) and *continuity* (i.e., the consequence due to loss of system integrity and continuity) on the performance of a system is under development for road transport systems.

Table 2.3: Required Navigation Performance (RNP) Parameters for ITS

ITS user group	Source	Accuracy in metres (95%)	Integrity			Continuity risk (Per 1 hour)	Availability (%)
			Alarm limit (metres)	Time to alarm (Seconds)	Integrity risk (Per 1 hour)		
Highway							
Navigation and Route Guidance	DOT (2004)	1 to 20	2 to 20	>=5	--	--	>95
	Quddus (2006)	5 to 20	--	1 to 15	--	--	99.7
	FRP (1999)	5 to 20	--	--	--	--	--
	Ochieng and Sauer (2002)	5	--	1	--	--	99.7
	Feng and Ochieng (2007)	5-20	7.5-50	10	10^{-6}	10^{-5}	99.7
Automated Vehicle Identification	DOT (2004)	1	3	>=5	--	--	99.7
	FRP (1999)	30	--	--	--	--	--
	Sheridan (2001)[1]	10 to 50	50	300	--	--	>99.0
Automated Vehicle Monitoring	Quddus et al. (2007)[2]	5	--	--	--	--	--
	DOT (2004)	0.1 to 30	0.2 to 30	5 to 300	--	--	>95.0
	FRP (1999)	30	--	--	--	--	--
	Feng and Ochieng (2007)	30	75	10	10^{-6}	10^{-5}	99.7
Emergency Response	DOT (2004)	0.3 to 10	0.5 to 10	Near zero	--	--	99.7
	Ochieng and Sauer (2002)	5	--	1	--	--	99.7
	Quddus (2006)	5 to 10	--	1 to 5	--	--	99.7
Collision Avoidance	DOT (2004)	0.1	0.2	5	--	--	99.9
	FRP (1999)	1	--	--	--	--	--
	Quddus (2006)	1	--	1 to 15	--	--	99.7
	Feng and Ochieng (2007)	1	2.5	1	10^{-7} per case	10^{-5}	99.7
Accident Survey	DOT (2004)	0.1 to 4	0.2 to 4	30	--	--	99.7
Transit							
Vehicle Command and Control	DOT (2004)	30 to 50	--	--	--	--	99.7
	Quddus (2006)	30 to 50	--	1 to 15	--	--	99.7
	FRP (1999)	30 to 50	--	--	--	--	--
Automated Voice Bus Stop Announcement	DOT (2004)	5	--	--	--	--	99.7
	Quddus (2006)	5 to 10	--	1 to 15	--	--	99.7
Emergency Response	DOT (2004)	75 to 100	--	--	--	--	99.7
	FRP (1999)	75 to 100	--	--	--	--	--
Data Collection	DOT (2004)	5	--	--	--	--	99.7
	FRP (1999)	5	--	--	--	--	--

[1] Automated Vehicle Identification for Road tolling and taxation purpose.

[2] Vehicle monitoring for GPS based VRUC

As shown in Table 2.3, the level of accuracy, integrity and availability required for different ITS applications vary significantly. For instance, the values of *accuracy* (1m to 20m, 95% of the times) and *availability* for vehicle *navigation and route guidance* are high (i.e., low requirements) compared with those of *collision avoidance* service. This is because safety is the most critical criterion for the collision avoidance service. The requirements of integrity risk and continuity risk for land navigation are still under development (DOT, 2004).

Moreover, the RNP requirements for each ITS service is given in a range. For example, the accuracy requirements for vehicle identification vary from 1m to 50m (95%) (DOT, 2004; FRP, 1999; Sheridan, 2001). The automatic vehicle identification is a necessary component of various ITS services and applications such as *electronic parking system* (EPS), *parking facility management* (PFM), *electronic toll collection* (ETC), *emergency vehicle management* (EVM), and *automatic bus arrival announcements* (ABAA). A highly accurate positioning data is required for electronic parking (EP) since it is essential to identify the presence or absence of a vehicle in a tightly spaced parking bays. The horizontal accuracy requirements for EP may be 5m (95%); because it has to locate a vehicle on a parking bay. For ECT and EP performance parameters are mainly driven by commercial issues. In case of EVM system, RNP parameters are driven by safety issues which demands high requirements. On the other hand, ABAA at bus stops which enables the operator to find out the location of the vehicle on a network may not require high accurate positioning data; 40 to 50 m (95%) accuracy may be sufficient for this purpose.

2.5 Target ITS applications

RNP parameters for various ITS services are mainly driven by either safety or commercial or system operational and performance issues. If they are driven by the safety issue, then the requirements are high. This is due to the fact that such ITS services are safety-of-life (SOL) critical applications. If the RNP parameters

for an ITS service are driven by the operational issue, then the requirements are relatively low.

Unlike aviation and marine transport, land navigation vehicles' position is always referred-to on a spatial road network map. In aviation and maritime, the positioning accuracy is measured with respect to the distance between true position and the measured position. In the case of land vehicle navigation, firstly it is necessary to identify a road segment (from a set of candidate road segments) on which a vehicle is travelling and then the vehicle position on that road segment. Here, in the RNP parameter *accuracy* is categorised as the percentage of correct link identification and horizontal positioning accuracy. Due to a lack of accurate (true) vehicle location data, researchers usually measure whether the vehicle is located on a correct road segment on which vehicle is travelling. This is often represented as percentage of correct road segment identification (Pyo et al., 2001; Bouju et al., 2002; Yang et al., 2003; Syed and Cannon, 2004; Wu et al., 2007). Very few authors have measured the accuracy of the navigation system with respect to both percentage of correct link identification and the accuracy in distance (Quddus 2006; Yu et al., 2006; Taylor et al., 2006).

The vehicle navigation module needs to be further improved to support location-based real-time transport services (White et al., 2000; Quddus et al., 2003; Ochieng et al., 2004a; Jabbour et al., 2008; Velaga et al., 2009). As discussed in section 1.1, further improvement in map-matching algorithms and development of an integrity method is a viable approach in order to enhance the vehicle navigation system. In this research, the ITS services that require about 98 percent correct link identification[3] with positioning accuracy of 5 to 10 m are targeted.

[3] RNP requirements on percentage of correct link identification is not available in the literature; but generally the performance of any MM algorithm is reported with respect to percentage of correct link identification along with horizontal accuracy.

These include:

(a) Liability-critical applications: electronic toll collection, electronic parking payment and GPS based variable road user charging;
(b) System performance critical applications: navigation and route guidance, public transport management, automatic bus arrival announcements;
(c) Commercial applications: fleet management, commercial vehicle administrative processes and electronic clearance; and
(d) Safety-critical applications: emergency vehicle routing, incident and accident management.

While enhancing a map-matching algorithm or integrity method it should be noted that, in order to support the real time ITS applications, both the MM algorithm and the integrity method, which will be developed in this research, should be generic (not specific for a particular ITS application), simple, fast and easy to be implemented by the industry. Moreover, the validation of the developed algorithm and integrity method using a higher accuracy reference (truth) of the vehicle trajectory obtained from a carrier phase GPS observables is essential. This will indicate the reliability of both the enhanced algorithm and the integrity method.

Chapter 3

Review of Topological Map-Matching Algorithms

3.1 Introduction

This chapter introduces map-matching (MM) algorithms along with their constraints and limitations. This is followed by a detailed review of existing topological MM algorithms which classifies them as in sections: (1) non weight based real time (on line) algorithms (2) weight based real time algorithms and (3) off-line algorithms. The performance of these topological map-matching algorithms in terms of correct link identification and horizontal accuracy is also presented. Then a brief discussion on how to further improve existing topological map-matching algorithms is provided.

Map-matching is the process of identifying a vehicle's position on a digital map. The purpose of MM is firstly to identify correctly the link on which a vehicle is travelling and secondly to determine the vehicle's position on the selected link (Zhao, 1997; Ochieng et al., 2004b; Chen et al., 2005). A MM algorithm integrates navigation data from positioning sensors and spatial road network data from a digital map to determine the location of a vehicle on a road segment. There are different ways to accomplish the purpose of a MM algorithm. MM algorithms can generally be classified into four categories (Quddus et al., 2007a): geometric, topological, probabilistic and advanced MM algorithms.

A MM algorithm that uses only geometric information (the shape of the curve of the road segment) is known as a geometric MM algorithm (gMM) (Kim et al., 1996; Bernstein and Kornhauser, 1998; White et al., 2000; Quddus et al., 2007a). Geometric MM algorithms are improved by incorporating historical data (such as the previously matched road segment), vehicle speed and topological information about 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; Quddus et al., 2003; Li et al., 2005; Quddus et al., 2007a). Probabilistic MM algorithms use a probability theory in the identification of the correct road segment on which the vehicle is travelling (Ochieng et al., 2004b). 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; Kim and Kim, 2001; Yang et al., 2003; Syed and Cannon, 2004; Quddus et al., 2006a; Zhang and Gao, 2008a; Sohn, 2009).

3.2 Geometric Map-matching Algorithms

The earliest MM algorithms, developed in the 1990s, used geometric information (shape of the curve) of road network to locate the vehicle on a GIS map (e.g. Krakiwsky et. al., 1988; Tanaka, 1990; Collier, 1990; Bernstein and Kornhauser, 1998; Kim et al., 1996; White et al., 2000). These algorithms are referred to as geometric MM algorithms. They are further categorised into: (1) point-to-point MM, (2) point-to-curve MM and (3) curve-to-curve MM algorithms (Bernstein and Kornhauser, 1998).

3.2.1 Point-to-point MM

In a point-to-point MM algorithm, a position (x and y coordinates) of a moving vehicle obtained from a positioning system is matched to the closest node or shape point of a road segment (Bernstein and Kornhauser, 1998). A node is a

point, represented by a pair of longitude and latitude coordinates (*x* and *y* coordinates), at which a link begins or terminates (i.e., the beginning and ending of a road segment) (Wu et al., 2007). A link may be a straight line segment (a straight section of a road) or a general curve, having intermediate points (shape points) along the link. An example using a hypothetical road network is shown in Figure 3.1.

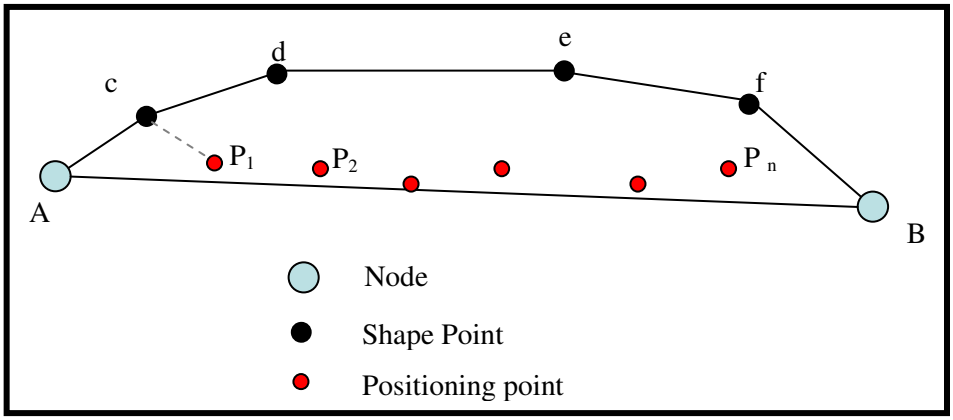


Figure 3.1: Problems with Point-to-point MM algorithm

In the above figure, points A and B are nodes, points c, d, e and f are shape points; and P_1, P_2, …, P_n are positioning points. A point-to-point MM algorithm, which is also known as the nearest node searching method, calculates the distance between the positioning point and node or shape points and selects the closest one (Krakiwsky et. al., 1988; Bernstein and Kornhauser, 1998). For positioning point P_1, the closest shape point is 'c' and for P_2 the closest one is 'd'. Based on the point-to-point algorithm, the map-matched route is determined as link c-d, d-e and e-f. However, the vehicle is actually travelling on link A-B suggesting that the algorithm wrongly identifies the travelling route. This algorithm is easy to implement and fast, but it has the following characteristics:

1. Snapping positioning points to the nearest node or shape point.
2. Positioning fixes are treated individually, no historical information is used.
3. No topological information, such as road connectivity, is used.

4. Additional data, such as vehicle speed and heading, are not used.

The above characteristics in the map-matching process may lead to the following problems:

1. A lack of certainty about the road segment on which a vehicle is travelling.
2. Mis-matching at junctions and in dense urban areas where roads are close to each other.
3. Sudden switching of map-matched vehicle position from one road segment to the other.

These problems are illustrated in Figure 3.2 and discussed below.

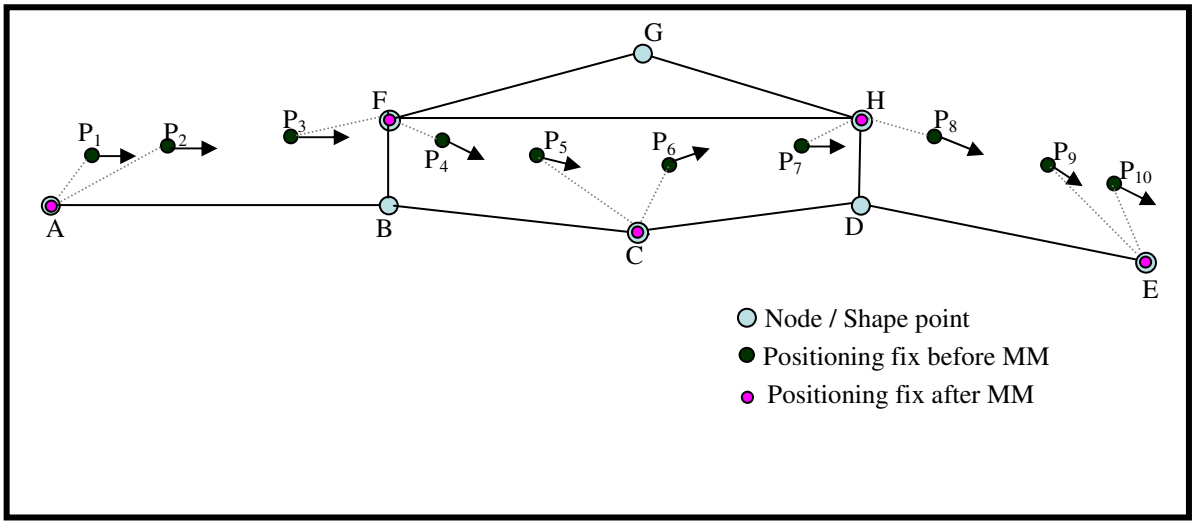


Figure 3.2: Mis-matches by the Point-to-point MM algorithm

A lack of certainty about the road segment on which a vehicle is travelling: In a hypothetical road network shown in Figure 3.2, points P_1, P_2, P_3,….., P_{10} are positioning points of a vehicle obtained from positioning sensors. A, B, C,…., H represent node/shape points of the road network. According to the point-to-point MM algorithm the positioning fixes are snapped to the nearest node or shape points. For position points P_3 and P_4 the nearest node is F and similarly for P_7

and P_8 the nearest node is H. The point-to-point algorithm assigns these positioning fixes to F and H. There are three routes, *route 1*: F-G-H and *route* 2: F-H (straight road) *route 3*: F-B-C-D-H, that connect the node F and H. A Point-to-point MM will not show the route which the vehicle actually travelled. This problem is due to snapping positioning points to the node/shape points.

Mis-matching at junctions and in dense urban areas: In Figure 3.2, the vehicle actually travelled on links A-B, B-C, C-D and D-E (i.e., through nodes A, B, C, D and E). Point-to-point matching shows the vehicle travelling through nodes A, F, C, H and E, because the positioning points P_3, P_4, P_7 and P_8 are nearer to junctions F and H compared to junctions B and D.

Sudden switching from one road segment to other road segments: In Figure 3.2, initially, the positioning fixes P_1 and P_2 are assigned to node A. The following positioning fixes (P_3 and P_4) are assigned to node F, though there is no direct link connecting these two nodes (A and F). This mis-matching is due to the fact that the point-to-point algorithm treats all positioning fixes individually and does not consider historic MM information.

The point-to-point algorithm performance critically depends on the way the road network is digitised (the way in which shape points are used in the network digitisation). In Figure 3.3 a modified/re-digitised version of road network of Figure 3.2 is shown. Additional shape points such as a, b, c, d, e, f, are incorporated between nodes A, B, C, D, E. Now the point-to-point algorithm is capable of identifying the correct route (i.e. route A-B-C-D-E) on which the vehicle actually travelled. The greater the number of shape points between two nodes the more likely the actual travel path will be matched. One could argue that just increasing the number of shape points in the network could solve the problem. However, modifying road network maps nationally is a costly and time consuming process.

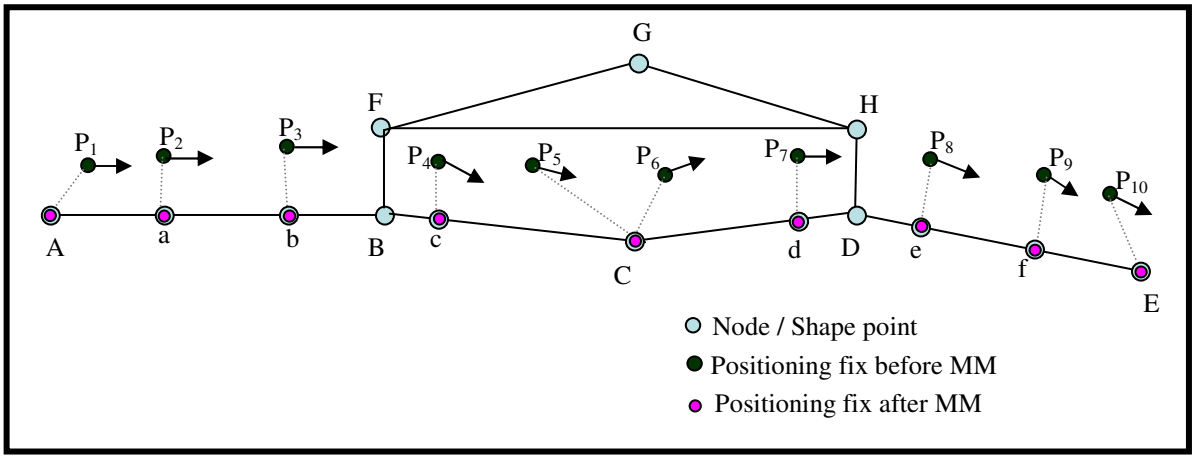


Figure 3.3: Point-to-point MM algorithm performance with more shape points

3.2.2 Point- to- curve MM algorithm

This algorithm identifies the link that is the closest to the positioning fix as the correct link (Tanaka, 1990; Bernstein and Kornhauser, 1998; Kim et al., 1996). In Figure 3.1, link A-B is the closest one for positioning points P_1, P_2 …, P_n, and this link is selected as the true link for these positioning fixes. The perpendicular projection of a positioning fix on the selected link gives the vehicle position on that link.

The performance of point-to-curve algorithm is better than that of the point-to-point approach as it gives the vehicle location on a link, but still the algorithm has limitations. These include:

1 Positioning fixes are treated individually; no historical/past matching information is used.
2 Topological information (such as road connectivity) and additional data (such as vehicle speed and heading) are not used.

The above limitations in the map-matching process may lead to the following problems:

1. Mis-matching at junctions and in dense urban areas where roads are close to each other.
2. Sudden switching from one road segment to the other road segment.

These problems are explained in Figure 3.4.

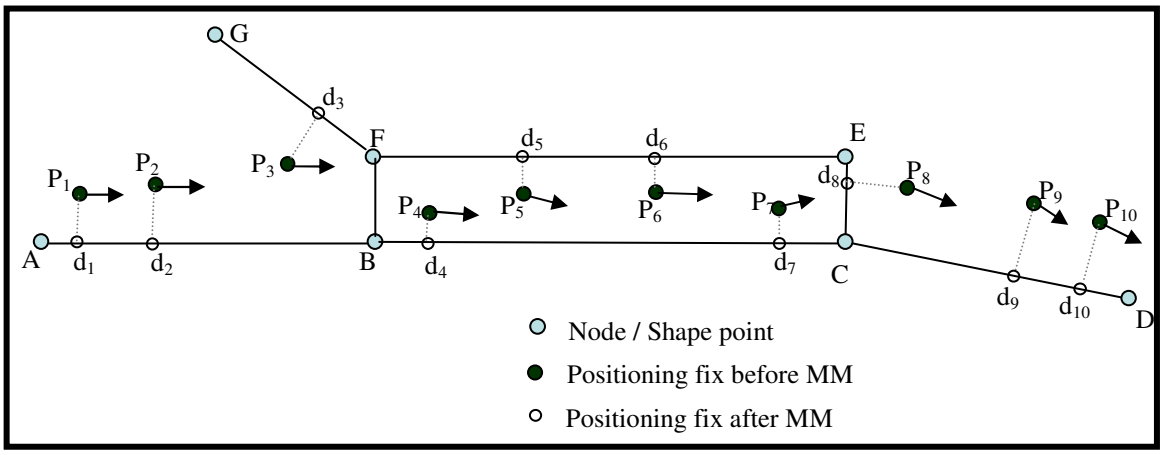


Figure 3.4: Point-to-curve algorithm performance

Mis-matching at junctions and in dense urban areas: In Figure 3.4, points P_1, P_2,….., P_{10} are the positioning points, points d_1, d_2,…., d_{10} are the corresponding map-matched points. The vehicle actually travels on link A-B, B-C and C-D. The positioning fixes P_5 and P_6 are assigned to link E-F, because these positioning points are slightly nearer to link E-F than link B-C. At junction C, position fix P_8 is almost equidistant from links C-D and C-E, and slightly nearer to link C-E. Based on the minimum distance this positioning point, P_8, is assigned to link C-E. Here, two parallel routes A and D, are *route 1*: A-B-C-D, *route 2*: A-B-F-E-C-D. The algorithm results show that the vehicle travelled on both routes, though the vehicle actually travelled on route A-B-C-D.

Sudden switching from one road segment to other road segment: The positioning fixes P_1 and P_2 are assigned to link A-B. The next positioning fix P_3 is nearer to link F-G and assigned to d_3. The result shows that the vehicle suddenly switches from link A-B to F-G.

3.2.3 Curve-to-curve MM algorithm

The curve-to-curve algorithm considers positioning fixes simultaneously and constructs piece-wise linear curves (Bernstein and Kornhauser, 1998; Kim et al., 1996; White et al., 2000). The algorithm compares this piece-wise vehicle's trajectory curve with the road segments that are passing through the nearest node. The road segment that is the nearest to the curve formed by positioning fixes, is considered as the correct road segment on which vehicle travels. The perpendicular projection of positioning fixes on the selected link gives the vehicle position on that link.

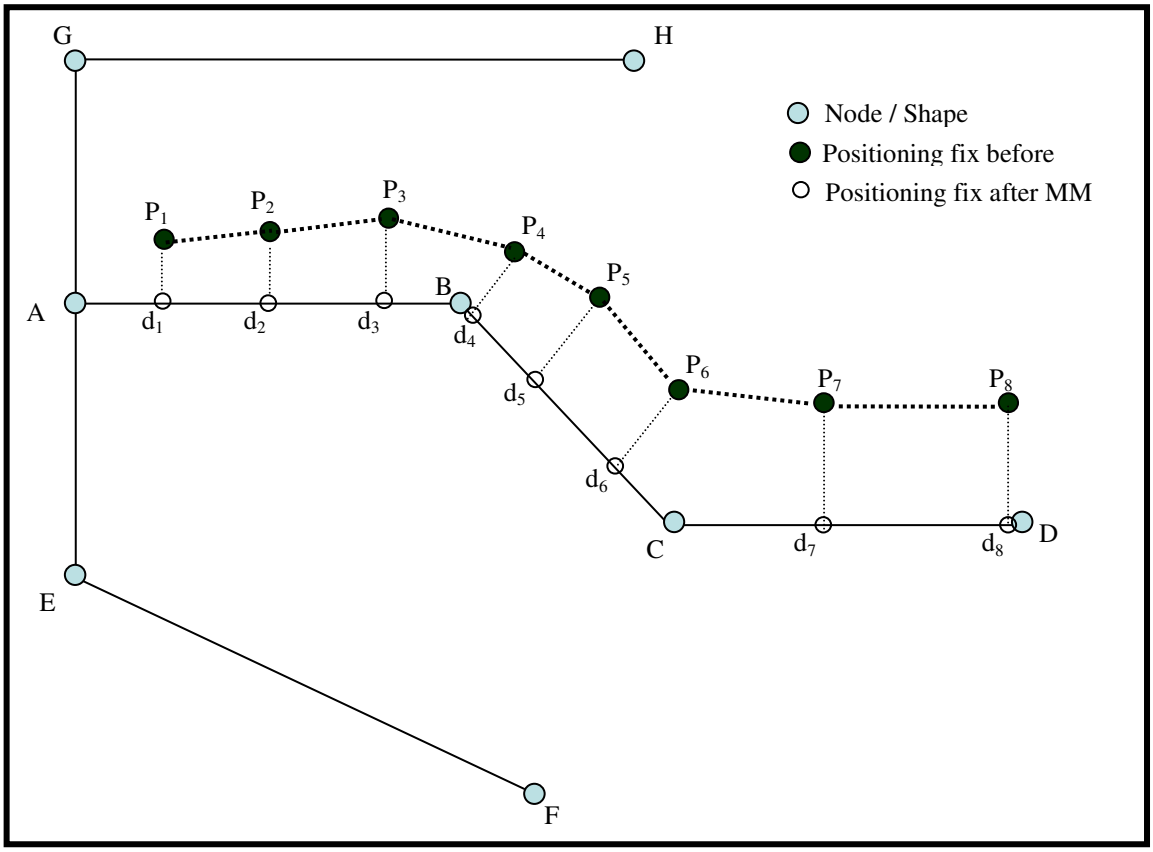


Figure 3.5: Curve-to-curve MM algorithm

As seen in Figure 3.5, for the first positioning point P_1, the nearest node is 'A'. The piece-wise road segments that pass through node 'A' are: (1) A-B-C-D; (2) A-E-F and (3) A-G-H. The algorithm constructs a piecewise vehicle trajectory

curve by using positioning fixes P_1 to P_8, then the vehicle trajectory curve is compared with the road segments that pass through the nearest node/shape point 'A'. Among the three candidate road segments, the piecewise road segment, A-B-C-D, is identified as the nearest to the vehicle trajectory curve. The perpendicular projection of positioning points on road segment A-B-C-D gives the vehicle position.

The major limitations of the MM algorithm are:

1. Algorithm performance depends on the point-to-point algorithm (i.e., the nearest node search) that identifies a set of piecewise candidate road segments.
2. The algorithm cannot provide real-time vehicle location information.
3. Topological information and additional data, such as vehicle speed, heading data are not used.

The above limitations in this map-matching process may lead to the following problem:

1. The algorithm is quite sensitive to outliers, sometimes it gives an unexpected result.

As shown in Figure 3.6, a vehicle travels on road segment A-B-C-D. For the first positioning fix P_1, the nearest node point is 'G'. Piece-wise road segments that pass through 'G' are: G-I and G-H. Among these two road segments, segment G-I is identified as the nearest segment and the vehicle is assigned to that link. Here, the true road segment on which the vehicle actually travels (i.e., A-B-C-D) is eliminated from candidate piecewise road segments. The curve-to-curve algorithm depends on the point-to-point matching and sometimes, keeps the vehicle on the wrong road segment.

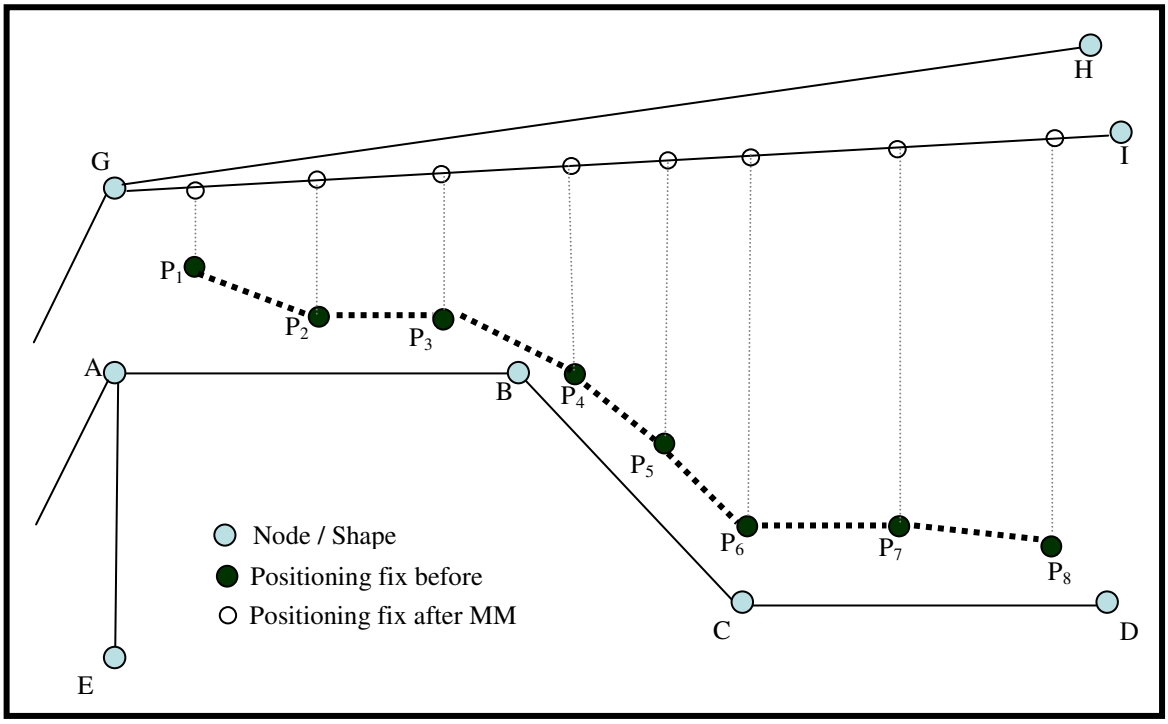


Figure 3.6: Curve-to-curve algorithm performance

Algorithms that use only geometric information are the simplest and fastest. However, in many cases (e.g. matching at junctions and between two parallel roads) geometric MM algorithms may yield undesirable results. Further improvement of the geometric MM algorithms can be achieved by considering additional information in correct road segment identification. An example of such algorithms is a topological map-matching algorithm.

3.3 Topological Map-matching algorithms

The topological map-matching uses additional information including historical/past matching information, vehicle speed, vehicle turn restriction information and topological information of the spatial road network (e.g. link connectivity) in addition to road geometry (Greenfeld, 2002; Quddus, 2006; Xu et al., 2007; Pink and Hummel, 2008). This additional information enables tMM to outperform geometric MM algorithms. An example of a topological map matching output is shown in Figure 3.7.

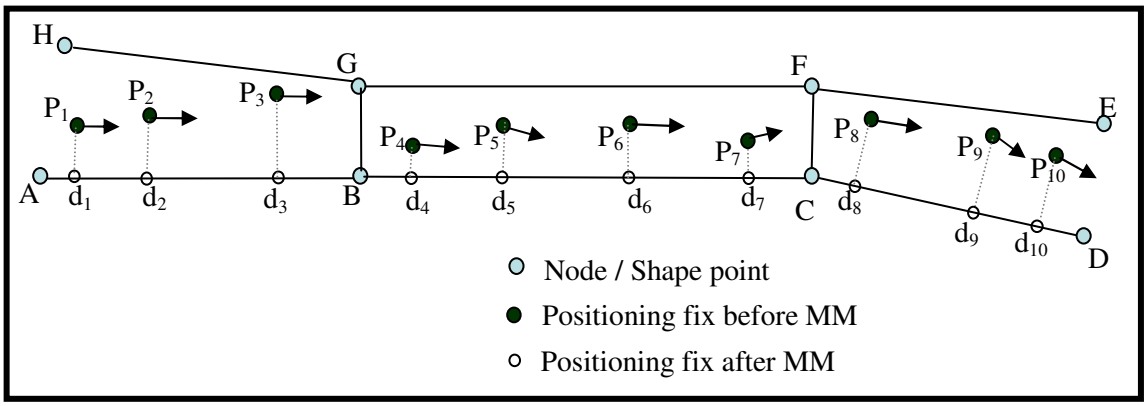


Figure 3.7: Topological Map-matching output

In Figure 3.7, the vehicle is actually travelled on road segment A-B, B-C and C-D. If map-matching is carried out by a geometric algorithm (example, point-to-curve algorithm), the corresponding road segment for the positioning points P_5 and P_6 would be link G-F, and for the positioning points P_8, P_9 and P_{10}, the correct link would be F-E. This is because, the positioning points P_5 and P_6 are nearer to the link G-F, and P_8, P_9 and P_{10} are nearer to the link E-F. A topological map-matching algorithm will identify the links correctly. For instance, at junction C, the positioning point P_8 is nearer to the link E-F and also coincides with the vehicle movement direction, but it is not directly connected to the previously travelled road link (i.e., B-C). Therefore, the algorithm selects link C-D as the correct link. A similar scenario applies to the other successful positioning points (P_5, P_6, P_8, and P_9) in Figure 3.7.

The advantages of the tMM algorithms over geometric algorithms are:

1. Positioning fixes are not treated individually; historical/past matching information is used.
2. The correct link identification process is more logical.
3. Topological and additional information, such as vehicle speed, heading are used in the correct link identification process.

Topological MM algorithm performance is better than that of the geometric algorithms, but, still there are some flaws:

1. Sometimes a tMM algorithm fails in identifying the correct road segment at 'Y' junctions, roundabouts, and dense urban areas.
2. Problems with the initial map-matching.

Problems with the tMM at a complex road network: although a tMM uses more information than a geometric algorithm; sometimes, the tMM algorithm still cannot keep the vehicle on the correct road segment. This is particularly so in dense urban areas, at roundabouts and junctions. An example of mis-matching at a Y junction is shown in Figure 3.8.

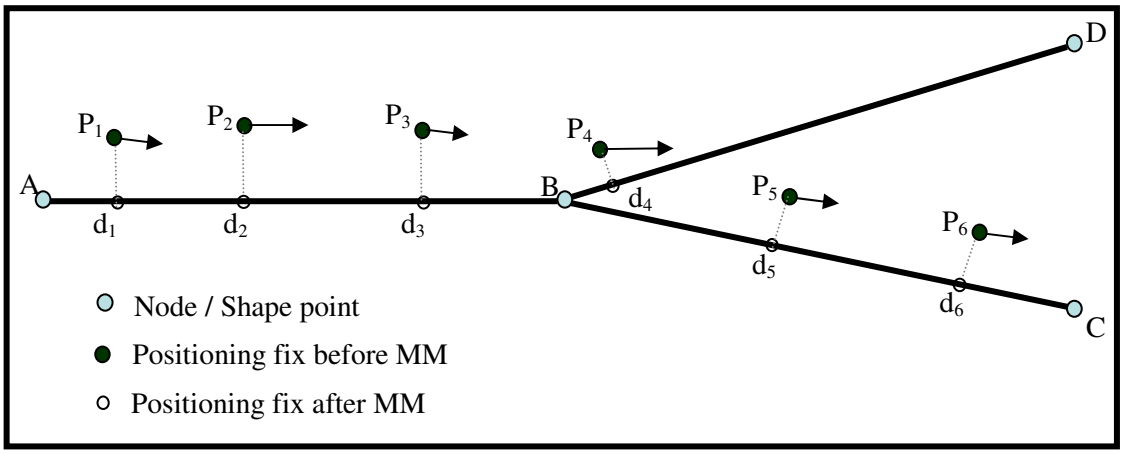


Figure 3.8: Mis-matching at a 'Y' Junction

In Figure 3.8, a vehicle travelled on road segment A-B and B-C. The positioning fixes (P_1, P_2, and P_3) are assigned to link A-B. For the positioning fix (P_4), which is at junction B - based on the vehicle movement direction (heading), nearness to the road segment link, connectivity to the previously travelled to road segment, and vehicle speed - the algorithm selects B-D as the correct link. In this typical scenario, the tMM algorithm fails to keep the vehicle on the correct road segment.

Problems with the initial map-matching: The initial map-matching is a process of finding the correct road segment for the first positioning fix (Greenfeld, 2002; Quddus et al., 2003). As explained above, the tMM algorithm considers the previous history (such as previous road link on which a vehicle travelled). The subsequent map-matching (i.e., MM after the first positioning point), therefore, depends on the output of the initial map-matching. In general, for the initial map-matching, existing tMM algorithms select all the links that are connected to the nearest junction for the first fix, as the candidate road links. Then the algorithm chooses the correct link from these candidate links (Greenfeld, 2002; Quddus et al., 2003; Taghipour et al., 2008).

In Figure 3.9, P_1, P_2 ….P_5 are positioning points. For first positioning point (P_1) the nearest node point (junction) is node c. Links passing through node c are c-b, c-d, c-e. Algorithms consider these three links to be the candidate links. Among these three candidate links, link c-d is nearer to the first GPS point (P_1) and the vehicle movement direction and link c-d direction is also similar. Based on vehicle heading and proximity, link c-d is identified as the correct link for positioning point P_1. The following positioning points (P_2, $P_{3....}P_5$) are assigned to the same link c-d. But the true link on which the vehicle was travelled is link b-a.

In the candidate link identification for initial map matching the algorithm, sometimes, fails to consider the true link, on which vehicle travels. If the algorithm selects the wrong link for the first positioning fix, this may affect the subsequent map-matching. In Figure 3.9, for the subsequent positioning points (P_2, P_3, P_4 and P_5) the algorithm identifies link c-d as the correct link, but the vehicle is on link b-a.

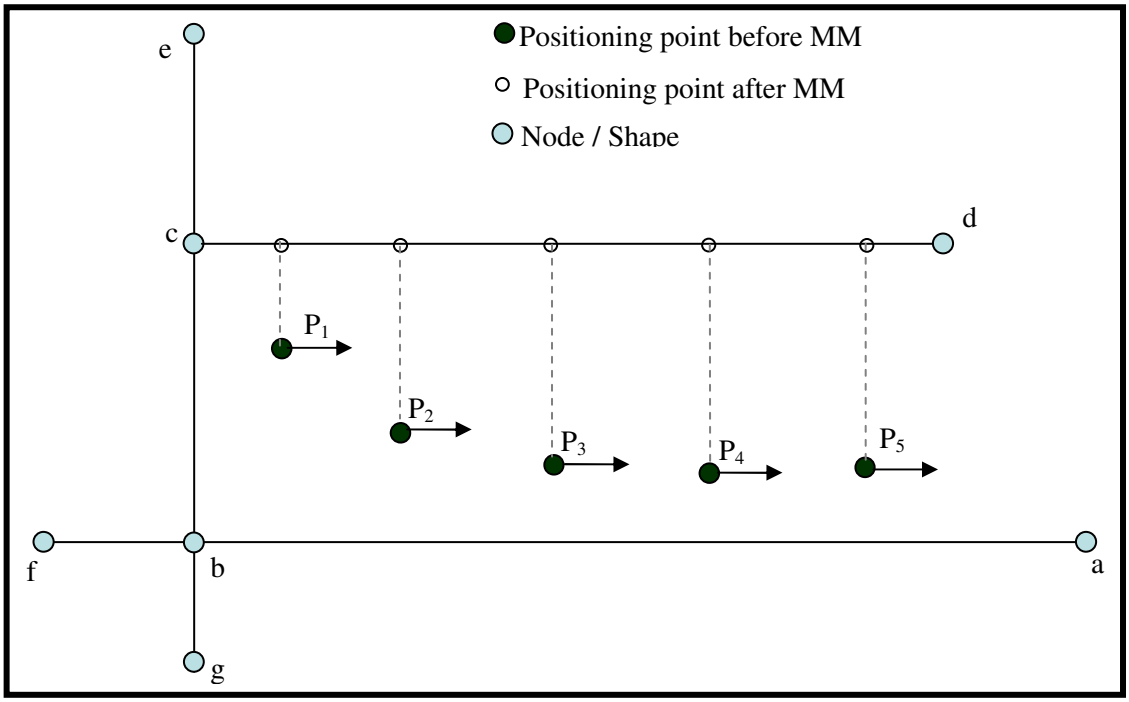


Figure 3.9: Errors in initial MM process

3.4 Probabilistic map-matching algorithms

The main advance in probabilistic MM algorithms is the selection of a set of candidate links for a positioning point. A tMM algorithm identifies the correct link among the links that are connected to the nearest junction from that positioning fix. This process may not be reliable in dense urban areas where junctions are very close to each other. In the probabilistic algorithm, initially an error bubble (shaped either as an ellipse, a rectangle, a square or a circle) is drawn around a positioning fix; and the links that are inside and around the error bubble are considered as the candidate links for that positioning fix. The probabilistic algorithm can therefore solve some problems with the initial map matching process, even so map-matching errors in dense urban areas are inevitable.

The process in which the error bubble is derived and the candidate links selected that are both inside and just touching the bubble is explained in Zhao (1997),

Kim et al. (2000) and Pyo et al. (2001). In this research, an error circle is used to identify a set of candidate links. A detailed description of the process is therefore provided in section 6.3.1.

3.5 Advanced map-matching algorithms

Advanced algorithms use more refined concepts such as: Extended Kalman Filter (EKF) (e.g. Krakiwsky, 1993; Tanaka et al., 1990; Jo et al., 1996; Kim et al 2000; Torriti and Guesalaga, 2008;), Bayesian interference (e.g. Pyo et al., 2001; Smaili et al., 2008), belief theory (e.g. Yang et al., 2003; Najjar and Bonnifait, 2005; Nassreddine et al., 2008), fuzzy logic (e.g. Zhao, 1997; Kim and Kim, 2001; Syed and Canon, 2004; Quddus et al., 2006a; Su et al., 2008; Zhang and Gao, 2008b) and artificial neural networks (e.g. Ding et al., 2007; Su et al., 2008). The difficult part of any map-matching process is to detect the correct road segment from a set of candidate segments. In most algorithms, the advanced techniques are used in the correct link identification process.

Among these MM algorithms (geometric, topological, probabilistic and advanced) an aMM algorithm, that uses more refined concepts, outperforms other MM algorithms. However, these aMM algorithms require more input data and are relatively slow and difficult to implement compared to a tMM algorithm (Velaga et al., 2009). Because these advanced algorithms are slow, sometimes, they may not be able to support real-time transport applications. 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. However, there are a number of constraints and limitations in existing tMM algorithms. Once such limitations are addressed, it is expected that the performance of a tMM algorithm may be comparable to that of a pMM algorithm or aMM algorithm. The remainder of this chapter concentrates on detail review and performance of topological map-matching algorithms. The

following section provides a critical review of existing topological MM algorithms and their performances.

3.6 Review of Topological MM algorithms

Before starting the review of existing tMM algorithms, the process by which a general topological MM algorithm works is provided :

Step 1: Identify a set of candidate links for the first positioning point (P_n).

Step 2: Identify the correct link among the candidate links using vehicle heading, speed, turn restriction and road network topological information such as link connectivity.

Step 3: Determine the vehicle position on the selected link.

Step 4: Check whether next positioning point (P_{n+1}) is near to a junction or not.

Step 5: If positioning point (P_{n+1}) is not near to a junction, assign the current position fix (P_{n+1}) to the previously map-matched link.

Step 6: If positioning point (P_{n+1}) is near to a junction, repeat steps 1, 2 and 3.

Step 7: Repeat the above procedure until the vehicle stops.

A MM algorithm can be performed in two ways: on-line or off-line (Jagadeesh et al., 2004). On-line MM algorithms determine the real-time vehicle position during a trip (Basnayake et al., 2005). Off-line matching finds the overall route of the vehicle after the trip is over (Jagadeesh et al., 2004). Most ITS services, such as route guidance, vehicle-based collision avoidance, emergency vehicle management, fleet management and bus priority at junctions require real-time vehicle positioning information. However, for a few ITS services such as road maintenance and management systems and GPS based accident surveys, real-time vehicle position may not be required.

Various topological MM algorithms have been developed for location-based ITS applications. Different authors have used the topological information at various

levels. For instance, White et al. (2000) and Li and Fu (2003) used topological information to identify a set of candidate links for each positioning point. Srinivasen et al. (2003) and Blazquez and Vonderohe (2005) used topological information to check the map-matched point after geometric (point-to-curve) matching. Greenfeld (2002), Quddus et al. (2003), and Taghipour et al. (2008) introduced weight-based algorithms to identify the correct road segment among the candidate segments for on-line map-matching problem. Marchal et al. (2005) developed a off-line weight based topological MM algorithm. Li et al. (2005) introduce the buffer band concept to identify candidate links for off-line matching problem. Each of these algorithms has strengths, complexities and limitations. This section discusses different techniques and methodologies used in previous research by categorising them into three groups: (1) non weight based on-line algorithms (2) weight based on-line algorithms and (3) off-line topological algorithms.

3.6.1 Non-weight based on-line algorithms

White et al. (2000) and Li and Fu (2003) used vehicle heading information (i.e., vehicle movement direction with respect to the north) to identify a set of candidate links for each position fix. If the vehicle heading is not in line with the road segment direction, then the road segment is discarded from the set of candidate segments. If the algorithm has confidence in the previous map-matched position it will use the topology (road connectivity) of network for current positioning point map-matching. That means the algorithm considers the road segments that are connected to the previous map-matched road link to identify the current road segment. Srinivasen et al. (2003) and Blazquez and Vonderohe (2005) used the point-to-curve approach for the correct link identification for all positioning points. After selecting the correct link for a positioning point, map-matched link is checked using topological information of the road network and vehicle speed and heading. This section discusses how and

to what extent topological information is used by different authors. The performance of the algorithms is also considered.

White et al. (2000) discussed some simple MM algorithms that used geometric and topological information of road network. In their study, four different algorithms were implemented and tested using field data collected in New Jersey. *Algorithm 1* found nodes that are close to the GPS fix and finds the set of arcs that are connected to those nodes. It then finds the closest of those arcs and projects the points on that arc. *Algorithm 2* is similar to algorithm 1 except that it makes use of vehicle heading information. *Algorithm 3,* is a variant of algorithm 2, adds topological (connectivity) information. *Algorithm 4,* uses curve to curve matching; firstly, it locates candidate nodes using the same techniques as in algorithm 3, then, given a candidate node, it constructs piece-wise linear curves from the set of paths that originate from that node. Secondly, it constructs a piece-wise linear curve using the GPS fixes, and calculates the distance between this curve and the curves corresponding to the network. Finally, it selects the closest curve and projects the point onto that curve.

Field tests were conducted in four pre-determined routes, in total 1.2 km in length, in New Jersey. The study concluded that percentage of correct link identification using algorithms 1, 2, 3 and 4 are 66.3%, 73.6%, 85.8% and 68.7% respectively. The best algorithm is identified as algorithm 3 which uses topological (connectivity) information for candidate link identification. The authors also concluded that these algorithms worked better when the distance between the GPS point and closest road was small; and the correct matches tend to occur at greater vehicle speeds on straight roadways.

Li and Fu (2003) developed an improved topological MM based on pattern recognition. The algorithm is divided into three different stages, namely: 1) searching mode 2) normal running mode and 3) turning mode. The first step, searching mode, is the process of searching for a road link, for the first GPS point. Candidate links are identified by comparing the link direction and the

vehicle movement direction, then the point-to-curve approach (closest road segment) is used to select the correct link out of all candidate links. In normal running mode vehicle position fixes are continuously matched to the previous map-matched link until the vehicle makes a turn. In turning mode the algorithm uses the same method as the first step that is to complete the process of searching new road link. This method is very sensitive to outliers as these may cause the determined vehicle heading to be inaccurate as GPS position fixes are scattered randomly when the speed is less than 3 m/sec. (Taylor et al., 2001 and Ochieng et al., 2004b). The other deficiency of this algorithm is not to take into account the vehicle speed and the way links are connected to each other. Unfortunately, the performance of algorithm with respect to percentage of correct link identification and 2D horizontal accuracy is not reported by the authors.

Srinivasen et al. (2003) developed point-to-curve map matching algorithm for electronic road pricing (ERP) system using enhanced road segment information. The software, named ERP2, was developed using 'ArcPad' GIS platform. The algorithm developed for the ERP2 program, firstly identifies the closest road segment. Before assigning the GPS point on the closest segment, checks are performed to see if it is feasible for the vehicle to actually be on the road segment. Two checks, a heading check (bearing difference between vehicle movement and link direction) and a turn prohibition check, are incorporated. The turn prohibition check is implemented only when the vehicle is travelling from one road segment to another segment. If a vehicle reaches a junction point, the segment identification is compared to the turn restriction table to see whether it was possible for the vehicle to move from the previous map-matched road segment to the current road segment. If a turn prohibition is detected the algorithm will search for the next nearest segment and repeat heading and turn prohibition checks again. Their results shows that the general point-to-curve algorithm can identify the true link 69.87% of the time and for the point-to-curve algorithm with topological information, this improves to 98.5%. The result shows that the second algorithm (that uses topological information) performs well. However the reported results were based on a limited data set of 242 GPS

data points collected on a university road network. The algorithm result may not be reliable in a more complex road network and larger data sample.

Blazquez and Vonderohe (2005) reviewed several approaches for solving the map-matching problem, and proposed a simple map-matching algorithm that works based on shortest paths between snapped 'differential GPS' (DGPS) data points using network topology and turn restrictions. The algorithm selects all roadways within a 'buffer circle[4]' around a DGPS data point and snaps the point to the closest roadway by determining the minimum perpendicular distance from the data point to each roadway. Subsequently, the speed using the shortest path distance between the two snapped DGPS data points using network topology and turn restrictions is compared with the distance travelled by multiplying vehicle speed and time. This is explained in Figure 3.10. It has been observed that the performance of the algorithm depends on the size of buffer bubble.

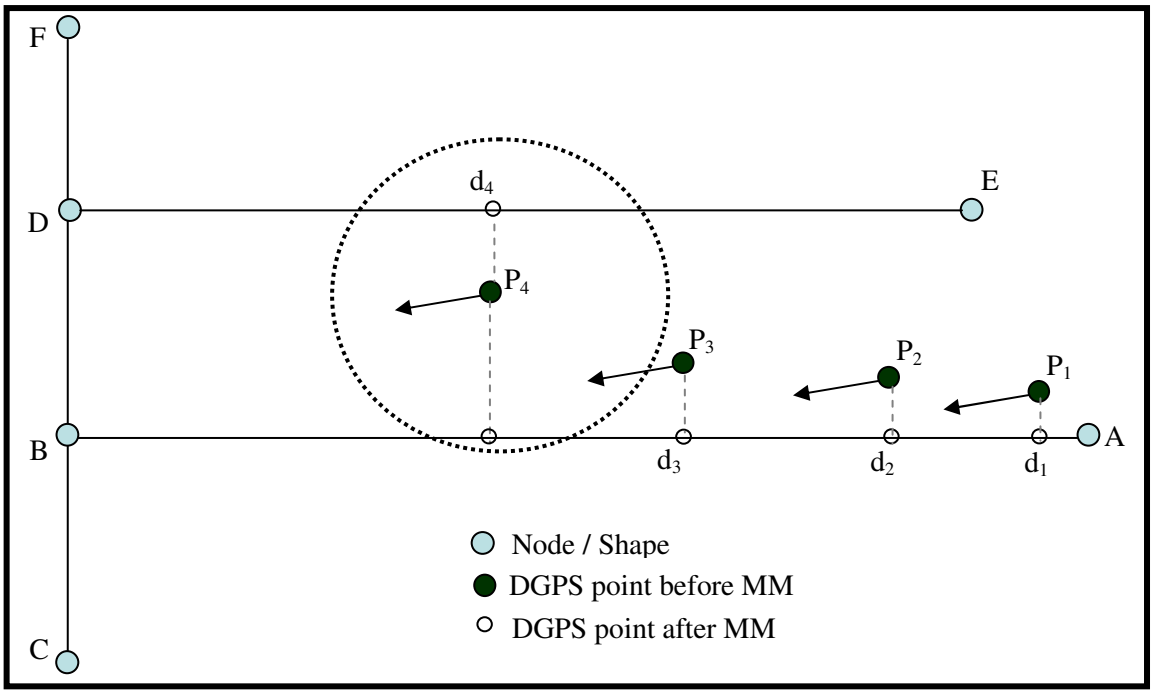


Figure 3.10: Speed checks after point-to-curve matching

[4] The buffer circle size is decided based on quality and geometry of spatial data (dGPS points and digital road network data).

In Figure 3.10, the vehicle is actually travelling on link A-B, from A to B. DGPS point P_1, P_2, P_3 are assigned on link A-B at d_1, d_2, d_3 points respectively. For DGPS point P_4 links that are inside the buffer band are A-B and D-E. The algorithm selects the nearest link D-E as the correct link and assigns DGPS point P_4 at point d_4 on link D-E. The algorithm calculates the shortest distance between d_3 and d_4 (i.e., $d_3B + BD + Dd_4$). Based on this distance vehicle speed is calculated. The snapped DGPS point d_4 is considered to be correct if similarity exists between the calculated speed and recorded vehicle speed. If the two speeds are not comparable then the algorithm searches for an alternative road link that is nearer to the DGPS point, the shortest paths are recalculated and speeds are again compared. In this example, after a second iteration the algorithm selects the correct link A-B. This algorithm achieved 94.8% correct link identification based on 600 DGPS points collected in Columbia. The authors concluded that actual link identification is sensitive to size of the buffer circle.

3.6.2 Weight based online algorithms

A weight-based topological MM approach for the correct link identification among candidate links was first developed by Greenfeld (2002). This approach was further improved and tested by Quddus et al. (2003). Taghipour et al. (2008) demonstrate the advantages of a weight-based approach, and propose modifications to the topological map-matching algorithm developed by Quddus et al. (2003). These algorithms divide the map-matching process into initial MM and subsequent MM. Initial MM process locates vehicle on a network (i.e., map-matching for the first positioning point). In the subsequent MM process, if a vehicle travels on the same link that is previously map-matched the current positioning point is simply snapped on previously identified link. When a vehicle is near to a junction, the correct link identification is based on sum of individual weight scores.

Greenfeld (2002) reviews several approaches for solving the MM problem, and proposes a weight based topological MM algorithm. The matching process

consists of two separate algorithms: initial map-matching and subsequent map-matching. In initial matching, the closest node to the GPS point is identified first. Links connected to the closest node are considered as the candidate links. Among these candidate links the nearest link is considered to be the link on which the vehicle is travelling. The perpendicular projection of raw GPS on the selected link gives the vehicle position on that link. In subsequent MM, the correct link selection among candidate links is based on total weighting score (TWS), sum of three individual weights. These weight parameters are:

1) weight for heading (i.e. the degree of parallelism between the GPS line and the road network),
2) weight for proximity (i.e. the perpendicular distance of GPS point from the arc segment) and
3) weight for intersection (i.e. the intersecting angle if an intersection exists). TWS formula is:

$$TWS = W_{AZ} + W_D + W_I \tag{3.1}$$

Where

Weight for heading (W_{AZ}) = $C_{AZ} \cos^{n_{AZ}}(\Delta AZ)$

Weight for proximity (W_D) = $C_D - a\, D^{n_D}$

Weight for intersection (W_I) = $C_I \cos^{n_I}(\Delta AZ)$

The values of the parameters in the equation are (Greenfeld, 2002):

$C_{AZ} = 10,\ n_{AZ} = 5,\ C_D = 10,\ a = 0.1,\ n_D = 1.2,\ C_I = 10,\ n_I = 10$

ΔAZ is the heading difference between vehicle movement direction and link direction in degrees.

D is the distance from the GPS point to a candidate road segment.

A pictorial form of the three weights, heading, proximity and intersection, are shown in Figures 3.11, 3.12 and 3.13.

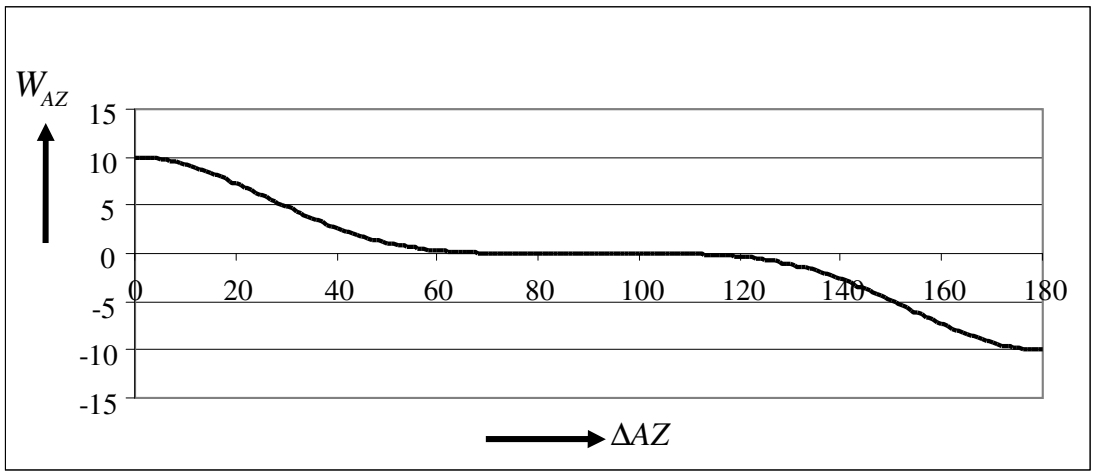


Figure 3.11: Heading weight proposed by Greenfeld (2002)

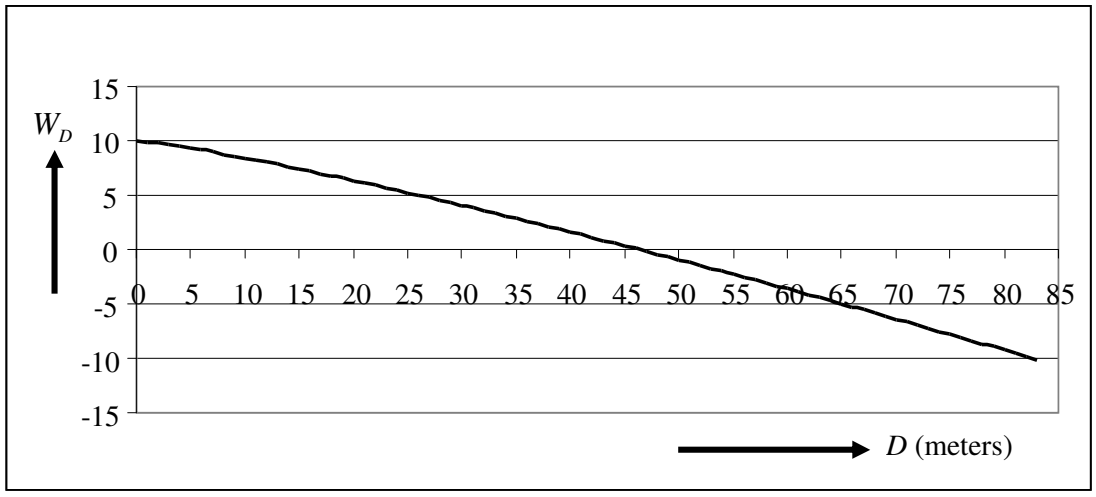


Figure 3.12: Proximity weight proposed by Greenfeld (2002)

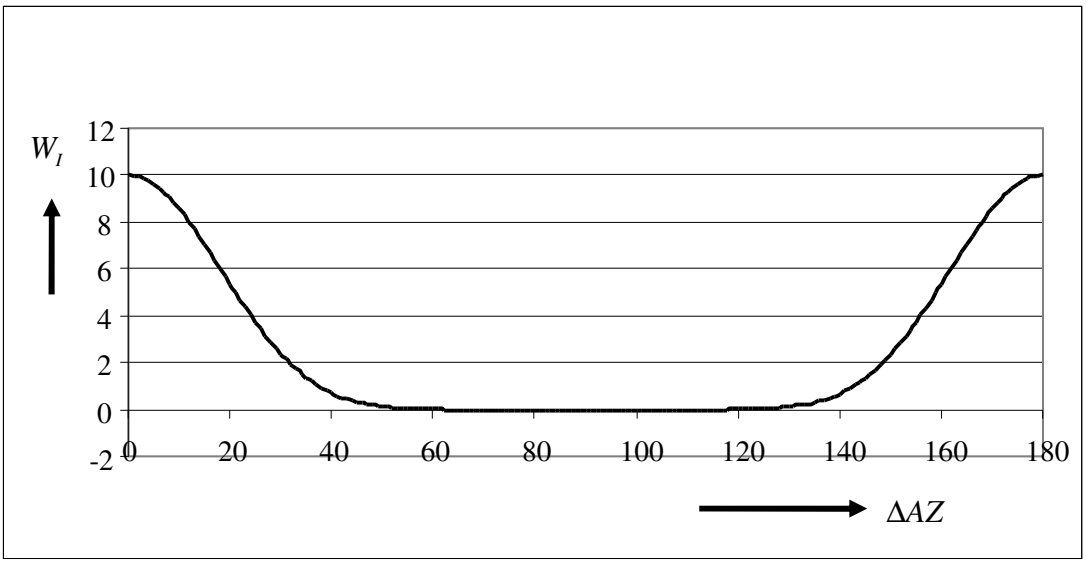


Figure 3.13: Intersection weight proposed by Greenfeld (2002)

Heading weight (W_{AZ}) is a function of the difference between the vehicle movement direction and link direction. Proximity weight (W_D) is a function of perpendicular distance GPS point to a link and weight for intersection (W_I) is a function of difference between the bearings of the GPS line and the evaluated network arc.

Figures 3.11, 3.12 and 3.13 show that heading and proximity weights vary from '+10' to '-10' and intersection weights vary from '+10' to zero. The weight based approach developed by Greenfeld (2002) assumes equal importance to each of the three weights.

Greenfeld's algorithm uses vehicle speed, heading and link orientation information, but it does not consider turn restriction information. Algorithm performance was not reported. However, Quddus et al. (2003) tested this algorithm and concluded that sometimes the algorithm identified incorrect road segments.

Quddus et al. (2003) describe limitations of existing topological MM algorithms and develop a general MM algorithm intended to support the navigation function

of a real-time vehicle performance and emission monitoring system and other ITS applications. The MM process is initiated with nodal matching (i.e., identification of the nearest node to the first GPS point). Identification of the correct link among all the links connected with the closest node is based on three weighting parameters. Those are: 1) weight for vehicle heading 2) weight for proximity and 3) weight for position of the point relative to the link. The highest total weight score (TWS) (sum of above three scores) determines the most likely candidate link for the correct match. The same procedure is repeated when vehicle is near junction. The TWS is defined as (Quddus et al., 2003):

$$TWS = WS_H + WS_{PD} + WS_{RP} \qquad (3.2)$$

Where

Heading weight $(WS_H) = A_H \cos(\Delta\beta)$

Proximity weight $(WS_{PD}) = A_{PD}\ \omega$

Weight for position of point relative to link $(WS_{RP}) = A_{RP} \cos(\theta)$

A_H, A_{PD} and A_{RP} are the weighting parameters for heading, proximity and position of point relative to the link respectively.

$\Delta\beta$ is the angle difference between vehicle movement direction and link direction.

$\Delta\beta = \alpha$ if -180°<= α <= 180°

$\Delta\beta = 360 - \alpha$ if α >180°

$\Delta\beta = 360 + \alpha$ if α <-180°

$\alpha = \alpha_1 - \alpha_2$

α_1 is the heading of each candidate link with respective to the north

α_2 is the vehicle heading with respective to the north

ω = 1 if D < 5 m

$\omega = 1.0 - 0.01D$ if $5 \leq D \leq 100$

$\omega = -1$ if $D > 100$

D is the perpendicular distance from a GPS point to a candidate road segment.

θ is the intersecting angle between the line between two consecutive GPS points and the link. If the two lines do not intersect then the score corresponding to $(A_{PD} * Cos\theta)$ is taken as zero.

α is the angle between the candidate link and the link through the nearest node and GPS point. α is shown in Figure 3.14.

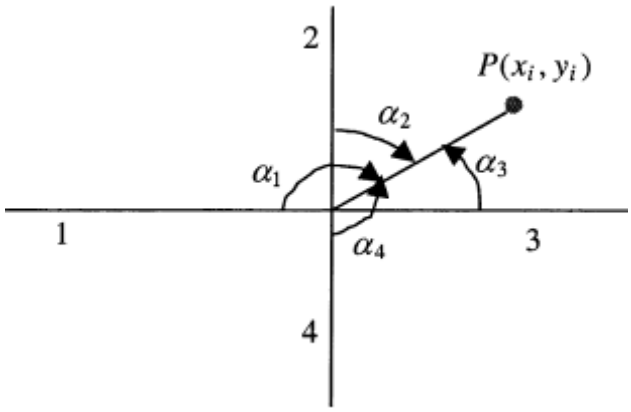


Figure 3.14: Location of GPS point relative to link (Source: Quddus et al. (2003))

Relationship between the three weight parameters (A_H, A_{PD} and A_{RP}) are:

$$A_H = a * A_{PD}$$
$$A_{RP} = b * A_{PD} \quad (3.3)$$

Where a and b are weighting factors that give the strength of relationship between A_H, A_{PD} and A_{RP}. Using an empirical analysis, Quddus et al. (2003) identified the weighting factors a and b as 3 and 2 respectively. The empirical analysis was done by checking the algorithms performance, with respective to correct link identification, for different combination of a and b values. Here, the minimum and maximum values of both a and b were considered to be 0.5 and

4.0. In each iteration an increment of 0.5 was added to the *a* and *b* values. With these weight factors (*a* is 3 and *b* is 2), the algorithm is capable of snapping 89% of GPS/DR points correctly, with 2D horizontal accuracy of 18.1m (95%). Quddus et al. (2003) concludes that actual link identification using weighting scheme is sensitive to weighting factors (a and b) and suggested further research to obtain the optimal values for the weighting parameters.

Taghipour et al. (2008) used the weight-based topological MM algorithm develop by Quddus et al. (2003) and applied a consistency check at junctions. But, no further improvement is done in total weight score calculation, the relative importance of each weight is as recommended by Quddus et al. (2003). The new consistency check at a junction is shown in Figure 3.15.

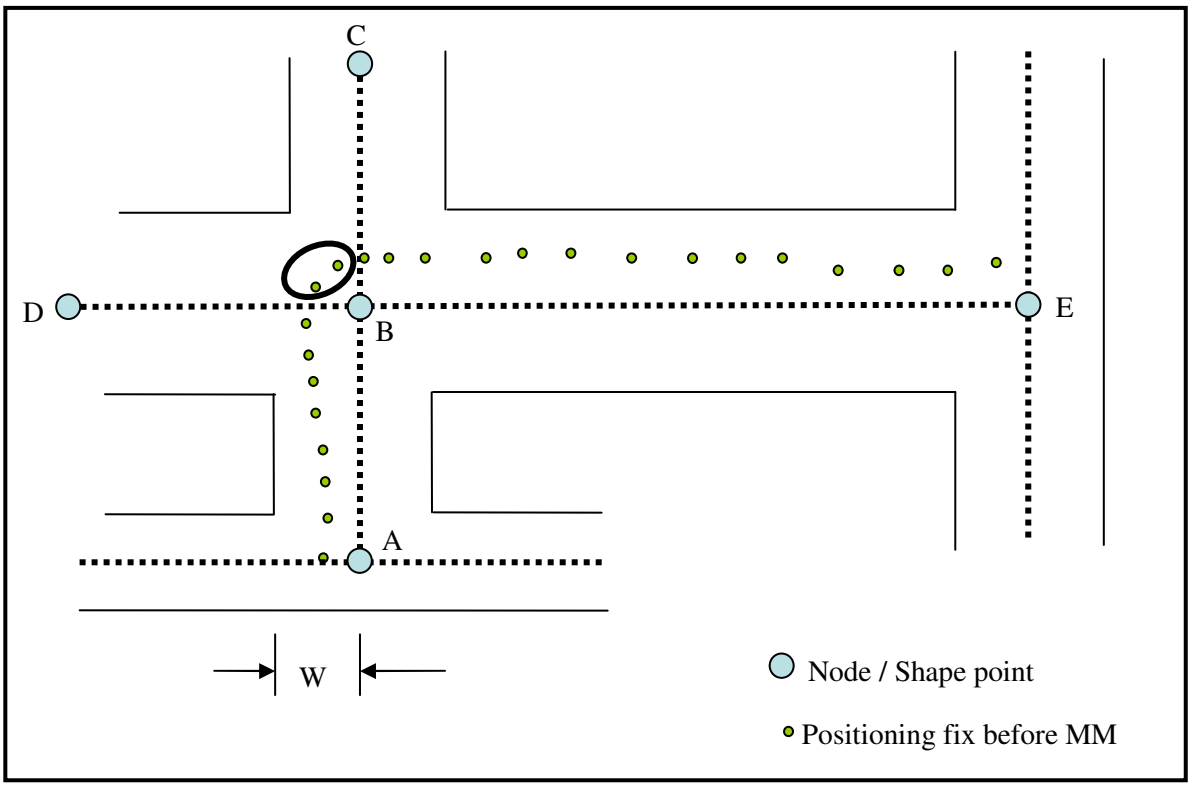


Figure 3.15: Consistency check at Junction

In Figure 3.15 the dotted line represents the road central line. Here, a vehicle travelled from junction A to junction B and then to junction E. At the junction B,

positioning fixes near to junction B (highlighted with a bold circle in the figure) do not fall either on link A-B or link B-E. The weight-based algorithms that choose the link based on the total weight score (sum of heading, proximity and intersection weights) choose either link B-C or B-D as the correct link for the positioning points at junction B.

In the consistency check at a junction, if the distance calculated by adding the length of the road link and the half of the width of previous street travelled by the vehicle (in Figure 3.15, the sum of length of link A-B and end the road width W) is less than the distance travelled by the vehicle from the previous junction to the current positioning point (multiplying the vehicle speed and time) then it is considered as vehicle is still travelling on the previous road segment. Otherwise, it is considered that the vehicle is at a junction and based on the TWS the new road segment is identified. The consistency check has improved the map-matching process, but requires that the width of the each road segment in the network is known. The enhancement of the proposed algorithm was demonstrated using a few critical junction scenarios (where Quddus tMM algorithm mismatches the vehicle position) in Arak city, Iran. However, the performance is not quantified with respect to the correct link identification and horizontal accuracy.

3.3.3 Off-line topological MM

In this section off-line topological MM algorithms developed by Yin and Wolfson (2004), Li et al. (2005) and Marchal et al. (2005) are reviewed.

Li et al. (2005) develop an algorithm based on the connectivity of road network. Continuous buffer bands are used to identify the road candidate links, and then evaluate the candidates based on heading, proximity and connectivity. A buffer band is created by using three parameters. They are: (1) buffer distance (BD) (2) distance between two key points (ND) (3) searching distance (SD) are shown in Figure 3.16.

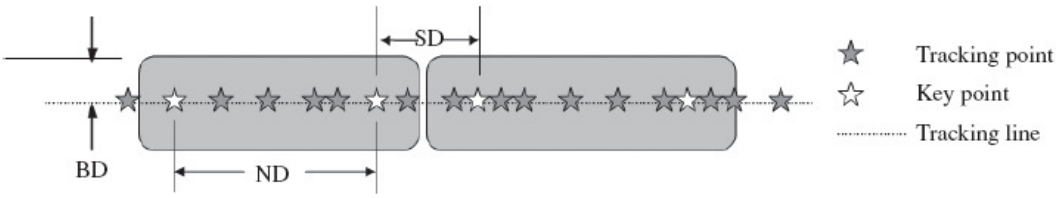


Figure 3.16: Buffer band (Source: Li et al. (2005))

Buffer distance (BD) is determined by error associated with the tracking data. Searching distance (SD) is the distance between two continuous buffer bands. The value of ND is set several times BD.

The value of SD is

$$SD = BD * P \tag{3.4}$$

P value varies in between 0 and 2.

$P = 0$ if end of one buffer band is also start point of its successive buffer band.

$P = 2$ for direct tracking line

From field trials, it has been observed that, in most of the travelled route, GPS tracking line is within 15m from the road centre line. The maximum distance of tracking line from the road centre line is observed as 75m. Based on the error associated with tracking data, in their study, two sets of buffer band parameters are considered:

set 1: BD=20m, SD=30m and ND=100m and

set 2: BD=80m, SD=80m and ND=200m.

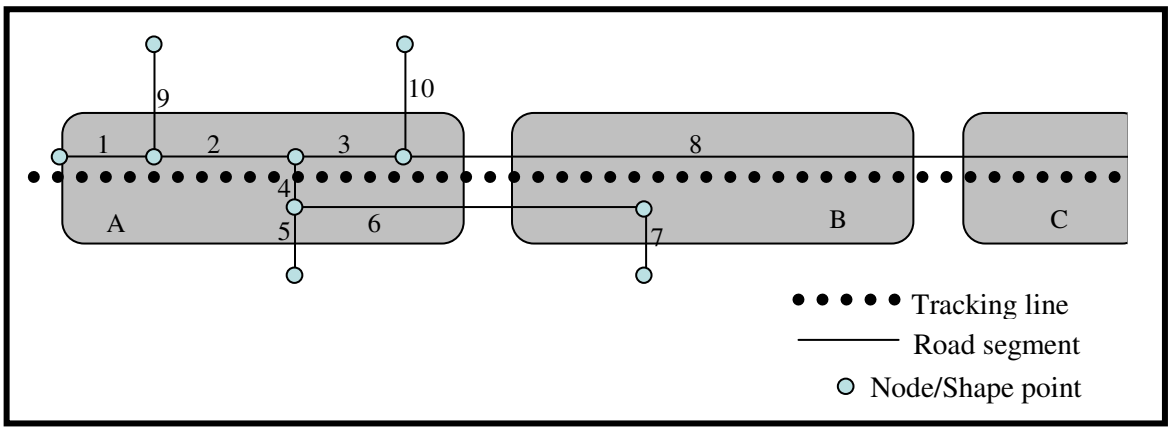


Figure 3.17: Buffer band concept for candidate link identification
(Source: Modified from Li et al. (2005))

A rounded rectangle error region, with centre line connecting several tracking point, is constructed to identify a set of candidate links. Every road segment interacting with or covered by buffer band are considered as candidate links. Further candidate links are classified into five categories. They are:

Type 1: road segments intersecting buffer bands, not within the buffer band (in Figure 3.17 road segments 5, 7, 9 and 10).

Type 2: segments interacting with two continuous buffer bands, and segment runs through one of the buffer band (in Figure 3.17 road segments 8 comes under this category).

Type 3: segments interacting with two continuous buffer bands, and starts former band (in Figure 3.17 road segments 6 and 8 of buffer band A).

Type 4: segments interacting with two continuous buffer bands, and ending in later buffer band (in Figure 3.17 road segments 8 of buffer band B).

Type 5: segments totally covered by buffer bands (in Figure 3.17 road segments 2, 3 and 4).

In their study, few candidate links are eliminated based on heading and proximity and connectivity. After evaluating candidate links based on heading, proximity and connectivity, several road links may still be available to select as the correct link. The final correct link identification is based on length comparison. Minimum length difference between length of road segment between previously map-matched point to current map-matched point and the length of tracking line covered by same GPS points. Correct link identification among candidate links is not based on total weight scores. Candidate links are checked with heading and proximity. The performance of the developed algorithm was not presented.

Yin and Wolfson (2004) propose a weight based map-matching algorithm for offline snapping problem. To compute the weight of each arc following two factors are considered: 1) possible route of the trajectory should be closer to the geometry of the arc and 2) each arc of possible route of trajectory is in the similar direction to that of corresponding sub-trajectory. The algorithm was tested in Chicago metropolitan area and identified 94% of the links correctly.

Marchal et al. (2005) present an off-line map-matching algorithm using network topological information. The algorithm finds a set of candidate link that are closer to the GPS point. Among the candidate links the correct link identification is based on absolute score value. The absolute score 'F' of path 'P' is defined as:

$$F = \sum_{j=1}^{p} \sum_{i=1}^{T} d(Q_i, E_j) * \delta_{ij} \qquad (3.5)$$

Where,

$P\{E_1, E_2 .... E_j ..., E_P\}$ denote the path composed of the P subsequent links

$E_1, E_2 ....., E_P$.

$\{Q_1, Q_2 .... Q_i ...., Q_T\}$ is the set of GPS points.

$d(Q_i, E_j)$ represents the distance between the i th GPS point to j th link.

$\delta_{ij} = 1$ if Q_i was assigned to link E_j else $\delta_{ij} = 0$

The algorithm developed by Marchal et al. (2005) concentrates on computational speed of the algorithm in addition to 2D accuracy. They concluded that the algorithm has capability of handling huge volumes of data in a considerable time period. The computational speed of developed algorithm is 2,000 GPS points per second. The major drawback is the algorithm does not consider vehicle heading, speed information and turn restrictions at junctions.

3.7 Performance of the existing MM algorithms

As discussed, a number of different algorithms have been proposed for different applications. 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; Taghipour et al., 2008). 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 3.1. Most did not assess algorithm performance with respect to 2-D horizontal accuracy due to a lack of high 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 columns 7 and 8 of Table 3.1. This positioning data, employed by Quddus (2006), is also used later in this

research to test the performance of the enhanced tMM algorithm to enable a real test of comparative performance to be made. Table 3.1 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 (university roads). 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 3.1 also suggests that when tested on the same data set, weight-based algorithms perform better than non-weight-based algorithms.

Table 3.1 Review of the existing topological MM algorithms

Author and Year of publication	Navigation Sensors	Test Environment	Map Scale	Sample size	Correct link Identification	Correct link identification by (Quddus, 2006)	Horizontal Accuracy by (Quddus, 2006)	Topological information used
Non-weight based algorithms								
White et al. (2000)	GPS	Suburban	--	1.2 Km	85.80%	76.8%	32 m (95%)	Heading, proximity and link connectivity
Li and Fu (2003)	GPS and DR	Urban	--	--	--	--	--	Heading and proximity
Srinivasan et al. (2003)	GPS	University road network	--	242 GPS points	98.5%	80.2%	21.2 m (95%)	Heading and turn restriction
Blazquez and Vonderohe (2005	DGPS	Urban and Suburban	1:2400	600 DGPS points	94.8%	--	--	Connectivity and turn restrictions
Weight-based algorithms								
Greenfeld (2002)	GPS	Urban and Suburban	--	--	--	85.6%	18.3 m (95%)	Heading, proximity and intersection weights
Quddus et al. (2003)	GPS and DR	Urban	1:1250	--	88.6%	88.6%	18.1(95%)	Heading, proximity and position of point relative to link
Taghipour et al. (2008)	GPS	Suburban	--	--	--	--	--	Same weights as Quddus et al. (2003)
Off-line map matching								
Li et al. (2005)	GPS	Suburban	1:1000	--	--	--	--	Heading weight and proximity weight
Yin and Wolfson (2004)	GPS	Urban	--	--	94%	--	--	Heading and proximity
Marchal et al. (2005)	GPS	Urban and Suburban	--	2 million GPS points	--	--	--	Proximity and connectivity

3.8 Conclusion

Performance evaluation of tMM algorithms suggests that the weight based algorithms, which select the correct link based on the TWS assigned to each of the candidate links, are more logical; and performs better than non-weight based algorithms. Evidence from literature suggests the ways in which existing weight-based 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 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.

Considering the above enhancements, a new weight based topological MM algorithm is developed in this research and presented in Chapter 6. The following chapter provides the review of integrity methods in the context of land vehicle navigation.

Chapter 4

Review of Integrity Methods for Land Vehicle Navigation

4.1 Introduction

The raw positioning data from a navigation system contain errors due to satellite orbit and clock bias, atmospheric (ionosphere and troposphere) effects, receiver measurement errors and multipath errors (Kaplan and Hegarty, 2006). Digital maps include errors which can be geometric, e.g. displacement and rotation of map features, or topological, e.g. missing road features (Goodwin and Lau, 1993; Kim et al., 2000). Even where the raw positioning data and the map quality are good, MM techniques sometimes fail to identify the correct road segment especially at roundabouts, level-crossings, Y junctions, dense urban roads and parallel roads (White et al., 2000; Quddus et al., 2007). Any error associated with either the raw positioning fixes, digital map, or MM process can lead to wrong location identification (Quddus et al., 2006b). Users should be notified when the system performance is not reliable. This is vital for safety critical ITS applications such as emergency vehicle management, vehicle collision avoidance and liability critical applications such as electronic toll collection system and distance-based pay-as-you-drive road pricing. For example, the wrong vehicle location identification due to errors associated with a navigation system used in an emergency vehicle routing service delays an ambulance arrival at the accident site; this may lead to loss of life. If the user is informed when system

performance is not reliable then the user will not blindly depend on the navigation system.

Integrity monitoring of a navigation system can be categorised into system, network and user level monitoring (Ochieng et al., 2007; Feng and Ochieng, 2007). System level monitoring can provide integrity information by considering satellite orbit and clock bias. An example of system level integrity is Galileo. Network level integrity monitoring can be further classified into: Satellite Based Augmentation System (SBAS) and Ground Based Augmentations System (GBAS) (Bhatti et al., 2006). The SBAS is an overlay system to enhance the accuracy of GPS by using additional satellites. The SBAS considers satellite orbit and clock corrections, geometry of satellite and user, and ionosphere error, but not the tropospheric effects, receiver measurement error and multipath error (Feng and Ochieng, 2007). An example of SBAS is European Geostationary Navigation Overlay Service (EGNOS), operated by the European Space Agency. In case of GBAS, ground survey stations monitor the health of satellite and transmit correction to the user receivers via very high frequency transmitters. The GBAS considers troposphere correction in addition to the other errors covered in SBAS. An example of GBAS is the United States' Local Area Augmentation System (LAAS). Neither system level or network level integrity monitoring cover multipath error that generally occurs in urban canyons.

Integrity monitoring, at the individual user level which considers receiver measurement errors and multipath error along with all the other errors discussed above is referred to as the User Level Integrity Monitoring (ULIM) process. A ULIM with a stand-alone GNSS is generally called Receiver Autonomous Integrity Monitoring (RAIM). Integrity monitoring combining GNSS and other sensors (e.g. dead reckoning system or inertial navigations system) is commonly known as User Autonomous Integrity Monitoring (UAIM) (Ochieng et al., 2007). The UAIM is a special case of RAIM. The UAIM considers sensor (odometer and gyroscope) errors along with GPS receiver measurement errors and multipath error.

Integrity monitoring methods are further classified as snapshot or filtering methods (Feng et al., 2007). The snapshot method uses the current positioning point information to predict whether the positioning point is reliable or not. Whereas, the filtering method considers the history of previous positioning information, in addition to the current positioning point, to predict the untrustable positioning output. The filtering methods performs better than the snapshot method (Yun et al., 2008). The following section describes the integrity methods.

4.2 Integrity methods

As discussed in the previous section, the integrity monitoring process can be done at system network or user level. Integrity monitoring at a user level (i.e. integrity checking within the user equipment) considers all the errors including multipath error. It may not be possible to detect this error mode, at a system level or at a network level. Therefore, integrity checking within the user's receiver has more advantages. As mentioned before, the user level integrity monitoring with a stand-alone GNSS is generally referred to Receiver Autonomous Integrity Monitoring (RAIM). Various RAIM methods and their advantages and disadvantages are discussed in the section. The key RAIM methods are (Broughton, 2003):

1. Range comparison method
2. Least square residual method
3. Weighted least square residual method
4. Kalman filter method

The first three methods are snapshot methods (SM) and the fourth is a filtering method (FM).

4.2.1 Range Comparison Method

In the range comparison method (RCM), firstly the position is estimated using any four available satellites. Then, the range to each individual satellite is calculated. The difference (range residual) between the calculated range to individual satellite and the predicted range (using four satellites) is then measured. This is shown in Figure 4.1 (Lee, 1992). The reliability of positioning information depends on the quantity of range residuals (i.e., small or large). If the range residuals (obtained from all visible satellites) are relatively small the algorithm declares 'no failure'; otherwise 'failure.' The judgement on whether the range residuals are small or large, is made using an empirical test that checks whether the typical sample GNSS point lies inside or outside the alert limit (AL).

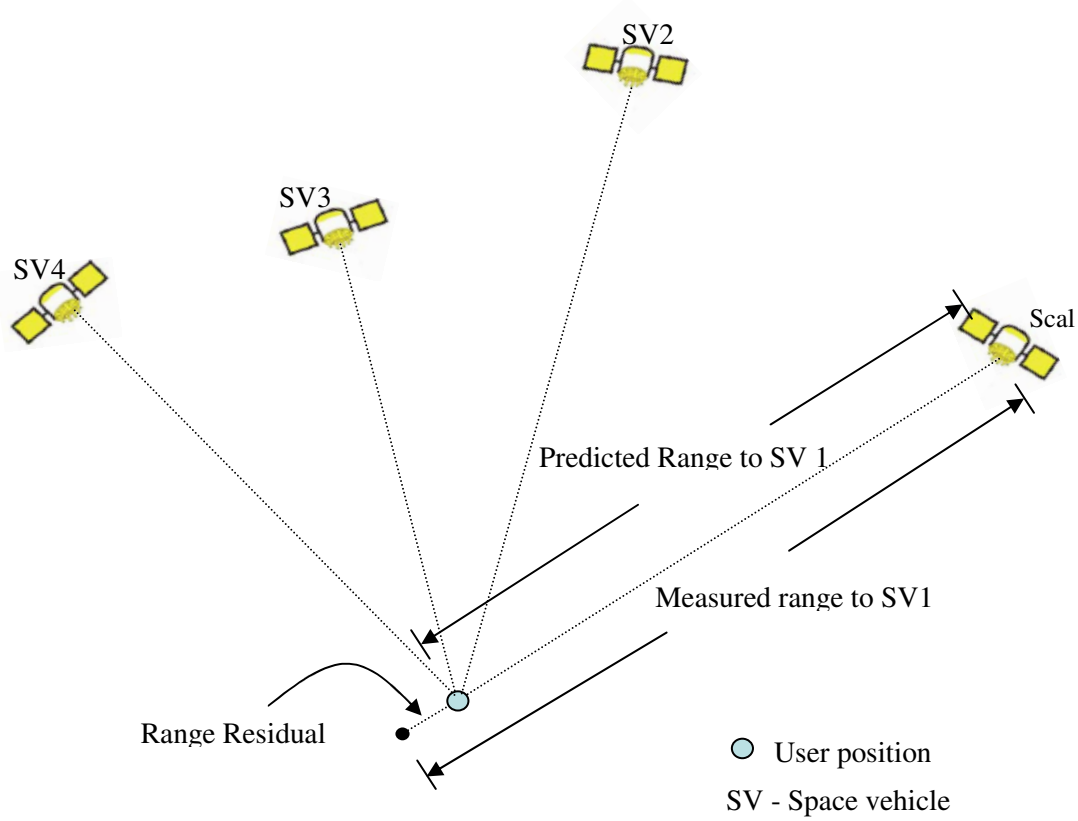


Figure 4.1: Satellite range comparison method
(Source: modified from Lee, 1992)

The RCM is very simple and easy to develop. But the basic disadvantages are:

(1) This method uses only current positioning point information to identify the reliability of positioning information. (2) Further, the initial estimated range (using any four visible satellites) could also contain some error; this error is not considered in decision making process on detecting a failure.

(3) It is not suitable for an integrated system (GNSS and DR).

(4) It is an empirical method.

4.2.2 Least squares residual method

There are four unknown terms: user longitude, latitude, height and time (*x, y, z* and *dt*) associated with the user position estimation. For example, if six satellites are available for a GNSS positioning point; instead of considering any four satellites to calculate the four unknowns, the least squares residual method considers all the available satellites; and the four unknowns are calculated by minimising the sum of squares of estimated errors. The derivation of a least squares technique is provided here (Taylor and Blewitt, 2006):

The vector matrix form of residual observation is:

$$P = AX + e \tag{4.1}$$

In equation (4.1),

P is the vector of residual observations of size *n by 1*.

A is the design matrix of size *n by 4*.

X is a vector of size *4 by 1* representing three components of user position and time. and

e is the error matrix of size *n by 1*.

From equation (4.1) the matrix form of estimated error can be written as:

$$\hat{e} = P - A\hat{X} \tag{4.2}$$

'^' in equation (4.2) represents the estimated value.

The square of the above error term in equation (4.2) for *n* number of satellites in functional form is:

$$f(x) = \sum_{i=1}^{n} \hat{e}_i^2$$

The above function in a matrix from:

$$f(x) = \hat{e}^T \hat{e} = (P - A\hat{X})^T (P - A\hat{X})$$

The least squares solution can be found where sum of squares of error (that is *f(x)*) is minimum. Therefore, if the change in the function (*f(x)*) is zero, then the function is minimum (Taylor and Blewitt, 2006).

$$\text{i.e., } \delta f(\hat{x}) = 0$$

$$\delta\left[(P - A\hat{X})^T (P - A\hat{X})\right] = 0$$

$$\delta(P - A\hat{X})^T (P - A\hat{X}) + (P - A\hat{X})^T \delta(P - A\hat{X}) = 0$$

$$(-A\delta\hat{X})^T (P - A\hat{X}) + (P - A\hat{X})^T (-A\delta\hat{X}) = 0$$

$$(P - A\hat{X})(-2A\delta\hat{X})^T = 0$$

$$(P - A\hat{X})A^T \delta\hat{X}^T = 0$$

$$(PA^T - AA^T \hat{X})\delta\hat{X}^T = 0$$

$$AA^T \hat{X} = PA^T$$

$$\hat{X} = (AA^T)^{-1} PA^T \tag{4.3}$$

From equation (4.1) and (4.3) estimated residual observation vector can be written as

$$\hat{P} = A\hat{X} = A(AA^T)^{-1}PA^T \tag{4.4}$$

Therefore, the range error (E) vector can be written as:

$$E = P - \hat{P} \tag{4.5}$$

From equation (4.4) and (4.5)

$$E = P - A(AA^T)^{-1}PA^T$$

$$E = P(I - A(AA^T)^{-1}A^T) \tag{4.6}$$

Substitute equation (4.1) in equation (4.6)

$$E = (AX + e)(I - A(AA^T)^{-1}A^T)$$

$$E = \left\{(AXI - A(AA^T)^{-1}A^T AX)\right\} + \left\{e(I - A(AA^T)^{-1}A^T)\right\}$$

$$E = e(I - A(AA^T)^{-1}A^T) \tag{4.7}$$

Where

E is the range error vector of size n by 1

I is the unit matrix of size n by n

A is the design matrix of size n by 4

e is the measurement error vector of size n by 1 and

n is the number of satellites.

Therefore, the square root of sum of squared range error is:

$$SSE = E^T E \tag{4.8}$$

Here, to declare the presence of a fault in the calculated receiver position two quantities are used: test statistics and decision threshold. A decision threshold value is chosen, for given statistical characteristics of test statistics, by controlling the false alarm and missed detection rate (Feng and Ochieng, 2007). The $\sqrt{SSE}$ is used as the test statistic. By using the test statistics and the decision threshold, the protection level[5] (PL) is calculated. The comparison of the PL against alert limit[6] guides whether to raise the alarm or not.

Disadvantages:

(1) Calculated SSE (or test statistic) is independent of satellite and user geometry.

(2) This method uses only current positioning point information

4.2.3 Weighted least squares residual error method

In the least squares residual method, the SSE is independent of satellite and user geometry. But, in practice, some satellites may have probably large error compared to other satellites. For instance, if the satellite elevation is higher the multipath effect and the atmospheric effects are less compared to a lower elevation satellite. The weighted least squares residual method provides more accurate receiver positioning information by giving more weight to satellites that are less likely to have errors (Broughtom, 2003).

Therefore, the weighted sum of squared range error is (from equation 4.7 and 4.8):

[5] Protection level is the upper boundary of the confidence region (circle with centre as computed position) of a GPS sensor, in which position error can be detected outside that region

[6] Alert level is the error tolerance not to be exceeded without issuing an alert to users.

$$WSSE = E^T WE \tag{4.9}$$

Where

$$E = e(I - A(AWA^T)^{-1}WA^T) \tag{4.10}$$

Where,

W is the weight matrix of size n by n, that is inverse of the covariance matrix

I is the unit matrix of size n by n

A is the design matrix of size n by $4$and

e is the measurement error vector of size n by 1.

The weighted least square residual error has similar disadvantages of least square residual error method except it gives different weights to all visible satellites.

4.2.4 Kalman filter method

For land vehicle navigation, stand alone GNSS systems face signal masking, particularly in urban canyons. To overcome this problem, often the GNSS system is integrated with other sensors such as DR. A Kalman filter is a set of mathematical equations, that integrate the data from the GNSS receiver and the data from other sensors, to provide a precise receiver positioning information (Krakiwsky et al., 1988). The Kalman filter approach, which computes an integrated solution, is also used for integrity monitoring purpose (Philipp and Zunker, 2005; Feng and Ochieng, 2007). A brief description of Kalman filter (proposed by Sun and Cannon, 1998) is provided here:

A Kalman filter primarily consists of a system equation and an observation equation.

The system equation is given by:

$$X_k = \Phi_{k-1} X_{k-1} + w_k \tag{4.11}$$

Where

X_k is the state vector

Φ_{k-1} is the transition matrix,

w is the system error matrix

the subscript k represents the time step (i.e., epoch k)

The observation equation is given by

$$z_k = H_k X_k + v_k \tag{4.12}$$

where

z_k is the measurement vector

H_k is the measurement matrix

v_k is the measurement error vector

The prediction of state vector and variance-covariance matrix can be written as:

$$\hat{X}_k^- = \Phi_{k-1} \hat{X}_{k-1} \tag{4.13}$$

$$P_k^- = \Phi_{k-1} P_{k-1} \Phi_{k-1}^T + Q_{k-1} \tag{4.14}$$

In equation (4.13) and (4.14), '-' represents the predicted value. P_k is the variance-covariance matrix and Q is the process error covariance matrix. For initial iteration (i.e., k=1), $\hat{X}_{k-1}$ and P_{k-1} are the inputs to the system.

Then, the system calculates the Kalman gain matrix, which is given by

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_K \right)^{-1} \tag{4.15}$$

In equation (4.15) K_k is the Kalman gain matrix; H_k is the measurement matrix; P_k variance-covariance matrix which is obtained from equation (4.14); and R_K is measurement error covariance matrix.

Using equation (4.13) and (4.15) the state vector and variance-covariance matrix are updated.

$$\hat{X}_k = \hat{X}_k^- + K_k(z_k - H_k \hat{X}_k^-) \tag{4.16}$$

$$P_k = (1 - K_k H_k)\hat{P}_k^- \tag{4.17}$$

Where,

$\hat{X}_k$ is the final state vector for epoch k

H_k is the measurement matrix

z_k is the measurement vector (from equation 4.12); and

$\hat{X}_k^-$ and K_k are obtained from equation (4.13) and (4.15).

Further detailed description of Kalman filter to integrate GPS and DR sensors can be found in Zhao et al. (2003). The basic integrity monitoring process, using Kalman filter, was designed by Lu and Lachapelle (1991), Sun and Cannon (1998), and Feng and Ochieng (2007).

The innovation matrix is given as (Feng and Ochieng, 2007)

$$v_k = z_k - H_k X_k^- \tag{4.18}$$

Where

v_k is the measurement error vector

z_k is the measurement vector

H_k is the measurement matrix

X_k^- is the predicted state vector

The weight matrix (W), which is inverse of co-variance matrix, is

$$W_k = (H_k P_k^- H_k^T + R_K)^{-1} \tag{4.19}$$

In the above equation, H_k is the measurement matrix; P_k variance-covariance matrix, which is obtained from equation (4.14); and R_K is measurement error covariance matrix.

Then, sum of squared errors is (Feng and Ochieng, 2007)

$$SSE = v_k W_k v_k^T \tag{4.20}$$

$\sqrt{SSE}$ is used as test statistic. It is assumed that the SSE follows the Chi-square distribution with n degrees of freedom. The calculation of protection level is similar to the procedure explained in least squares residual method. This PL is compared against alert limit to decide whether to raise an alarm for that positioning fix or not. The derivation of decision threshold depends on the probabilities of false alert and missed detection.

Philipp and Zunker (2005) also consider the state variance covariance matrix (i.e., P in equation 4.15), which is calculated for each positioning fix, to determine the integrity of positioning output. In matrix P, first *2* by *2* sub-matrix is the variance and covariance of positioning information in X and Y directions with 68.3% confidence level (i.e. one sigma). Using this P matrix, horizontal tolerance limit (HTL) is estimated with higher level of confidence (6-sigma level). The HTL, which is analogous of protection limit (PL), concept is introduced to detect the failures in the positioning data.

The Kalman filter based integrity monitoring process is useful for integrated sensor systems (such as GNSS and DR). Moreover, this filtering approach uses the previous positioning points data in addition to the current GNSS point information.

As a part of the integrity monitoring process, the quality of satellite and user geometry also plays an important role. To check the geometry of satellite, with respect to the user, in the failure detection process three commonly employed methods are used in the literature (Broughtom, 2003). They are: (1) $\delta H_{\max}$ method (Parkinson and Spilker, 1996) (2) approximated radial error protected (ARP) method (Parkinson and Spilker, 1996) and (3) horizontal protection limit (HPL) method (Broughtom, 2003). The following section describes applications and performance of integrity method in the context of land vehicle navigation.

4.3 Application and performance of integrity methods

Basic approaches for integrity methods vary from a very simple process (e.g. range comparison method) (Walter and Enge, 1995; Lee, 1992) to complex methods (e.g. Kalman Filter method) (Yun et al., 2008; Lee at al., 2007). Studies focusing on the enhancement of integrity methods for land vehicle navigation are discussed in this section (Jabbour et al., 2008; Feng and Ochieng, 2007; Lee et al., 2007; Santa et al., 2007 Kuusniemi et al., 2007; Gomes and Pereira, 2007; Santa et al., 2006; Quddus et al., 2006c; Yu et al., 2006; Philipp and Zunker, 2005; Syed, 2005; Wullems et al., 2004; Gustafsson et al., 2002; Sun and Cannon, 1998). The land vehicle navigation integrity monitoring process can be categorised into: integrity of map-matching process, positioning integrity monitoring and other integrity methods.

4.3.1 Integrity of Map-matching process

Quddus et al. (2006c) developed an integrity method for map-matching algorithms. The integrity of the map-matched solution was assumed to be a function of: 1) uncertainty associated with the positioning solution 2) comparison of vehicle heading with respect to selected road link direction and 3) distance comparison between raw positioning fix and map-matched position. The integrity of the map-matched position is represented by a scalar value, that ranges from 0 to 100 (0 indicates a low level of confidence and 100 indicates a high level of confidence). The derivation of the scalar value (0-100) is done using a *sugeno fuzzy inference system*. The authors tested the method, on three different map-matching (MM) algorithms, using stand-alone GPS data and GPS/DR data, for three different (scale) digital maps. The valid integrity warnings using a topological map-matching, probabilistic map-matching and advanced map-matching were found to be 91.1%, 97.5% and 98.2% respectively. Quddus et al. (2006c) used a simple empirical method to identify the integrity of MM positions, and proposed the employment of more rigorous statistical methods.

Jabbour et al. (2008) proposed a map-matching algorithm and an integrity method - using a multihypothesis technique - for land vehicle navigation. The integrity of a map-matched positioning fix is checked using two indicators: (1) number of efficient hypotheses and (2) normalised innovation square. In the integrity monitoring process, they used two threshold values; and these values are derived empirically. Their method did not consider the error sources associated with the digital map. As Quddus et al. (2006c) did, they also identify the overall correct detection rate, which is function of false alarm rate and missed detection rate. The MM algorithm and the integrity method were tested using real-time field data (3,661 positioning fixes) collected in Compiegne, France. And it was found that the integrity method can give 88.8% valid warnings.

Yu et al. (2006) used a curve pattern matching to identify the reliability of map-matched output. After map-matching a positioning point, the algorithm forms a curve using all the positioning points and the current point. Then the algorithm also forms another curve using map-matched position. By comparing these two curves the algorithm identifies mis-matching, and reinitiates the process and corrects the mis-matching. The process is more like a consistency check. Integrity process performance was examined using an extensive data (3,000 km travel length) collected in central Hong Kong. The performance was examined only with missed detection rate (MDR); which was 1.41%. The error associated with GNSS system and digital map were not considered in the integrity process.

4.3.2 Positioning integrity monitoring

Philipp and Zunker (2005) developed a novel integrity process for land vehicle navigation applications. A Kalman filter was designed which can consider errors with a GNSS system and other sensors (DR). The variance-covariance matrix obtained from a Kalman filter is used to identify the horizontal trust limit (HTL). The HTL is measure of accuracy of integrated position solution obtained from GPS and other vehicle sensors. The process was tested using a small data set collected in Ulm, Germany. However, a quantitative measure of performance of developed integrity method is not reported.

Feng and Ochieng (2007) developed a user-level Vehicle Autonomous Integrity Monitoring (VAIM) algorithm for safety and liability critical ITS applications. The integrity algorithm uses a hybrid method consists of a weighted least squares technique and an Extended Kalman Filter (EKF) approach. Though, the authors consider the odometer and gyroscope errors in the failure detection process, they did not consider the digital map errors. The developed integrity method was tested using a data collected near London, and it was found that the algorithm can efficiently detect potential failures in the navigation systems outputs. The percentage of valid warnings was not measured.

Sun and Cannon (1998) conducted a reliability analysis of navigation module of intelligent transport systems (ITS). Three separate cases were analysed. They are: (1) GPS only scenario (2) GPS integrated with gyro (3) GPS integrated with gyro and digital map. A standard kalman filter was used for the reliability analysis. Instead of employing the errors associated with the digital map, their study used the longitude and latitude coordinates of a map-matched point to aid the overall integrity monitoring. The performance in terms of integrity was not measured individually.

Gomes and Pereira (2007) discussed the fundamentals of integrity requirements and state-of-the-art in integrity monitoring process for electronic fee collection. The basic advantages of integrity monitoring for road tolling is categorised into: operational use and legal use. This study did not consider the digital map errors in the integrity monitoring process. Santa et al. (2006) developed a reliable integrity monitoring process for location based services. Embedded software, for integrity monitoring was developed.

4.3.3 Other Integrity Methods

Lee et al. (2007) developed an integrity method - using the Monte Carlo technique - for aviation and land vehicle navigation. The total process is divided into: fault detection (FD) and fault isolation (FI). The FD process is carried out using particle filter approach; and log-likelihood ratio method is used for FI. This research considered GNSS system errors but not the error associated with the other sensors (such as DR or INS). Gustafsson et al. (2002) also developed a particle filter - using a Sequential Monte Carlo method - for positioning navigation and tracking. The developed filter was tested for aviation and car navigation. Their method considers integrated navigation (GPS + DR / INS); but, the digital map errors are not considered in the filtering process. Kuusniemi et al. (2007) used the Modified Danish method for reliability testing of GNSS

positioning output for indoor applications. The Danish method is an iteratively reweighted least square estimation.

Few studies report the performance of the developed integrity monitoring processes. Those, that did are shown in Table 4.1. It can be concluded that the performance of the integrity methods varies widely. In column 6, the valid integrity warnings include false alarm rate (FAR) and missed detection rate (MDR). The performance of the integrity method developed by Yu et al. appears good, however, due to lack of true positioning fixes, they did not measure the performance with respect to FAR (Yu et al., 2006). If the FAR is also considered in performance evaluation, the percentage of valid warnings provided by their integrity process may be lower. Feng and Ochieng (2007) checked their integrity method in four different scenarios (i.e., in tunnels, flyovers, bridges, and parallel roads). Their algorithm did not quantify the performance with respect to the valid warnings. The other integrity methods developed in the literature provides about 90 percent valid warnings. Moreover the performance of the integrity method is sensitive to quality of digital map (map scale) and type of map-matching algorithms used in the vehicle navigation module (Quddus et al., 2006c). The integrity method developed by Quddus et al. (2006c) performs better when an advanced map-matching algorithm and more accurate digital map (scale 1:1,250) was used; and it gives 98.2% valid warnings. Its performance was recorded as 91.1% when it was tested with a weight based topological map-matching algorithm. However, they examined the integrity method performance only with 840 positioning fixes, in a sub-urban area. The performance of the existing integrity methods may not support few safety and liability critical ITS applications. Further enhancement of the integrity method is required. The following section discusses how this integrity method can be further improved.

Table 4.1: Performance of integrity methods for land vehicle navigation

Author and Year of publication	Methodology used	Sample size	Map Scale	Test environment	Performance
Map-matching integrity					
Quddus et al. (2006c)	Empirical method Fuzzy logic system is used to represent integrity in a scalar value	840 positioning fixes	1:2,500	Sub-urban	91.1% valid warnings[1]
					97.5% valid warnings[2]
					98.2% valid warnings[3]
Yu et al. (2006)	Curve pattern matching	3,000 KM length	1:1,000	Urban	MDR is 1.41%
Jabbour et al. (2008)	Multihypothesis technique	3,661 positioning fixes		Urban	88.8 % valid integrity warnings
Raw GNSS positioning fix integrity					
Sun and Cannon (1998)	Kalman filter	--	1:5,000	Urban and Suburban	79.9% valid warnings
Philipp and Zunker (2005)	Kalman filter	--	--	Urban (in Ulm, Germany)	--
Kuusniemi et al., (2007)	Modified Danish method	12 hours data	--	--	89.5% valid warnings
Feng and Ochieng (2007)	Weighted least squares technique and Extended Kalman Filter	--	1:2,500	Sub-urban	Four different scenarios (i.e., in tunnels, flyovers) were discussed

[1]Using a weight based topological MM algorithm.
[2]Using a probabilistic MM algorithm.
[3]Using an advanced MM algorithm.
MDR: missed detection rate

4.4 Research gap in integrity monitoring

The reliability of the final vehicle positioning output obtained for a vehicle navigation system mainly depends on three components:

(1) errors in raw positioning fix (such as satellite orbit and clock bias, atmospheric (ionosphere and troposphere) effects, receiver measurement error and multipath error)

(2) quality of digital road map and

(3) errors in map-matching process.

State-of-the-art research in land vehicle integrity monitoring has concentrated on either the integrity of raw positioning information obtained from GNSS/DR (Philipp and Zunker, 2005; Feng and Ochieng, 2007; Lee et al., 2007) or the integrity of map-matching process (Yu et al., 2006; Quddus et al., 2006c; Jabbour et al., 2008). The uncertainties associated with the digital map are considered only by Quddus et al., 2006c. Quddus et al., 2006c measured the digital map uncertainty using the length of road segment and GIS map scale.

Clearly, considering these three sources of errors together should lead to a better outcome. Moreover, taking the complexity of the road network (i.e., operational environment) into account can further improve the integrity process. Therefore, the ways to further improve the integrity of land vehicle navigation are identified as:

(1) Consider errors associated with the positioning data, GIS map data and map-matching process together in identifying the goodness (trustability) of final positioning fix obtained from a vehicle navigation module.

(2) Though a raw positioning fix includes considerable error, if the operational environment is not complex (i.e., a rural environment), simple map-matching algorithm can still identify the correct road segment on which a vehicle is travelling. So, considering operational environment in the integrity of land vehicle navigation can further improve its performance.

The development of an integrity method for land vehicle navigation, which considers the limitations of the existing studies, is shown in Chapter 8. The

following chapter provides the description of different positioning data sets collected and used in this thesis.

Chapter 5

Data Collection

5.1 Introduction

This chapter describes the positioning data sets used in this thesis. To test, implement and evaluate the performance of developed MM algorithm and integrity method a total of six positioning data sets were utilised. A detailed description of each data set is provided in this chapter. The use of positioning data from three different countries with three different digital road maps should address issues of transferability of the developed map-matching algorithm and the integrity method.

5.2 Positioning data collection

Six positioning data sets were collected from the UK, India and USA. These data sets are shown in Table 5.1. The first column of the table represents the data set number and the second column provides the location of the test route. The date of data collection and the equipment used are shown in column three and four respectively. Columns five and six provide sample size with respect to number of positioning points, and approximate test route length in km. The total number of positioning points including all six data sets is 88,481 with an approximate total test route length of 1,098 km. The last column provides the location characteristics of the test route. Data sets 1, 2, 3, and 4 were collected as part of

this research and the data sets 5 and 6 were obtained from Quddus (2006). As it can be seen in Table 5.1, positioning data from various operational environments including dense urban, suburban and rural areas were used in this research. This is because the performance of a vehicle navigation module depends on the complexity and operational characteristics (such as tall buildings, bridges) of the road network. All these data sets recorded positioning data every second. In the rest of the thesis positioning data is referred-to by the data set number given in Table 5.1. The data sets are now discussed in here.

Table 5.1: Positioning data

Data set	Test Location	Date	Equipment used	Sample size	Approximate test route length (km)	Location characteristics
1	Loughborough to London; Central London and south part of London, UK	May-08	AEK - 4P and AEK 4R	42,231 points	700	Mix of dense urban, suburban, and rural
2	Mumbai, India	Dec-08	AEK-4P	16,756 points	95	An urban area with narrow congested roads and construction site
3	Washington, DC, USA	Jan-09	AEK-4P	3,900 points	17	Dense urban road network, tall buildings, bridges and flyovers
4	In and around Nottingham, UK	May-09	AEK - 4P AEK 4R, a Carrier-phase GPS with high grade INS	10,347 points	89	Urban and suburban areas
5	Suburban: (West of London– near Reading)	Aug-05	GPS/DR and a carrier-phase GPS	2,040 points	49	Suburban area
6	Central London and suburban area of London	July-04	GPS	13,207 points	148	Dense urban and suburban areas

In the case of the first data set, a test vehicle equipped with a single frequency high sensitivity GPS receiver and a low-cost gyroscope were used. The positioning data were collected in the United Kingdom (a return trip from Loughborough to London and inside central London) on 26th of May 2008. The test route was selected carefully to ensure that the vehicle travelled through different operational environments including rural, suburban and a good mix of urban characteristics (tall buildings, bridges, flyovers and tunnels etc). As the test vehicle travelled on a pre-defined route the road segment on which vehicle travelled is known. The total trip length of the data set is about 700 km. The test route trajectory is show in Figure 5.1. The black line indicates the test route.

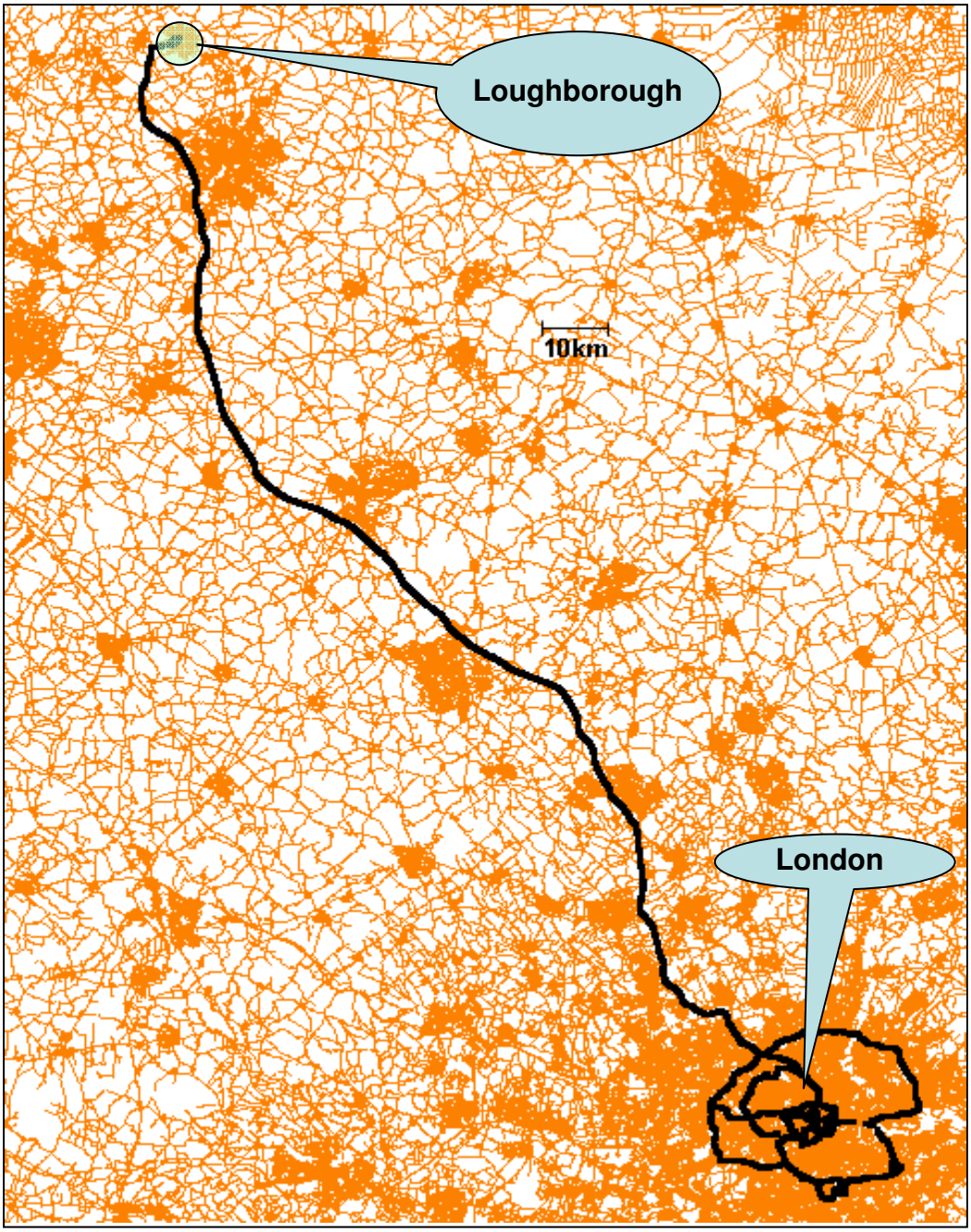


Figure 5.1: Data set 1- test route in the United Kingdom

The second and third data sets were collected using a 16-channel single frequency high sensitivity GPS receiver (AEK-4P) on pre-planned routes in congested road network of Mumbai, India and the downtown area of Washington, DC, USA on 8th December 2008 and 13th January 2009 respectively. For both data sets (2 and 3), the test routes were again selected

carefully to ensure that the vehicle travelled through a good mix of urban characteristics. The total trip length of data sets 2 and 3 were about 95 km and 17 km respectively. The test trajectory for data set 2 (in Mumbai) and data set 3 (in Washington, DC) are shown in Figures 5.2 and 5.3 respectively.

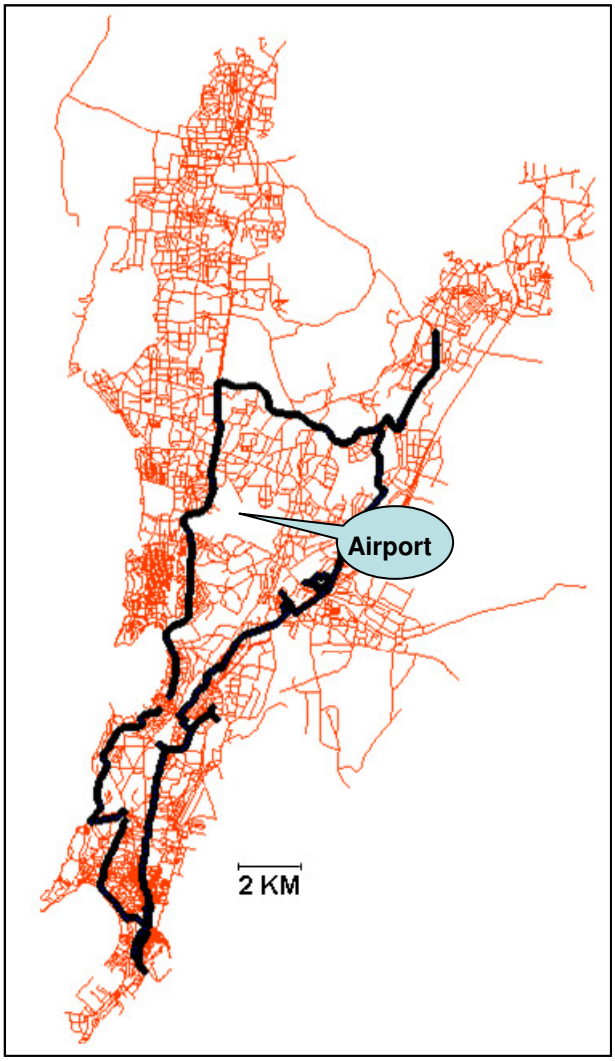


Figure 5.2: Data set 2- test route in Mumbai, India

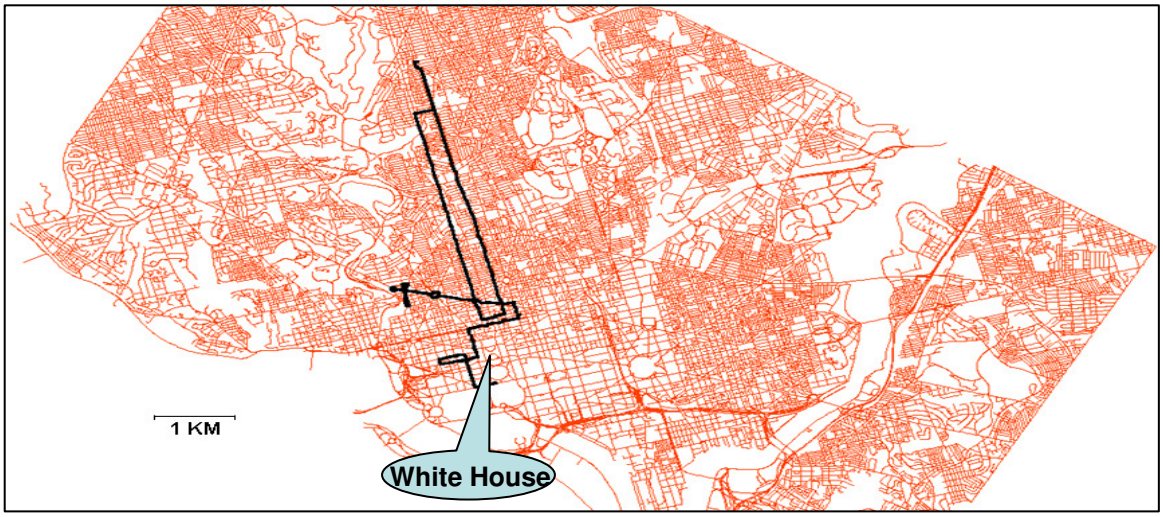


Figure 5.3: Data set 3- test route in Washington, DC

The satellite availability for data sets 2 and 3 are shown in Figure 5.4. The minimum and maximum number of satellites for data set 2 are rescored as 3 and 12 respectively. In case of data set 3 the minimum number of visible satellites are 4 and the maximum number of visible satellites are 11.

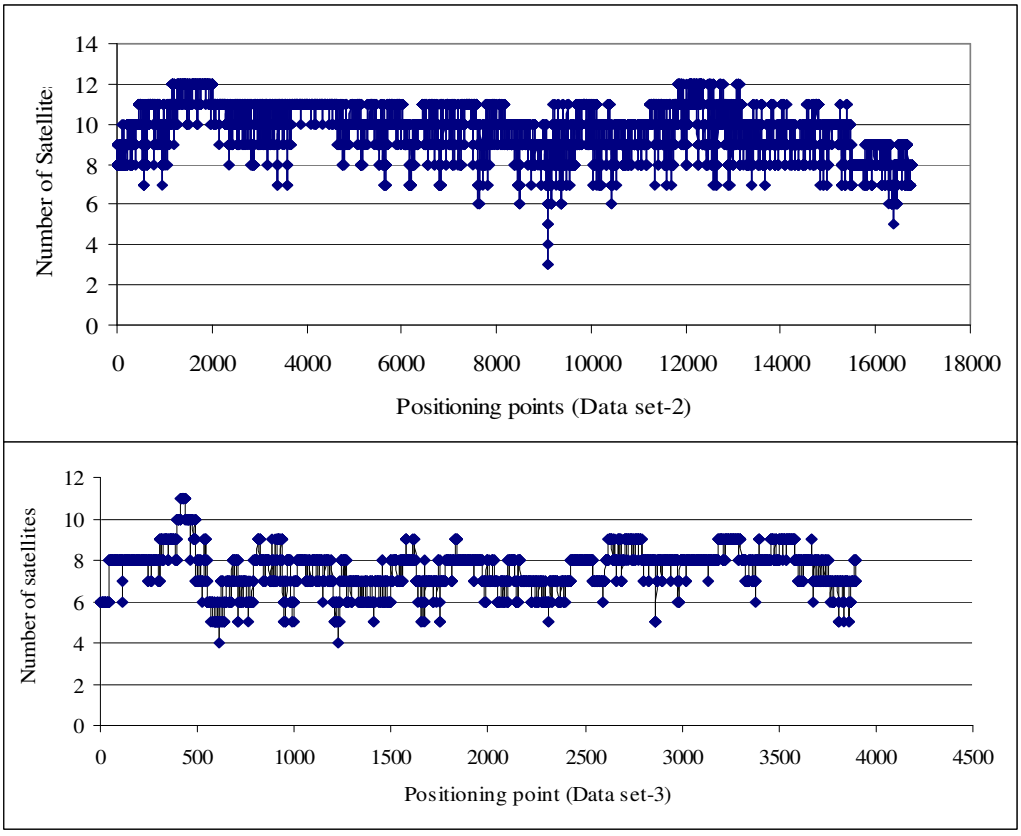


Figure 5.4: Satellite availability

Positioning data set 4 was collected in and around Nottingham, UK on the 1st of May 2009. The test trajectory is shown in Figure 5.5. For the collection of this dataset, a test vehicle equipped with a single frequency high sensitivity GPS receiver and a low-cost gyroscope were used. The vehicle was also equipped with an integrated carrier-phase GPS and high-grade Inertial Navigation System (INS). Accuracy of this equipment (carrier-phase GPS/INS) was found to be better than 5 centimetres over 97.5% of the time in all three coordinate components (Aponte et al., 2009). Therefore, the positioning fixes obtained from GPS carrier-phase observations integrated with a high-grade INS were used as reference positioning points. The test vehicle and the equipment setting is shown in Figure 5.6. and 5.7.

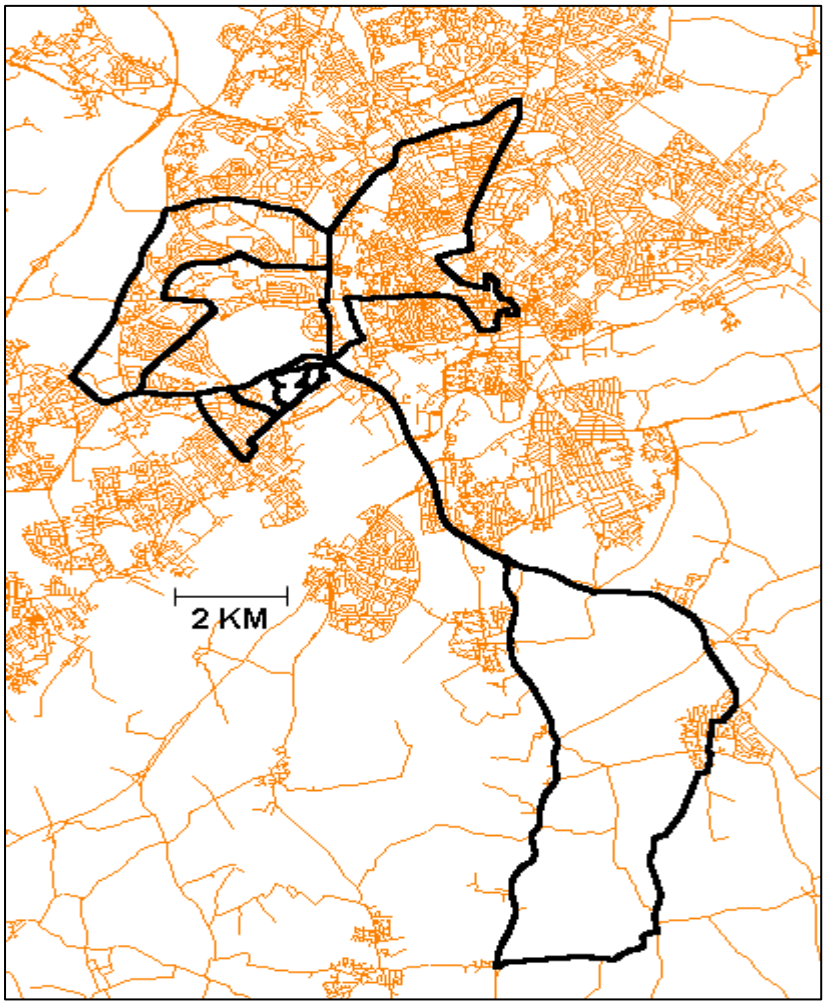


Figure 5.5: Data set 4- test route trajectory in and around Nottingham, UK

Figure 5.6: The vehicle used for positioning data set-4 collection

Figure 5.7: Equipment setup in the test vehicle

Positioning data sets 5 and 6 were obtained from Quddus (2006). Data set 5 is from suburban areas of London (near Reading), and was collected on August 2005, was conducted. A vehicle was equipped with a 12-channel single frequency (L1) high sensitivity GPS receiver, a low-cost MEMS rate gyroscope and the interfaces required to connect to the vehicle speed sensor (odometer). For this data a carrier-phase GPS (i.e., a geodetic receiver consisting of a 24-channel dual-frequency (L1 and L2) with C/A code and P code ranging) was used to obtain the reference (true) trajectory. The total length of the test trajectory is about 46 km (2,040 positioning points). The test route is shown in Figure 5.8. Positioning data 6 was collected in and around London using a 12-channel single frequency (L1) high sensitivity GPS receiver. The length of test trajectory is noted as 148 km.

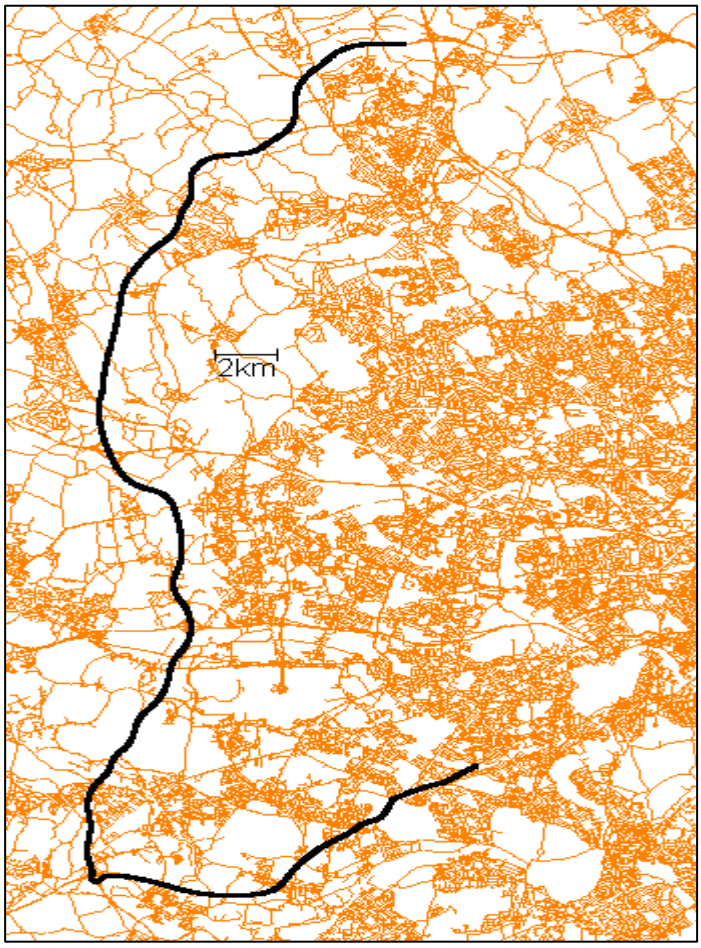


Figure 5.8: Data set 5- Test route in South part of London

5.3 Digital road maps

In this research, three digital road maps, which are obtained from different sources, were used. For all these maps, roads are represented by a single-line-road-network representing the road central line. The details of three digital maps are provided in Table 5.2.

Table 5.2. Digital road map data

Map	Location	Scale	Source
1	United Kingdom	1:2,500	Quddus (2006)
2	Mumbai, India	1:25,000	Indian Institute of Technology Bombay (IITB), India [1]
3	Washington, DC, USA	1:1,250	DC Geographic Information Systems (DCGIS)

[1] Developed by Maples, India; and obtained from IITB

Turn restriction data at junctions are an essential input to the developed map-matching algorithm and the integrity method. Such data are not available with these digital maps. For data sets 2 and 3, the legal turn restriction information was noted while collecting the field data and then crosschecked with the Google earth map; whilst for the other data sets (which were collected in the UK) the turn restriction information was noted from road markings using the Google earth map.

The turn restriction data (for tMM algorithm and integrity method) is stored in the form of turn restriction matrix to consider all the possible turns at a junction point. A four legged junction, where four road segments (1, 2, 3, 4) meet at junction, is explained in Figure 5.9. In this case, a vehicle travelling towards the junction along link 1 is restricted to take a right turn and a U turn. The corresponding vehicle turn restrictions information is represented with '1' in the turn restriction matrix. For example, the value '1' in the first row forth column of turn restriction matrix represents vehicle coming from link 1 can not turn to link 4.

Turn restriction matrix at Junction

Link ID	1	2	3	4
1	1	0	0	1
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

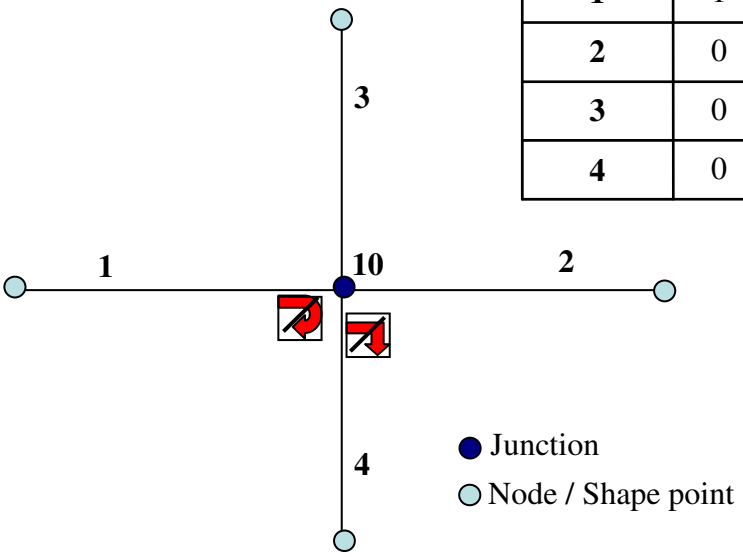


Figure 5.9 Turn restriction at junction

This chapter described the extensive positioning data sets and GIS road maps used in this research. The following chapter describes the development of a weight-based topological map-matching algorithm.

Chapter 6

Development of an Enhanced Topological Map-matching Algorithm

6.1 Introduction

An improved topological map-matching (tMM) algorithm is developed in this chapter. This tMM uses a standard procedure to identify a set of candidate links, then assigns weights for all candidate links - using similarity in network geometry and topology information and positioning information from a GPS system - and selects the link with the highest weight score as the correct road link from a set of candidate road links (see section 3.3 in Chapter 3). As the correct link identification is based on total weight scores (i.e., the sum of heading, proximity, connectivity and turn restriction weight scores), the developed algorithm is known as a weight-based topological map-matching algorithm. As a part of enhancement of the tMM algorithm, an optimisation technique is introduced to identify the relative importance of these weight scores for different operational environments (urban, suburban and rural). The performance of the algorithm, with respect to percentage of the correct link identification and horizontal accuracy, is assessed using field data.

6.2 Map-matching Process

The tMM algorithm requires a range of data input that include: positioning data, topological features of spatial road network data and turn restrictions at junctions. The positioning data includes GPS easting and northing (i.e., X and Y

coordinates), vehicle heading (i.e., vehicle movement direction with respect to the North direction), and vehicle speed data in m/sec. Digital map data includes node data and link data. The node data consists of node number, node easting and northing (i.e., X and Y coordinates). The link data consists of link ID, start node and end node of each link. The turn restrictions information, at each junction, is given in a matrix form (see section 5.3).

A simple flowchart of the proposed tMM algorithm is shown in Figure 6.1. 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 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 point to the selected link, the algorithm checks three criteria for matching the next position point:

(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 vehicle's 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 a downstream junction or not. If the vehicle is far from a downstream junction then this positioning fix is also assigned to the previously map-matched road segment. 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 identified link gives the location of the vehicle on that link.

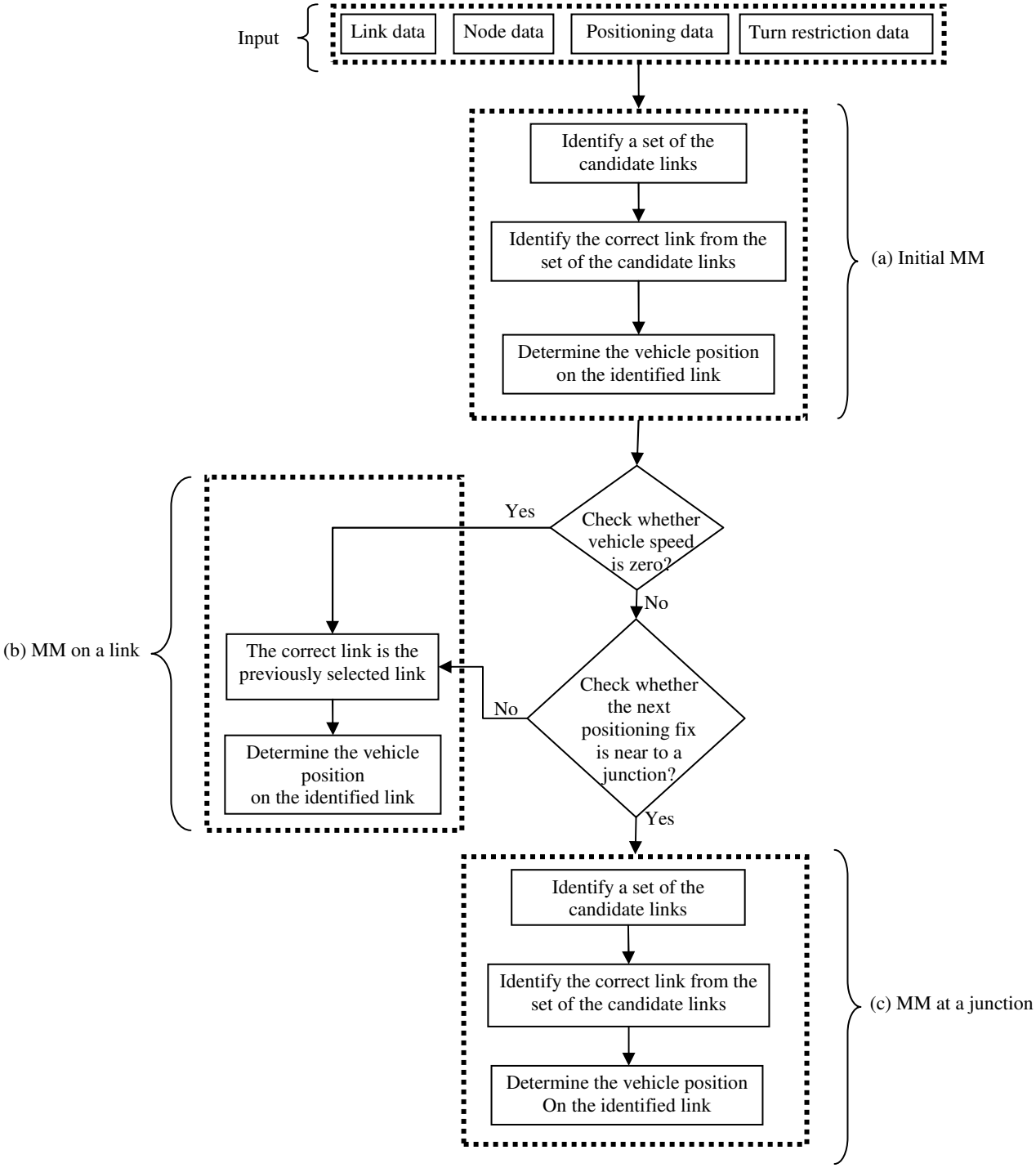


Figure 6.1: A flow-chart representing the enhanced tMM algorithm

6.3 Initial MM process

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 mis-matching 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.

6.3.1 Identification of candidate links

Firstly, the algorithm creates an error circle 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), Ochieng et al. (2004b) and Quddus (2006). All the links that are either inside the error bubble or touching the error bubble are considered as the candidate links for the first positioning fix.

In Figure 6.2, points *A*, *B*, *C*,…, *I* are referred to as node points; points a, b, c, d and e are referred to as shape points; and points P_1 and P_2 are positioning fixes. The scenarios that need to be investigated to identify the candidate links are: (1) All links inside the error circle (link A-a, A-c). (2) Links intersect with the error circle at one point (i.e., one node/shape point is inside the circle and other node/shape point is outside the error circle). Links A-H, A-d, a-b and c-I fall under this scenario. (3) Links intersect with the error circle at two points but both

node/shape points are outside the error circle (i.e., link H-I). (4) Links tangent to the error circle (i.e., link I-b). Finally, for the positioning fix P_1, candidate links are A-a, A-d, A-c, A-H, a-b, c-I, H-I, and I-B.

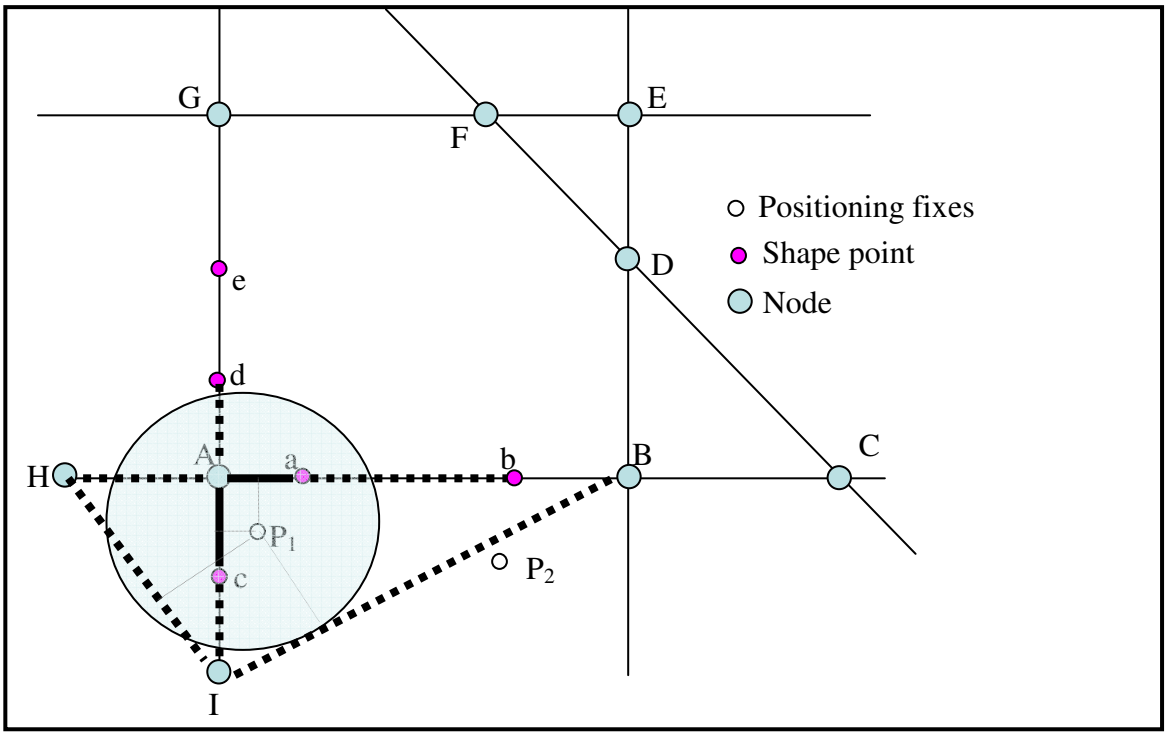


Figure 6.2: Candidate link identification

As explained above, candidate link identification falls into four major cases.

Case A: Road segments those are completely inside the error circle

Case B: Road segments intersect the error circle at one point

Case C: Road segments intersect an error circle at two points but the start and end nodes are out side the error circle

Case D: Road segments tangential to the error circle

Each of these is briefly discussed below:

Case A: Road segment completely inside the error circle

Assuming positioning coordinates (X and Y coordinates) are h and k; and link start node coordinates are (x_1, y_1) and link end node coordinates are (x_2, y_2), The

distance between (h, k) and (x_1, y_1) and between (h, k) and (x_2, y_2)' using the following equation:

$$d_i = \sqrt{(x_i - h)^2 + (y_i - k)^2} \qquad (6.1)$$

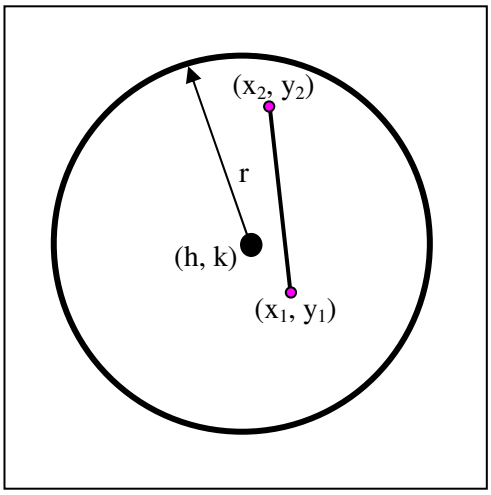


Figure 6.3: Road segment inside an error circle

If the distance between (h, k) and (x_1, y_1) is d_1 and between (h, k) and (x_2, y_2) is d_2; road segments, that are completely inside the error circle (see Figure 6.3), are identified by checking the condition: $d_1 < r$ and $d_2 < r$.

Case B: Road segment intersects the error circle at one point

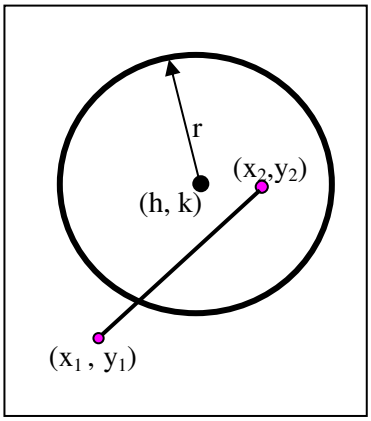


Figure 6.4: Road segment intersects an error circle at one point

If a road segment intersects the error circle at one point, as shown in the above Figure 6.4, then the following condition need to be satisfied: 'd_1< r and d_2> r' or 'd_2< r and d_1> r'

Case C: Road segments intersect the error circle at two points but the start and end nodes are outside the error circle

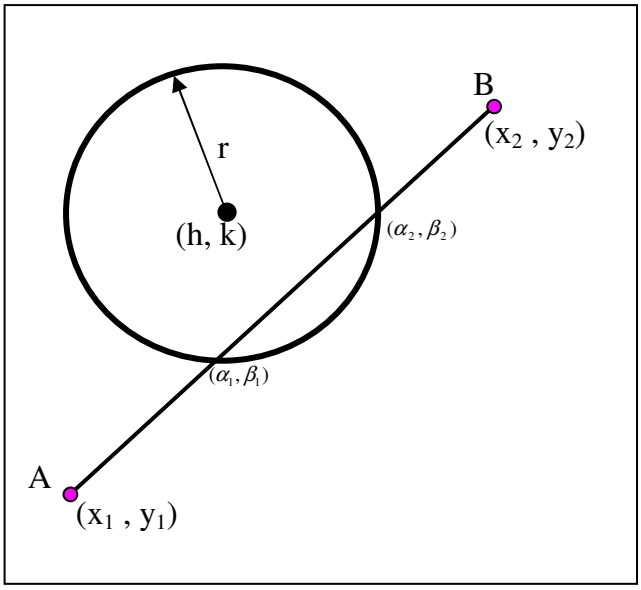


Figure 6.5: Road segments intersect an error circle at two points

If the differences in the x and y coordinates between the start node (x_1, y_1) and the end node (x_2, y_2) of a link A-B is Δx and Δy respectively, then

$$\Delta x = x_1 - x_2 \qquad (6.2)$$

$$\Delta y = y_1 - y_2 \qquad (6.3)$$

The equation of the circle of radius *r* and centre (*h, k*) is:

$$(x - h)^2 + (y - k)^2 = r^2 \qquad (6.4)$$

The equation of the line passing through point A (x_1, y_1) and B (x_2, y_2) is:

$$y = mx + c \qquad (6.5)$$

Where

$$m = \frac{\Delta y}{\Delta x} \tag{6.6}$$

$$c = \frac{x_1 y_2 - y_1 x_2}{\Delta x} \tag{6.7}$$

The points of intersection between equations (6.4) and (6.5) can be expressed as:

$$\alpha = \frac{m^2 k + h - mc}{1 + m^2} \pm \frac{\sqrt{r^2 - g^2}}{\sqrt{1 + m^2}} \tag{6.8}$$

$$\beta = \frac{m^2 k + mh + c}{1 + m^2} \pm \frac{m\sqrt{r^2 - g^2}}{\sqrt{1 + m^2}} \tag{6.9}$$

Where

$$g = \left(\frac{(k - mh - c)}{\sqrt{1 + m^2}} \right) \tag{6.10}$$

The expression ($r^2 - g^2$) in equation (6.8) or (6.9) can be used to see whether the line intersects the circle.

If ($r^2 - g^2$)>0 and '$x_1 \leq \alpha \leq x_2$ or $x_2 \leq \alpha \leq x_1$' or '$y_1 \leq \beta \leq y_2$ or $y_2 \leq \beta \leq y_1$' then the line intersects the circle at two points

Case D: Road segment that is tangential to the error circle

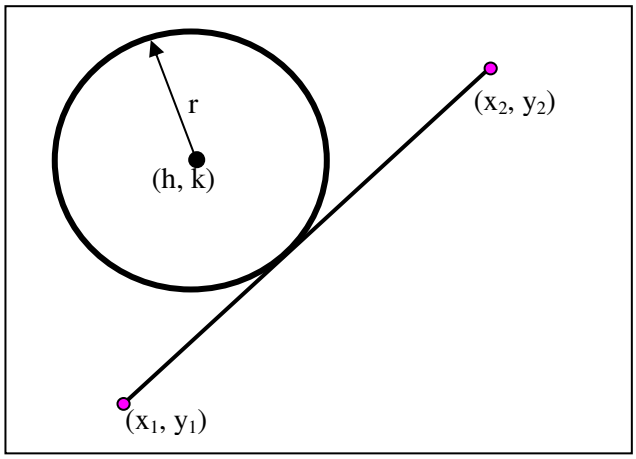


Figure 6.6: Road segment is tangent to an error circle

If the term ($r^2 - g^2$) in equation 6.8 and 6.9 is equal to zero, then the line is a tangent to the error circle.

6.3.2 Identification of the correct link among candidate links

The main task of any MM algorithm is to select the correct link among the candidate links. For the first positioning fix, topological information such as link connectivity and turn restrictions cannot be used because user location on the network is unknown. In this study, for the first positioning point, only heading and proximity weight are considered. A GPS receiver provides heading data for the first positioning fix based on the last stored position fix.

Among the candidate links, greater weight should be given to a link that is in-line with the vehicle movement direction. For instance, as shown in Figure 6.7, for a positioning point P vehicle heading from a navigation sensor is noted as 270°, link direction from the north for link a-b and a-c are 230° and 265° respectively. The values of heading difference between the vehicle movement direction and the link (a-b and a-c) direction are 40° and 5° respectively. Hence, the heading difference is less for link a-c relative to link a-b. Therefore, more importance should be given to link a-c compared to link a-b.

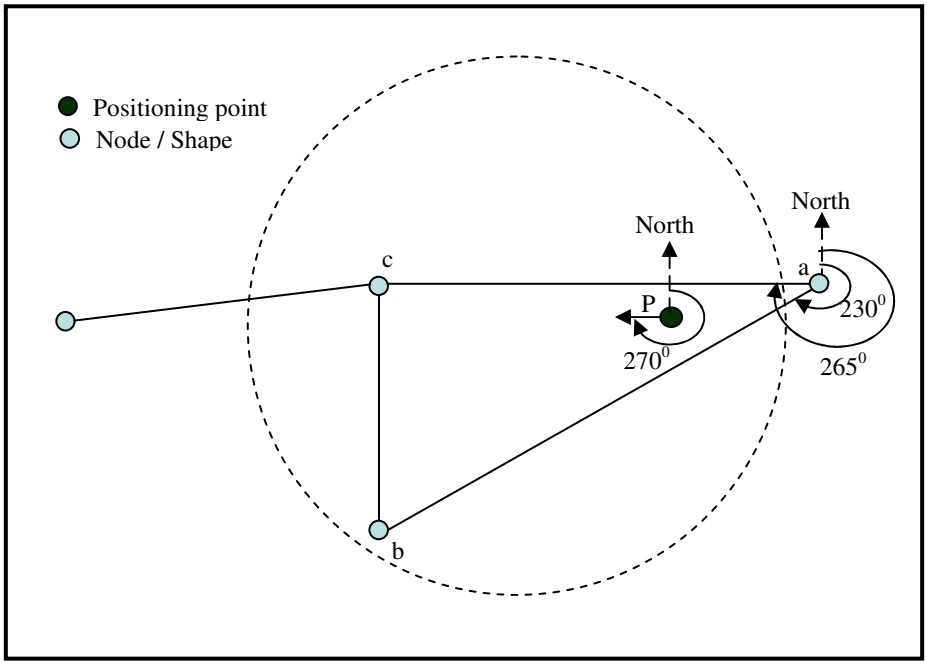


Figure 6.7: Heading weight

Heading weight is considered as a cosine function of angle between the vehicle movement direction and the link direction (as suggested by Greenfeld, 2002) and shown in equation 6.11. This is to ensure that if the difference in angle is small, the heading weight is large and vice versa.

$$W_h = H_w f(\theta) \qquad (6.11)$$

Where

W_h denotes the weight for heading

H_w is the heading weight coefficient

$f(\theta) = \cos(\Delta\theta)$

$\Delta\theta = \Delta\theta'$ if -180°<= $\Delta\theta'$ <= 180°

$\Delta\theta = 360 - \Delta\theta'$ if $\Delta\theta'$ >180°

$\Delta\theta = 360 + \Delta\theta'$ if $\Delta\theta'$ <-180°

In which $\Delta\theta' = \theta_1 - \theta_2$

θ_1 = Heading of each candidate link with respect to the North

θ_2 = Vehicle heading 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. In Figure 6.8, from positioning point 'P' perpendicular distances to candidate links a-b, b-c, a-c are d_1, d_2 and d_3 respectively. The perpendicular projection of positioning point 'A' falls outside the candidate link c-d. This case distance 'D' is taken as the sum of the distances between start node of the link to perpendicular projected point (d_4) and the perpendicular distance (d_5).

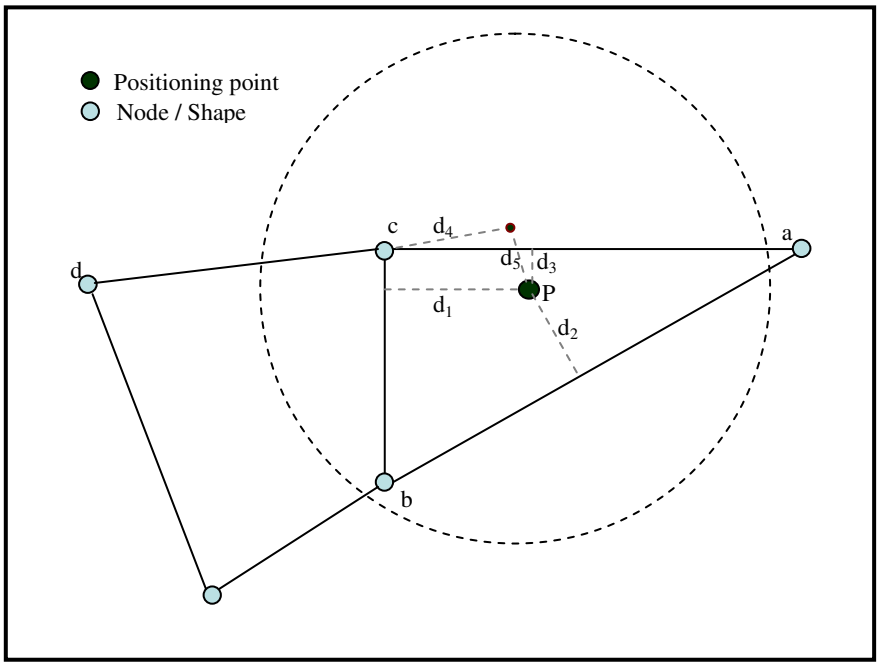


Figure 6.8: Weight for proximity

From Figure 6.9, the perpendicular distance from positioning point P(h, k) to a link A-B with the start node coordinates as (x_1, y) and the end node coordinates as (x_2, y_2) is:

$$D = \frac{h(y_1 - y_2) - k(x_1 - x_2) + (x_1 y_2 - x_2 y_1)}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}} \qquad (6.12)$$

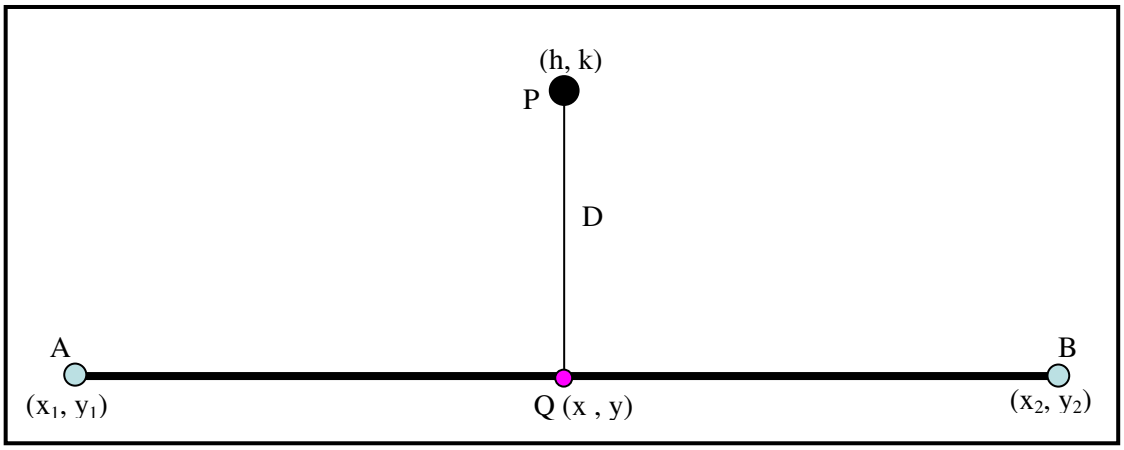


Figure 6.9: Perpendicular distance from positioning point to link

The weight for proximity (W_p) varies linearly with the perpendicular distance.

$$W_p = D_w f(D) \qquad (6.13)$$

Where

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

D_w is the proximity weight coefficient

If the positioning fix falls on the link (i.e., D = 0) the proximity weight parameter, $f(D)$, is consider to the highest possible value which is 1; and if the distance between the positioning point and the link is more than or equal to 160m (i.e. $D \geq 160m$), the proximity weight parameter, $f(D)$, is -1. Between 0 and 160m (that is $0<D<160m$) the weight parameter, $f(D)$, varies linearly with the distance.

The Total Weight Score (*TWS*) for the first positioning point is the sum of heading and proximity weights, as shown below:

$$TWS = W_h + W_p \tag{6.14}$$

TWS is calculated for each candidate link and the link with the highest *TWS* is identified as the correct link.

6.3.3 Estimation of vehicle location on the selected link

The above procedure (Section 6.3.2) identifies the correct link among all candidate links based on the maximum total weight (the sum of heading and proximity weights). The next step is to estimate the vehicle location on that selected link. This is achieved by the perpendicular projection of positioning point onto the link. For instance, in Figure 6.9, the estimated coordinates of the positioning point P (h, k) onto link A-B is Q (x, y). The perpendicular projection of point (h, k) is obtained using following equations 6.15 and 6.16.

$$x = \frac{(x_2 - x_1)[h(x_2 - x_1) + k(y_2 - y_1)] + (y_2 - y_1)(x_1 y_2 - x_2 y_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{6.15}$$

$$y = \frac{(y_2 - y_1)[h(x_2 - x_1) + k(y_2 - y_1)] - (x_2 - x_1)(x_1 y_2 - x_2 y_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{6.16}$$

6.4 Map-matching on a link

After successful completion of the initial MM process, the second stage of the tMM algorithm (i.e., MM on a link) starts with checking 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 to a junction using two criteria:

(1) the 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.

The mathematical representation of this check is:

$$d_1 \geq \left(d_2 + d_{threshold}\right) \tag{6.17}$$

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 point and errors with the digital map (i.e., the omission 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. $d_{threshold}$ is considered to have 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 that the vehicle makes a turn.

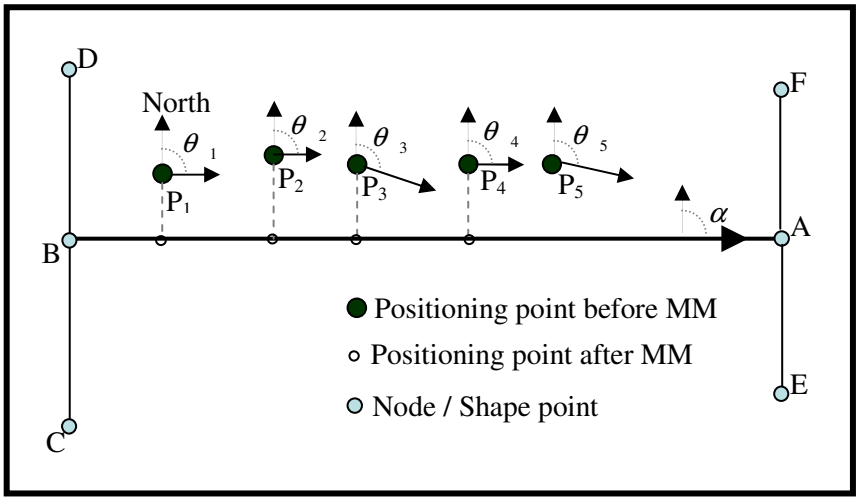


Figure 6.10: RMS value calculation

In Figure 6.10, points P_1, P_2, ..., P_5 are positioning fixes before MM. For instance, positioning fixes P_1 to P_4 are assigned to link B-A. For point P_5, the root mean square (RMS) heading value of the previously map-matched points on the same link is represented using equation (6.18).

$$H_{RMS} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\delta_i^2} = \sqrt{\frac{\delta_1^2 + \delta_2^2 + \delta_3^2 + \delta_4^2}{4}} \qquad (6.18)$$

Where

$$\delta_i = |\theta_i - \alpha|$$

Here, $\delta_1, \delta_2, \delta_3$ and δ_4 are absolute value of angle difference between vehicle heading $\theta_1, \theta_2, \theta_3$ and θ_4 and link 'a-b' direction 'α' with respect to the North.

The mathematical representation of the second check is shown in equation 6.19.

$$h_{RMS}^{i} \geq (\delta_i + h_{threshold}) \qquad (6.19)$$

h_{RMS}^{i} 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 for positioning point *i*. GPS position fixes are less reliable when the speed of the vehicle is less than 3 m/sec (Quddus et al., 2007a; 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 identified as 20m and 5^o respectively. This is accomplished by manually checking different threshold values at junctions using an independent field data set of 1,800 GPS fixes from data set 6. Firstly, few typical junctions are identified where a vehicle took a turn; then at individual junction by randomly giving different distance threshold ($d_{threshold}$) and bearing threshold ($h_{threshold}$) values it is examined at what threshold values the algorithm can correctly recognise when a vehicle reaches a junction. However, the derived threshold values may depend on the quality and the scale of digital map, time interval of each positioning points, and the quality of navigation data from GPS.

If the two checks are 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. Otherwise, the vehicle is at junction.

6.5 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: (1) turn restrictions at junctions and (2) link connectivity. If a vehicle approaches to a junction and is not legally permitted to 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.

The link connectivity weight (W_c) and turn restriction weight (W_t) are given below:

$$W_c = C_w C_c \tag{6.20}$$

$$W_t = T_w C_t \tag{6.21}$$

Where

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

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

C_w and T_w are weight coefficients for link connectivity and turn restriction respectively.

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.

The *TWS* at a junction, which is the sum of four weight scores, is given below:

$$TWS = H_w \cos(\theta) + D_w \left[\frac{(80 - D)}{80} \right] + C_w C_c + T_w C_t \tag{6.22}$$

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, C_c, and turn restrictions, C_t, are specified in such as way 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, C_c and C_t in equation (6.22) 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. The optimisation technique is described in Section 6.6.

6.5.1 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 difference between the TWS for two (or more) links are found to be less than or equal to 1% then the algorithm identifies this as an ambiguous situation. This is because an investigation of our data suggests that a 1% difference in the TWS values correctly picks all ambiguous situations. The algorithm then uses external information including the distance from the last map-matched position to the current map-matched position and compares this with the distance (speed×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 correct 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 the distance travelled by the vehicle within the last time interval, which is one second in our case), and identifies the road segment on which the 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 analysis of an independent data set (data set - 6), this threshold is fixed at 40m in this case.

6.6 Determining weight scores using an optimisation technique

The review of weight based topological map-matching algorithms in Section 3.6.2 indicated that various techniques are available to determine the relative importance of weight scores used in tMM algorithms. Most techniques applied to date are simple and are not based on scientific evidence. For example, Greenfeld (2002) considered three weights (heading, proximity and intersection weights) in the correct link identification process and simply assumed all three weights have equal importance. Quddus et al. (2003) developed an empirical approach to find the relative importance of weight scores in a total weight score (TWS) function. Two weight factors (a and b) were introduced to find the relative importance between heading, proximity and relative position (A_H, A_{PD} and A_{RP}).

The relationships between the three weight parameters (A_H, A_{PD} and A_{RP}) were defined as:

$$A_H = a * A_{PD}$$
$$A_{RP} = b * A_{PD}$$

Where a and b are weighting factors that indicate the strength of any relationship between $A_{\mathrm{H}}, \mathrm{A}_{\mathrm{PD}}$ and A_{RP}. An empirical analysis was carried out by checking the performance of the algorithm with respect to correct link identification, for different combinations of *a* and *b* values. Here, the minimum and maximum values of both *a* and *b* were set as 0.5 and 4.0. In each iteration an increment of 0.5 was added to *a* and *b* values. Quddus et al. (2003) identified the weighting factor for a as 3 and for b as 2. Concluding that actual link identification using a weighting scheme is sensitive to weighting factors (a and b) and suggesting further research to obtain optimal values for the weighting parameters. One of the major disadvantages of this one stage empirical approach is that the derived weight factors are not transferable to different operational environments.

To identify the relative importance of weights optimally in different operational environments, firstly a functional relationship between the weights and the percentage of wrong link identification for each operational environments is required; then, optimal weight scores can be derived by minimising the percentage of wrong link identification in the above functional form. In existing research the functional relationship between weight scores and algorithm performance is not identified; and for all operational environments the same weights are proposed using a one stage empirical approach.

In this research, a two-stage approach is adopted to optimally identify the relative importance of the four weight scores used in the TWS function. Firstly, the functional relationships between the percentage of wrong link identification and four weights coefficients (H_w, D_w, C_w and T_w) for different operational environments are identified. Secondly, the optimal values of weight coefficients are determined by optimising the above functional relationship.

It may be argued that the above two tasks can be done simultaneously in a one stage process. But, in the functional relationship, there are two unknowns: the

regression coefficients (β) and the weight coefficients (H_w, D_w, C_w, and T_w) (see equation 6.24). Solving these two unknowns in order to simultaneously obtain optimal values of both regression coefficients and weight coefficients would be difficult due to the complex functional form and the associated search space for each of the 18 regression coefficients and four weight coefficients. Moreover, the process of identification of relative importance of weight scores is a post-processing technique meaning that the relative importance of weight scores is identified for each operational environment and then they will be used in the MM process in real-time. Therefore, the two-stage optimisation technique adopted in this research does not affect the computational speed of the map-matching algorithm for real-time applications. This is the reason to use a two-stage process in identifying the relative importance of weight scores optimally.

The optimisation test process is shown diagrammatically in Figure 6.11. 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 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.

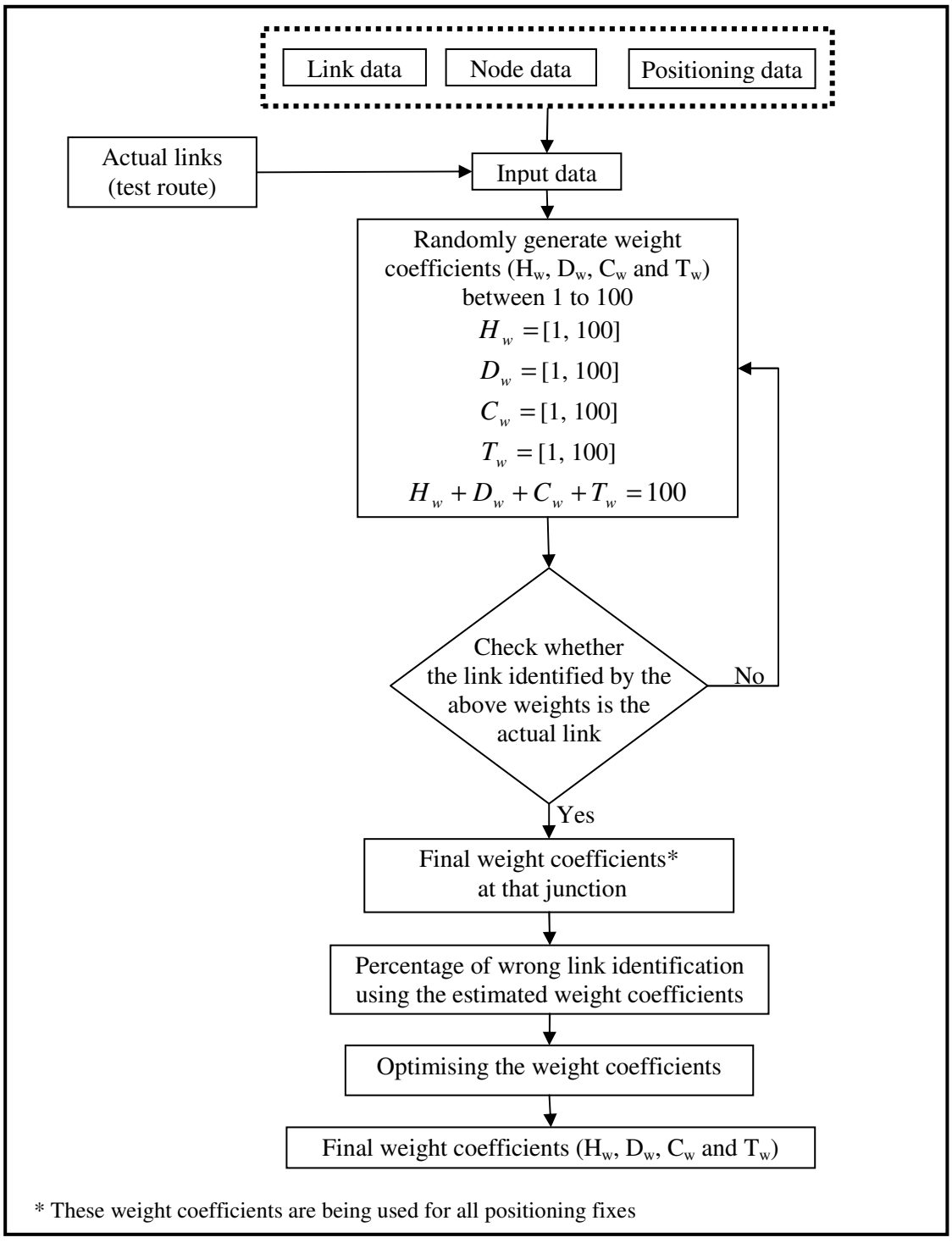


Figure 6.11: Weight coefficient optimisation process

As the other variables, $f(\theta)$, $f(D)$, C_c and C_t, in TWS function 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.

$$MM_{error} = f(H_w, D_w, C_w, T_w) \tag{6.23}$$

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 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:

$$\ln(MM_{error}) = \alpha + [\beta_{h1}H_w + ... + \beta_{t1}T_w] + [\beta_{h2}H_w^2 + .. + \beta_{t2}T_w^2] + \left[\frac{\beta_{h3}}{H_w} + .. + \frac{\beta_{t3}}{T_w}\right] + [\beta_{hd}(H_wD_w) + ... + \beta_{ct}(C_wT_w)] + \varepsilon_i \tag{6.24}$$

where

α is an intercept term.

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

ε_i is the error term.

It should be noted that one of the assumptions of a classical regression model is that there are no exact linear relationships between explanatory variables included in the model (Kennedy, 2008). The statistical phenomenon of very high correlation between two or more independent variables in a model is referred-to as Multicollinearity (Gujarati, 2004; Kennedy, 2008; Maddala and Lahiri, 2009). In this research, as various specifications (such as individual terms,

square terms, inverse terms and interaction terms) are assumed; approximate linear relationship exist among the explanatory variables and this may result in imprecise estimates. It is therefore necessary to examine whether the phenomenon of multicollinearity is a problem. This is carried out by a correlation analysis and a sensitivity analysis. A common method of detecting multicollinearity is through the analysis of correlation matrix (Kennedy, 2008; Maddala and Lahiri, 2009). A suggested rule of thumb is that if the correlation coefficients among the explanatory variables are more than 0.8 there might be multicollinearity among those variables (Gujarati, 2004; Kennedy, 2008). If multicollinearity is detected, there are a range of methods to test whether the multicollinearity affects the assumed specification. Among those methods, the *drop a variable* method is often used to check the effect of multicollinearity (Gujarati, 2004; Kennedy, 2008). Here, one of the collinear variables is dropped and its effect on the remaining highly correlated parameters is examined (Gujarati, 2004; Kennedy, 2008). Both Kennedy (2008) and Gujarati, (2004) indicate that multicollinearity among independent variables is not an issue if the t-statistics of correlated variables exceed 2 in the final model.

Initially, all 18 variables are included in the regression analysis. A correlation analysis is carried to examine the presence of multicollinearity amongst the explanatory variables. Three correlation matrices – one for each operational environment - are developed. The results suggest that there is a high degree of correlation (correlation coefficients greater than 0.94) between linear terms (H_w, D_w, C_w and T_w) and their corresponding squared terms (H_w^2, D_w^2, C_w^2 and T_w^2). For the other variables (inverse terms and interaction terms) the correlation coefficients are less than 0.8.

Considering all the 18 variables, regression analysis is carried out using a step-by-step backward elimination process, at each step one statistically insignificant parameter, based on a t-test (which is used to see whether the coefficient of an independent variable is statistically significant), is removed. Parameters with a t-

value exceeding 1.96 (95%) are retained. The final regression model, with all statistically significant variables, is the optimisation function. Before optimisation, sensitivity tests, to identify whether multicollinearity is a problem, are carried out.

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.

Minimisation

$$MM_{error} = \exp\left\{\alpha + \left[\beta_{h1} H_w + ... + \beta_{t1} T_w\right] + \left[\beta_{h2} H_w^2 + .. + \beta_{t2} T_w^2\right] + \left[\frac{\beta_{h3}}{H_w} + .. + \frac{\beta_{t3}}{T_w}\right] + \left[\beta_{hd}(H_w D_w) + ... + \beta_{ct}(C_w T_w)\right] + \varepsilon_i\right\} \quad (6.25)$$

subject to:

$$H_w + D_w + C_w + T_w = 100$$
$$1 \le (H_w, D_w, C_w, T_w) \le 100$$

Optimisation of equation (6.25) was carried out in MATLAB using the constrained nonlinear minimisation 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.

Part of data set 6 relating to urban, suburban areas and data set 1 rural areas were used to the optimisation process. Table 6.1 shows the best fitting regression models for each area type. The adjusted R^2 estimates the percentage of behaviour of dependant 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 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 may be inappropriate.

Table 6.1: 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 is Heading weight coefficient; D_w is Proximity weight coefficient; C_w is Connectivity weight coefficient; and T_w is Turn restriction weight coefficient.

Model sensitivity analysis:

Among the three models presented in Table 6.1, the first and third models corresponding to the urban and rural operational environments, do not include any squared terms. The second model corresponding to the suburban area

includes two square terms (H_w^2 and D_w^2). Thus, a sensitivity analysis (using the *drop a variable* method) is carried out for this model to see whether multicollinearity is an issue. Table 6.2 presents three scenarios: base model, exclude D_w^2 and exclude H_w^2.

Table 6.2: Regression model sensitivity (suburban area)

Weights	Base model		Exclude D_w^2		Exclude H_w^2	
	Coefficient	***T-stat***	***Coefficient***	***T-stat***	***Coefficient***	***T-stat***
H_w	0.0287	16.82	0.0277	16.61	0.0201	37.34
D_w	0.0233	17.99	0.0201	47.95	0.0215	16.78
C_w	0.00347	4.97	0.00456	8.21	0.00593	11.02
T_w	0.00467	6.72	0.00575	10.4	0.00564	8.21
H_w^2	-0.000115	-5.34	-0.0001	-4.82	--	--
D_w^2	0.0000476	-2.56	--	--	-2.2E-05	-1.21
$1/(H_w)$	1.266	34.3	1.272	34.35	1.194	33.9
$1/(D_w)$	1.137	25.36	1.083	27.29	1.144	24.9
$1/(C_w)$	0.197	6.12	0.157	5.54	0.149	4.69
$1/(T_w)$	0.26	5.94	0.228	5.41	0.209	4.78
(H_w*D_w)	-0.000539	-18.16	-0.0005	-19.84	-0.00044	-18.8
(H_w*T_w)	-0.000069	-2.97	-6.8E-05	-2.91	-3.25E-06	-0.16
(C_w*T_w)	0.00013	5.91	0.000104	5.3	9.24E-05	4.32
Adjusted R^2	0.9970		0.9970		0.9969	

It can be observed from Table 6.2 that after eliminating one of the correlated variables (H_w^2 and D_w^2) from the base model, the results are not significantly different. The sign of the remaining correlated variables (H_w, D_w, C_w and T_w) are not affected. Moreover, the variation in R^2 value is also negligible. Clearly, the t-stats increase showing the presence of a model with collinear variables. However, the significance of the variables is not a problem here because the t-stats of these variables are greater than 2. The sensitivity analysis indicates that the multicollinearity is not a problem in this model. Therefore, the three functional relationships presented in Table 6.1 are considered optimal and are utilised in the optimisation process.

The optimal values of weights for the three operational environments are shown in Table 6.3. 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. The two new weight scores (link connectivity and turn restriction) considered in this research are more important in urban areas where more than 50% of the weight score is allotted to these two new weights. It is expected that the contribution of these two new weights will improve the algorithm performance in terms of correct link identification especially in urban areas. In suburban and rural areas the weight for connectivity (C_w) and weight for turn restriction (T_w) are less 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 positioning fixes is good and the road network is less dense.

Table 6.3: Optimisation Result

Weights	Operational areas		
	Urban	**Suburban**	**Rural**
$\mathbf{H_w}$	39.99	46.24	44.48
$\mathbf{D_w}$	8.13	44.99	53.52
$\mathbf{C_w}$	36.40	4.46	1
$\mathbf{T_w}$	15.48	4.31	1

6.7 Algorithm performance evaluation

In order to evaluate the performance of the enhanced tMM algorithm, part of data sets 1 and 3 were used. The sample size (i.e., number of positioning data points) sets 1 and 3 are 2,814 and 3,600 respectively. The test trajectory for data set 1 (in central London) and data set 3 (in Washington, DC) are shown in Figure 6.12 and Figure 6.13 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 5 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 5 was also used to examine the existing tMM algorithms performance by Quddus (2006), which was illustrated in Table 3.1, and the results are therefore directly comparable.

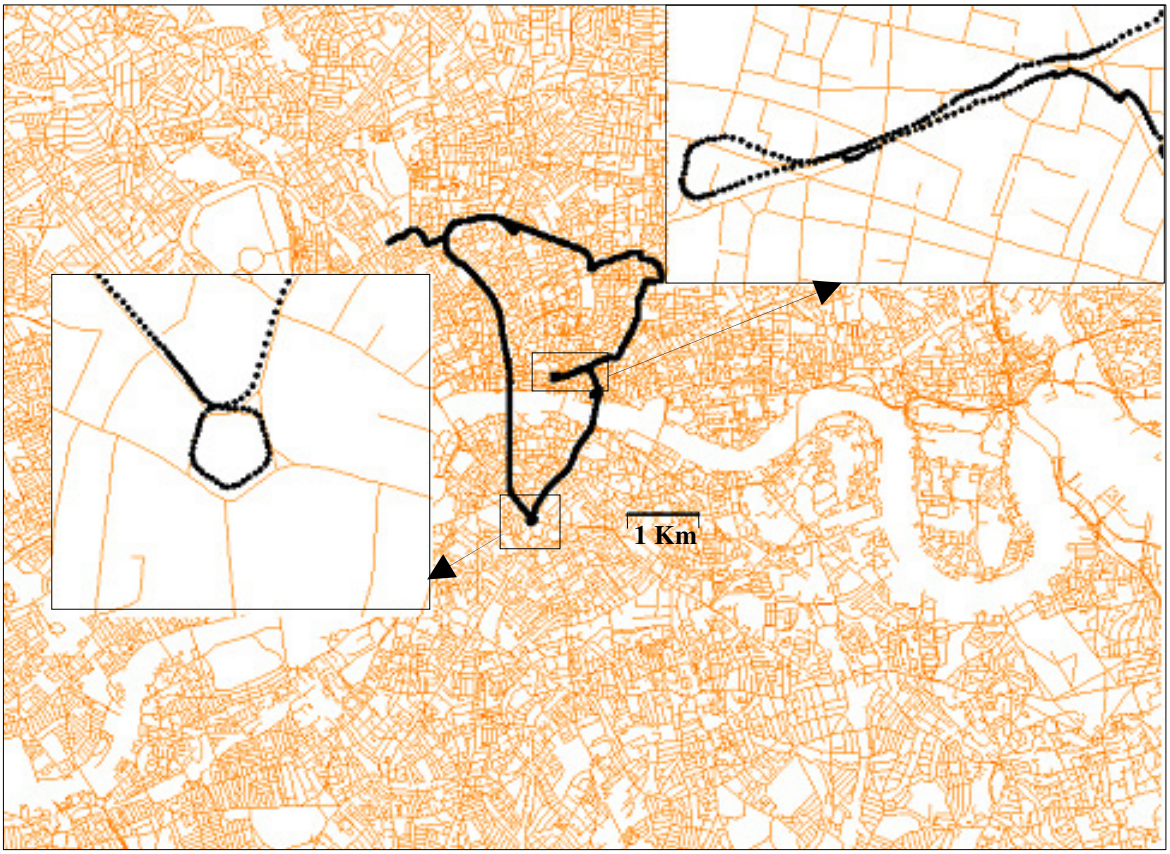


Figure 6.12: Test route in central London

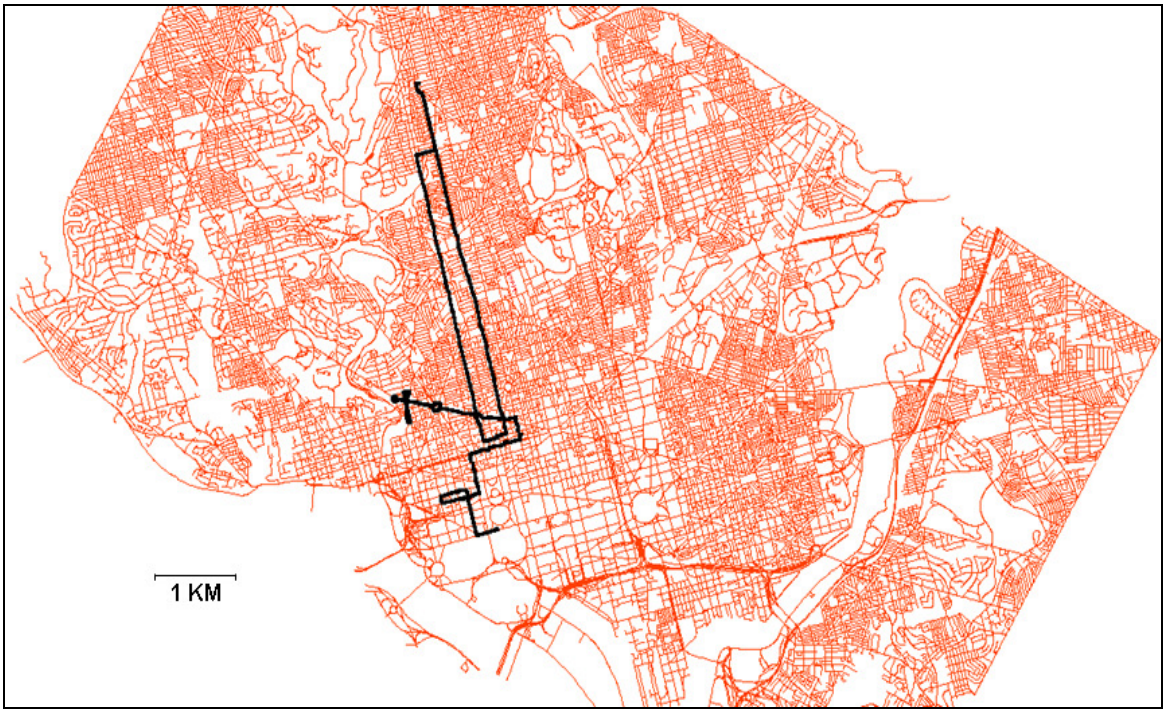


Figure 6.13: Test route in Washington, D.C., USA

The values of the weights from Table 6.3 are then applied for algorithm testing. For data set 1 (urban areas in central London) the enhanced tMM algorithm identified 96.8% of the road segments correctly. In case of data set 3 (urban areas in Washington, DC) the success rate is 95.93%. The test result, with data sets from two metropolitan cities, suggested 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 1GB RAM and 1.46 processor speed). This suggests that the algorithm is suitable for real-time implementation. Figure 6.14 shows a part of the test trajectory along with raw positioning fixes (with star symbols) and map-matched fixes (with round symbols).

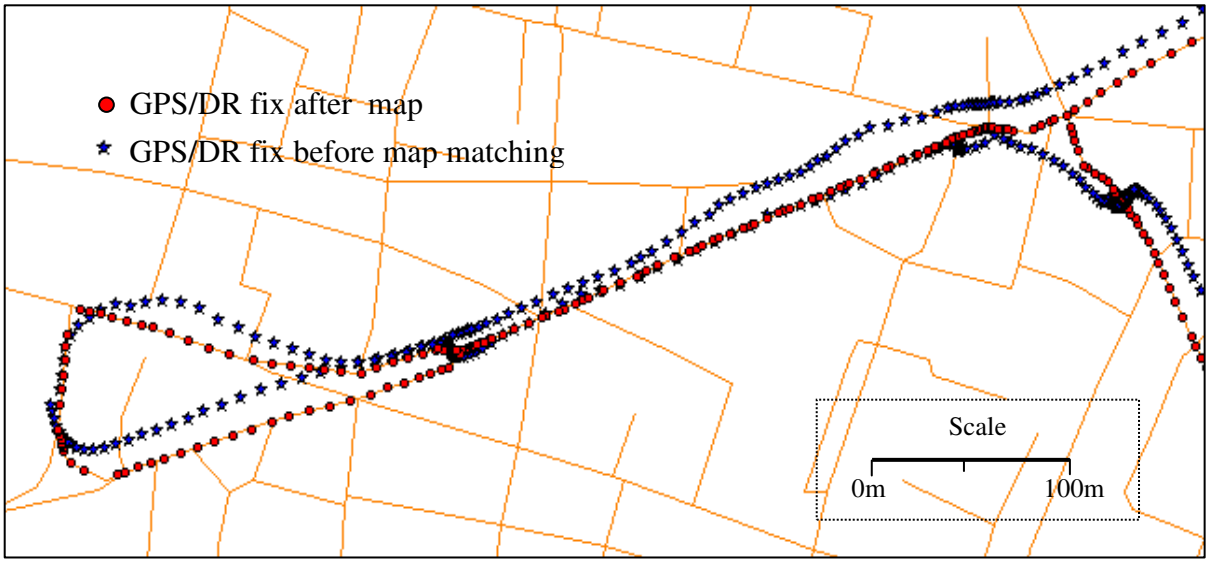


Figure 6.14: A part of test road with map-matched positions

The enhanced tMM algorithm was then applied to the fifth data set (suburban area), which was also used for performance evaluation in terms of both 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 15.44m ($\mu_h + 2\sigma_h$). This outperforms all the tMM algorithms previously tested by Quddus (2006) using this data. The tMM algorithm performance is summarised in Table 6.4.

Table 6.4: Algorithm performance

Data set number (location)	Percentage of correct link identification	Horizontal accuracy ($\mu + 2\sigma$)
1 (London)	96.80%	--
3 (Washington, DC)	95.93%	--
5 (South part of London)	96.71%	15.44m

Mean and standard deviation of the positioning accuracy for data set 5 is shown in Table 6.5. The along-track and cross-track errors were found to be 10.5m ($\mu_a + 2\sigma_a$) and 12.48m ($\mu_c + 2\sigma_c$) respectively.

Table 6.5: Algorithm positioning accuracy

Data set number (location)	Horizontal accuracy (m)		Along-track error (m)		Cross-track error (m)	
	μ_h	σ_h	μ_a	σ_a	μ_c	σ_c
5 (South part of London)	5.64	4.90	3.14	3.68	3.48	4.50

6.8 Summary

In this chapter, a real-time, weight-based topological MM algorithm has been developed to address 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.

All of these new features have contributed to the performance of the algorithm. The link connectivity and turn restriction weights are particularly important in urban areas. 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 9.81m ($\mu + 2\sigma$) in a suburban area of the UK.

This algorithm performs better than existing topological MM algorithms reported in the literature. This topological MM algorithm is fast, simple and efficient and therefore, has good potential to be implemented by industry. However, the performance of a tMM algorithm can further be improved by investigating the causes for the mismatches and modifying the algorithm accordingly. This is carried is by a sequential process of map matching error detection, correction and re-evaluation, which is presented in chapter 7.

Chapter 7

Map-matching error detection, correction and re-evaluation

7.1 Introduction

Although the performance of the topological map-matching algorithm developed in the previous chapter was found to be good, the algorithm incorrectly identified road segments about 4% of the time. Any error associated with either the raw positioning data, the digital map, and the MM process can lead to incorrect link identification. The aim of this chapter is to further improve the algorithm firstly by map-matching error detection, then, using a thematic analysis the identification of a number of strategies to correct these mismatches. This is followed by modifications to the algorithm. Finally, the performance of the tMM algorithm (before and after the improvement) is evaluated using an independent data set.

7.2 Algorithm enhancement methodology

A step-by-step process of the enhancement of the tMM algorithm, which mainly consists of error detection, correction and re-evaluation process, is shown in Figure 7.1.

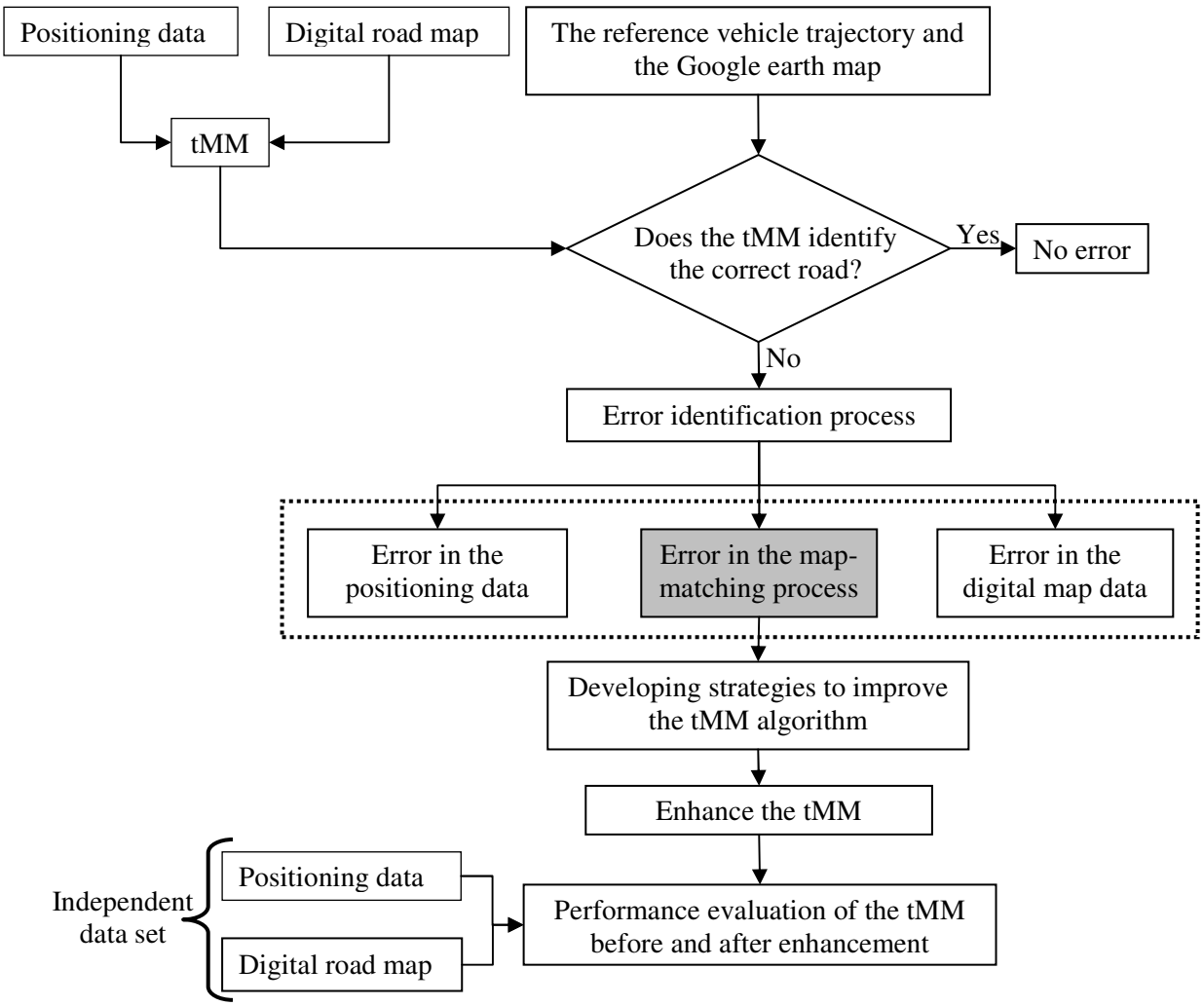


Figure 7.1: MM error detection, correction and performance re-evaluation

The output of the tMM algorithm provides a road segment on which a vehicle is travelling. If the road segment selected by the tMM algorithm is the actual (true) road segment for that particular positioning point, then it is assumed that there is no error in link identification. Otherwise, the map-matched point falls under the mis-matching case. This process was conducted for all 62,887 positioning points to identify the main reasons for mis-matching involving errors in the positioning data or the digital map or the map-matching process. The error detection process is followed by the development of different strategies to improve the performance of the tMM algorithm and then to modify the algorithm

accordingly. Finally, the performance of the enhanced tMM algorithm (before and after enhancement) is examined using an independent positioning data set. The following section describes the mis-matching identification process.

7.3 Error identification process

The error detection process is carried out by classifying the mismatches due to errors in the positioning data, the digital map, and the map-matching process. The quality of raw position points is decided based on the number of visible satellites and the value of the Horizontal Dilution of Precision (HDOP) representing the quality of positioning solution. The UK Google earth satellite image is used as a base map to check errors in the GIS road maps (examples include, topological and geometric errors, missing links, extra links and digitisation errors). If the quality of a raw positioning point is good and no errors are identified in the digital map, then the reason for mis-matching is assumed to be an error in the map-matching process. Each of mismatches due to the MM process is examined carefully to identify which part of the algorithm (i.e., the candidate link identification, the total weight score calculation and the consistence checks) caused the mis-matching. Though the overall performance of the tMM algorithm is good, there are some mismatches in situations where the vehicle took a 'U' turn at junctions or at complex road configurations (i.e., Y junctions, roundabouts and parallel roads etc). The main cause of each mis-matching case was identified through careful observation and critical judgement.

Table 7.1 shows the results from the tMM algorithm on three separate data sets collected in from UK, USA and India.

Table 7.1: Reasons for mismatches

	UK (Data set 1)	Washington, DC, USA (Data set 3)	Mumbai, India (Data set 2)
Positioning data sample size	42,231	3,900	16,756
Mismatches due to positioning sensor error	472 (33.9%)	47 (29.6%)	159 (11.6%)
Mismatches due to digital map error	238 (17.1%)	27 (17.0%)	985 (71.7%)
Mismatches due to MM process errors	683 (49.0%)	85 (53.4%)	230 (16.7%)
Total number of mismatches	1,393	159	1,374

The tMM algorithm has 96.7%, 95.9% and 91.8% success rate of correct road link identification with data set 1, 3 and 2 respectively. From 62,887 map-matched positioning points, a total of 2,926 mismatches were discovered. In order to find out the reasons (i.e., errors in positioning data, map-matching process and digital map) of mis-matching, each mis-matching case was individually examined. In Table 7.1, the percentage contribution of mismatches due to the corresponding error is provided in parenthesis. It can be seen that about half of the mismatches in data sets 1 and 3 are due to errors in the map-matching process. The similar results for data set 1 and 3 suggest that the algorithm may be transferable. In case of data set 2, the major contribution of mismatches is digital map errors. This is due to the fact that the Mumbai GIS map has more missing links and digitisation errors. Clearly, the map-matching errors are predominant where the digital maps are good.

7.4 Enhancement of the MM algorithm

From 62,887 map-matched positioning points, a total 998 of mismatches were found due to errors in map-matching process. The tMM algorithm failed to identify the correct road segment particularly in complex road configurations (such as Y junctions, roundabouts and parallel roads). Examples of mis-matching cases at a roundabout and at a Y junction are shown in Figures 7.2 and 7.3 respectively.

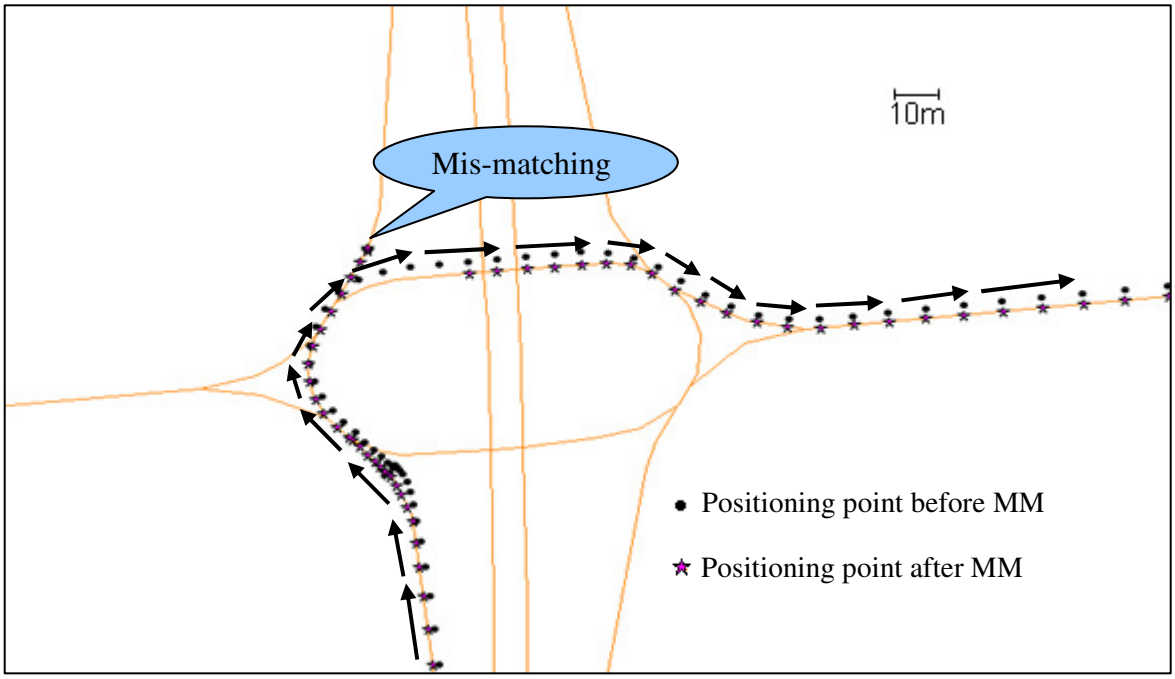


Figure 7.2: Mis-matching at roundabout

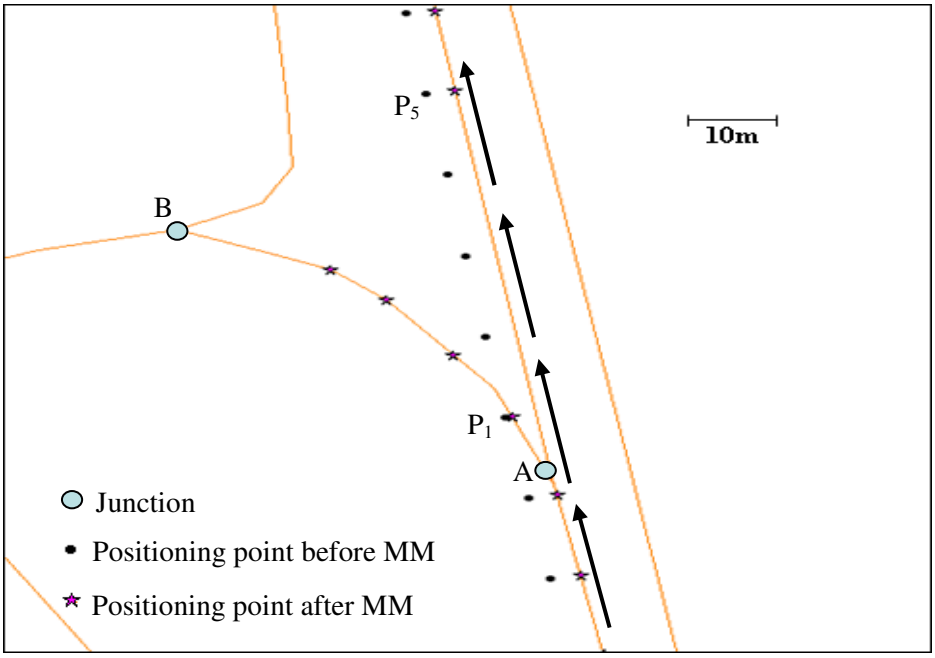


Figure 7.3: Mis-matching at Y junctions

In Figure 7.3, at junction *A*, for positioning fix P_1 the algorithm identified the wrong road segment (i.e., link *A*-*B*). However, to avoid continuous mis-matching, the algorithm measures the distance between the raw positioning point

and the map-matched positioning point. If the distance is more than the allowable limit (a threshold value), then the algorithm reinitiates the map-matching process. In this case, for positioning point P_5, the algorithm reinitiates the process and chooses the true road segment. Two other thresholds (a distance threshold and a heading threshold) were also used to check whether a vehicle is near a junction or not. As mentioned before, in the tMM algorithm, the correct road segment selection at a junction is based on the total weight score (TWS) which is the sum of four weights: heading, proximity, link connectivity and turn restriction. Further, the relative importance of these weights varies with the operational environments. Therefore, any mistake in the identification of the operational environment may lead to an error in the total weight score. This may subsequently lead to wrong road link identification. Moreover, the threshold values used in the algorithm may influence the correct road segment identification.

After careful observation of all mis-matching cases due to errors in map-matching process, the following three strategies were identified to enhance the tMM algorithm:

1. re-examining the optimal weight scores using a Genetic Algorithm (GA) optimisation technique;

2. using a lookup table to identify the weight scores corresponding to the operational environment (e.g. urban, suburban and rural); and

3. re-estimating the thresholds used in the algorithm

7.4.1 Optimisation of the weight scores using a Genetic Algorithm (GA)

Previously, a gradient search method was used to determine the optimal values of weight scores used in the map-matching process (see Section 6.6). In the gradient search minimisation problem there is a possibility that the optimisation

stops at a local rather than a global minimum (Michael et al., 2007). In order to ascertain whether the optimisation has reached a global minimum, Konar (2005) suggests the user of a more refined method such as a Genetic Algorithm (GA). Therefore, a GA based optimisation algorithm is used to determine the relative importance of different weights.

A GA is a stochastic search technique to find exact solutions to both constrained and unconstrained optimisation problems (Callan, 2003). Unlike, a conventional gradient search method, which starts with a single point and progresses to the optimal solution, the GA technique starts with an initial set of random points covering the entire range of possible solutions (Russel and Norvig, 2002; Callan, 2003; Karray and DeSilva, 2004). These random points are generally called as population. The individual random point/solution in the population is known as a chromosome. The function value (in the GA terminology, *fitness value*) is calculated for each chromosome. For the following generation, a new set of chromosomes are evolved, from the current population, based on given selection rules[7], crossover rules[8] and mutation rules[9] (MathWorks, 2008). For each new generation the new chromosomes (in the GA terminology, *child*) is identified using a pair of chromosomes in the previous generation. The process of regeneration stops when it reaches a termination conditions (i.e., pre-defined maximum number of iterations or change in fitness value or maximum time limit).

In the previous optimisation test the sample size for the rural operational environment was low with only 40 junctions. Here this is increased to 186. A new objective function (i.e., the relationship between the map-matching error and the weight coefficients) for the rural operation environment is identified. The detailed description on the derivation of the objective function is provided in

[7] Process of selecting individual chromosomes (parents) based on fitness value to develop next generation.
[8] Rules for combining two parents for new chromosomes (children) for next generation
[9] Random changes to individual parents to form children for the next generation.

Section 6.6. The objective function for the rural operation environment is identified as:

$$MM_{error} = 0.29H_w + 0.2D_w + 0.35C_w + 0.28T_w - 0.00079H_wD_w + 0.00037D_wT_w \quad (7.1)$$

Where H_w, D_w, C_w, and T_w are the weight coefficients for heading, proximity, link connectivity and turn restriction respectively. The adjusted R^2 value of the above model is found to be 0.98. To re-estimate the optimal weight scores, the Matlab GA toolbox is used. For the other two operational environments (i.e., urban and suburban), optimisation functions are the same as the functions provided in Section 6.7.

In the GA optimisation process, a population size of 20, which were uniformly distributed with lower range of 1 and higher range of 100 were used. Stopping criteria is selected as 5,000 generations. After approximately 1,500 generations, the function value (fitness value) shows that the function achieves the global optimal values. The optimal values of heading, proximity, connectivity and turn restriction weight scores for the three operational environments, using gradient search method and a Genetic Algorithm, are illustrated in Table 7.2.

Table 7.2: Optimal weight scores using gradient search method and GA

Weight coefficient	Gradient search method			Genetic Algorithm		
	Urban	Suburban	Rural	Urban	Suburban	Rural
H_w	39.99	46.24	44.48	37.15	46.42	42.37
D_w	8.13	44.99	53.52	8.06	43.76	55.63
C_w	36.4	4.46	1	35.85	4.29	1
T_w	15.48	4.31	1	18.94	5.53	1

It is noticeable from Table 7.2 that the weight scores from the two optimisation techniques are very similar. The main difference is the GA gives more weight to T_w, less weight to H_w in urban areas and a slight rise in T_w weight in suburban area. The optimal weight scores obtained using the GA are thought to be more

reliable as those are solved as a global optimisation problem. Therefore, the second set of optimal values (from GA) is included in the enhanced tMM algorithm.

7.4.2 Use of a lookup table to identify the operational environment

Table 7.2 shows that the relative importance of the weights varies with the operational environment. The algorithm should identify the operational environment in which the vehicle is travelling, and should select the corresponding weights from the weight matrix shown in Table 7.2. The identification of the operational environment (whether the vehicle is in an urban, suburban or rural area) can be based on the complexity of road network, land-use data, building height data, etc. For instance, the road network in an urban area is denser (i.e., more junctions and roads per unit area) than that of in a suburban or a rural area. Since, land-use data and building height data are not easily available, the identification of operational environment can be determined by a threshold which is a function of the total length of the road network and the number of junctions per unit area:

$$T_{OE} = f(L, N) \tag{7.2}$$

where,

T_{OE} is the threshold for operational environment,

L is the total length of road network (in km) within a given area and

N is the number of junctions in that area.

An empirical analysis was conducted, using a national level GIS road network, to identify the T_{OE} which can be used to detect the operational environment on which a vehicle is travelling. Firstly, using the UK Google Earth map, a sample of rural areas was identified in the network. Random points are selected within the road network of rural area, and a circle of radius 200m is drawn around each of the points. After performing a sensitivity analysis a circular area of radius

200m was found to successfully establish the threshold for the identification of operational environment. The total length of all road segments (L) and the number of junctions (N) within that circular area are calculated. This procedure is repeated for 300 different random points in the network. This procedure provides a set of L and N for that particular operational environment. A factor, which is the ratio of N and L, is identified and its mean (μ) and standard deviation (σ) are calculated. The same procedure is repeated for the urban and suburban environments. The means and standard deviations of the factor for urban, suburban and rural operational environments are shown in Figure 5. Here, μ_U, μ_S and μ_R are the mean values and σ_U, σ_S and σ_R are the standard deviations of the factor for urban, suburban and rural operational environments respectively.

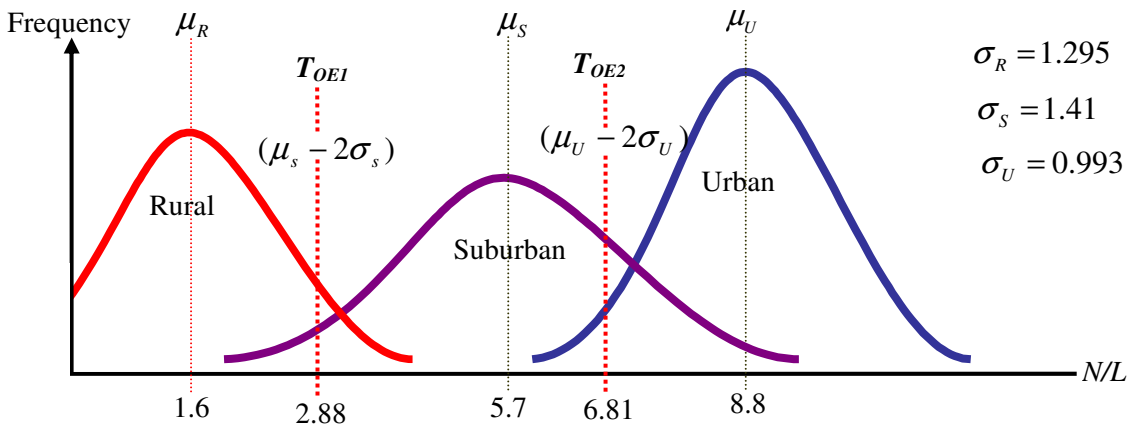


Figure 7.4: Thresholds for operational environment identification

As stated before, the identification of an operational environment is critical to a MM algorithm. Figure 7.4 shows that the values of the means and standard deviations of the N by L ratios for different operational environments are over-lapped meaning that it is not easy to derive threshold values for the identification of an operational environment. In the case where a vehicle travels through a mixed urban setting (i.e. partly urban and partly suburban), the algorithm should recognise that the vehicle is in an urban area so that more stringent weight coefficients are selected for the map-matching process. This is also true for the

case of mixed suburban area (i.e. partly suburban and partly rural) in which the weight coefficients for the suburban area should be employed.

Assuming the ratio of *N* and *L* follows a normal distribution and based on the above argument, two threshold values (T_{OE1} and T_{OE2}) are identified as:

$$T_{OE1} = \mu_S - 2\sigma_S \tag{7.3}$$

$$T_{OE2} = \mu_U - 2\sigma_U \tag{7.4}$$

if $T_{OE} \leq T_{OE1}$ then it is assumed that the operational environment is *rural*

if $T_{OE1} < T_{OE} < T_{OE2}$ then it is assumed that the operational environment is *suburban*

if $T_{OE} \geq T_{OE2}$ then it is assumed that the operational environment is *urban*

Where, T_{OE} is the calculated threshold in the map matching process.

The mean of N/L for urban, suburban and rural operational environments are identified as 8.8, 5.7 and 1.6 respectively; and the corresponding (σ) values are 0.993, 1.41 and 1.29 respectively. The T_{OE1} and T_{OE2} were found to be 2.88 (i.e., 5.7-2*1.41) and 6.81 (i.e., 8.8-2*0.993). In the map-matching process if the calculated threshold (T_{OE}) using the same area (i.e., 200m radius circle), is less than T_{OE1} (2.88) a vehicle is in a rural area; if it is more than T_{OE2} (6.81) vehicle is in an urban area, if it is between these two values then the vehicle is in a suburban area.

7.4.3 Checking threshold values used in the algorithm

In the topological MM algorithm three different threshold values are used. They are distance threshold (D_t), heading threshold (H_t) and a threshold value for a

consistency check (C_t). The former two threshold values are used to identify whether the vehicle is near a junction. The tMM algorithm checks whether a vehicle reaches a junction using two criteria: (1) checking distance from the previously map-matched vehicle position to the downstream junction; in which a distance threshold (D_t) is used. (2) checking the vehicle heading with respect to the previously matched link direction; in which a heading threshold (H_t) is used. The third threshold value is used in a consistency check (i.e., whether the distance between the raw position point and the map-matched position on the link is large). Every time the algorithm checks the distance between the raw positioning point and the map-matched positioning point. If it exceeds a certain limit (i.e., the threshold value) then the algorithm re-initiates the process. Previously, these three thresholds (D_t, H_t and C_t) were identified, using 1800 positioning points, by manually checking whether the algorithm selects 'map-matching at junction' process when the vehicle reaches a junction and whether the algorithm can recognise the continuous mismatches in order to reinitiate the process to identify the correct link. D_t, H_t and C_t thresholds were identified as 20, 5 and 40 respectively. Now, as part of the algorithm correction process, thresholds (D_t, H_t and C_t) were re-estimated using a positioning data set of 2,814 positioning points collected in Central London (part of data set 1 in Table 5.1). An experiment was conducted with different possible threshold values and its corresponding percentage of correct link identification was measured. The D_t, H_t and C_t values with minimum error in correct link identification are identified as 23, 5 and 37 respectively.

7.5 Performance of The Enhanced tMM Algorithm

An independent dataset (sample size 5,256 positioning points) collected in and around Nottingham, UK was used to re-evaluate the performance of the enhanced map-matching algorithm. This positioning data is a part of data set 4 in Table 5.1. A reference (true) trajectory was obtained from a carrier phase GPS receiver integrated with a high-grade Inertial Navigation System (INS).

Accuracy of this equipment (carrier-phase GPS/INS) was found to be better than 5 centimetres over 97.5 percent of the time in all three coordinate components (Aponte et al., 2009). The total length of the test trajectory is 55.9 km. The test trajectory is shown in Figure 7.5.

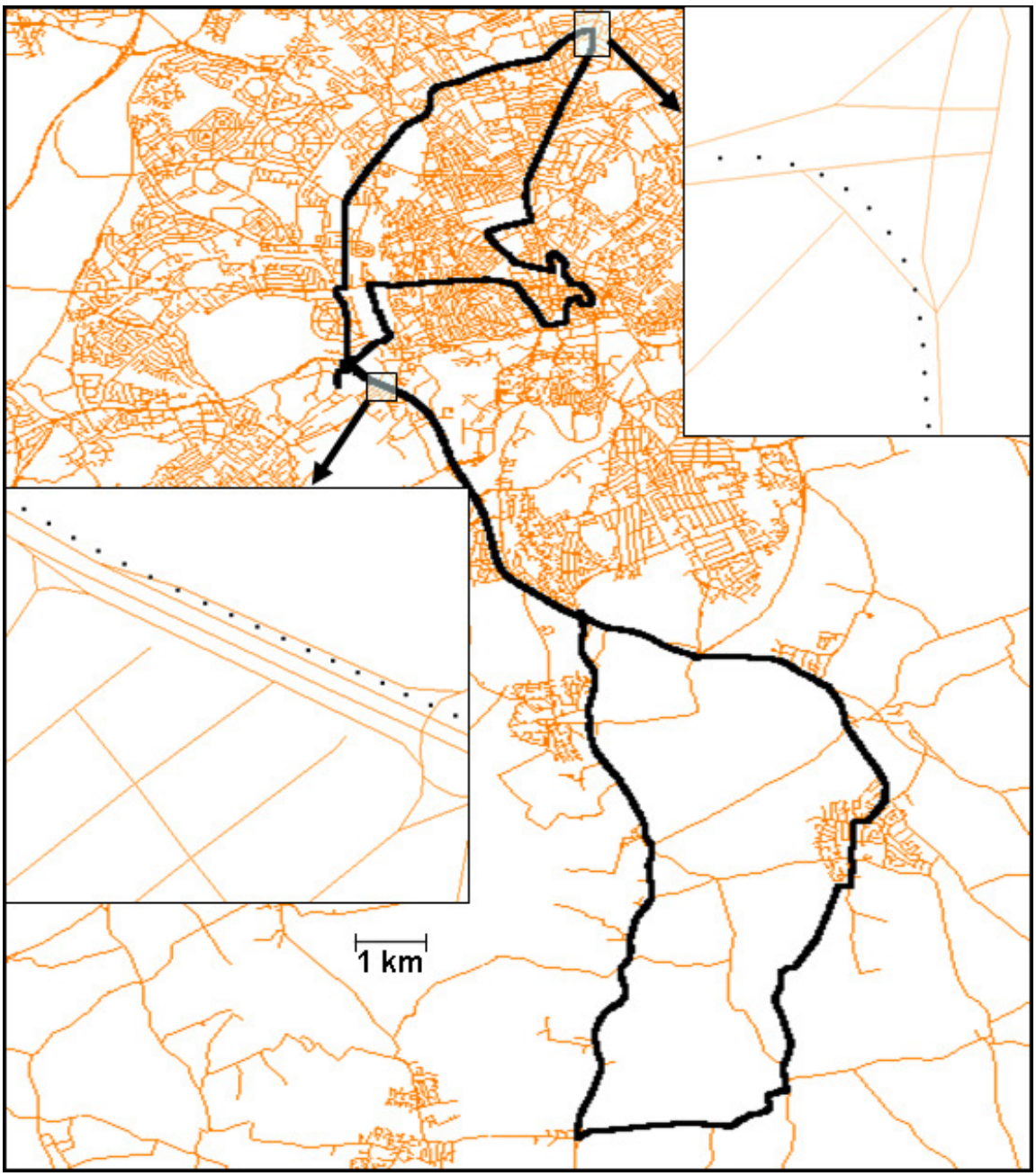


Figure 7.5: Test trajectory in and around Nottingham, UK

The improvement in the correct road link identification also affects the horizontal accuracy. The highly accurate positioning data from carrier phase GPS/INS enabled us to check the algorithm's horizontal positioning accuracy. The algorithm's performance for each enhancement strategy, with respect to the original (base) tMM algorithm is shown in Table 4.

Table 7.3: Enhanced algorithm performance

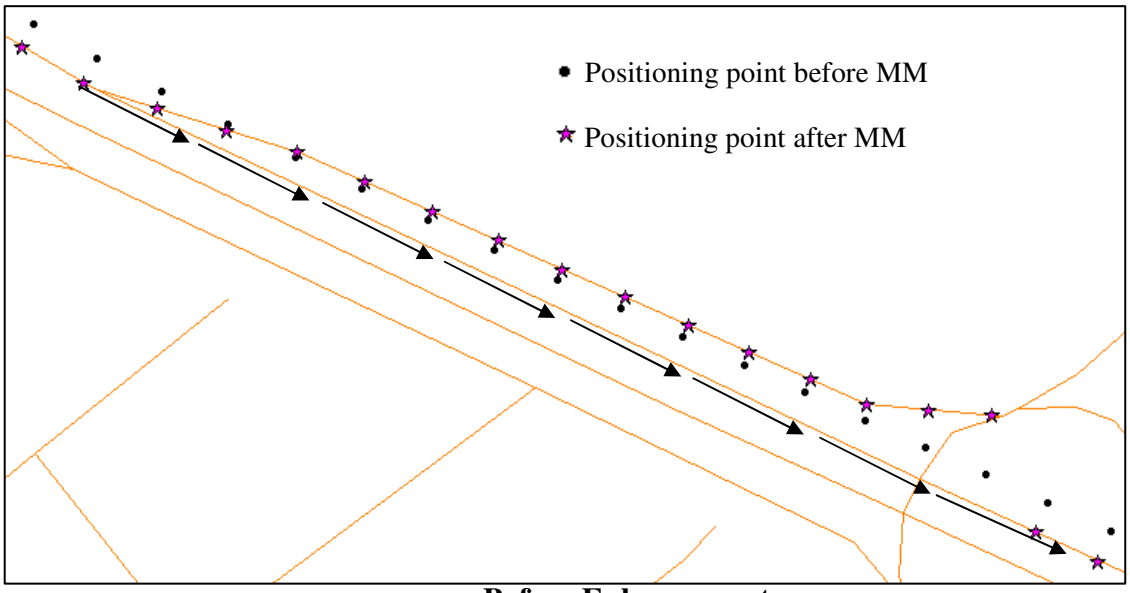
Enhancement	% of correct link identification	Horizontal accuracy (m)		Along-track error (m)		Cross track error (m)	
		Average (μ_h)	SD (σ_h)	Average (μ_a)	SD (σ_a)	Average (μ_c)	SD (σ_c)
Base tMM algorithm	96.5	4.33	2.83	2.16	1.74	3.29	2.86
1: New weight scores	96.7	4.31	2.78	2.14	1.69	3.28	2.85
2: Lookup table	97.7	4.20	2.48	2.12	1.53	3.20	2.60
3: Threshold values	96.5	4.33	2.83	2.16	1.74	3.30	2.87
1 and 2	97.8	4.19	2.47	2.11	1.52	3.19	2.59
1 and 3	96.7	4.31	2.79	2.15	1.69	3.29	2.85
2 and 3	97.7	4.20	2.48	2.12	1.53	3.20	2.60
1, 2 and 3	**97.8**	**4.19**	**2.47**	**2.11**	**1.52**	**3.19**	**2.59**

SD- Standard deviation

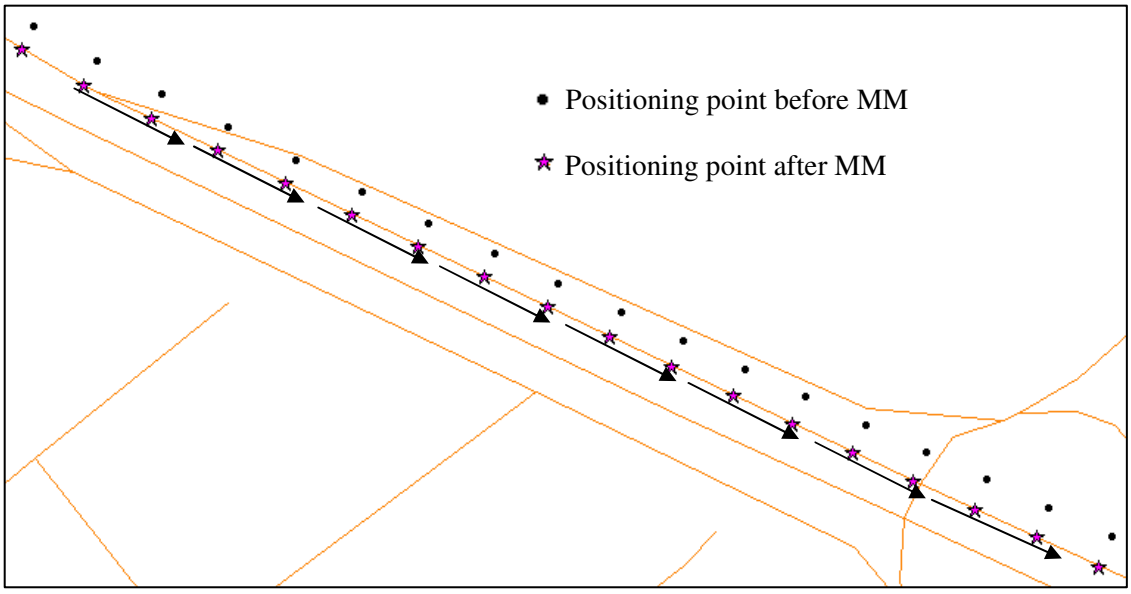
The original algorithm correctly identifies the road links 96.5% of the time. However, when all three improvements are included in the final algorithm which increases the correct link identified to 97.8%. An improvement of 1.3% in correct road link identification is noticed. The second improvement (i.e., using a lookup table to identify the operational environment) contributes most to the improvement in the algorithms performance. The first enhancement (i.e., re-examining the optimal weight scores using a Genetic Algorithm) slightly improves the algorithm and the third enhancement (re-estimating the thresholds used in the algorithm) did not contribute in improving the algorithm performance. The horizontal accuracy of the enhanced algorithm is identified as 9.1m ($\mu_h + 2\sigma_h$) with along track and cross track errors as 5.2 m ($\mu_a + 2\sigma_a$) and 8.4m ($\mu_c + 2\sigma_c$) respectively.

Some typical parts of the test route (a motorway junction and a dense urban road network) are shown in Figures 7.6 and 7.7. Here, map-matched positioning

points before and after enhancement are illustrated. Bold circle symbols represent raw positioning point and the star symbols are map-matched points. The arrows show the actual vehicle travelled path with the direction of movement.



Before Enhancement



After Enhancement

Figure 7.6: Algorithm performance on Motorways

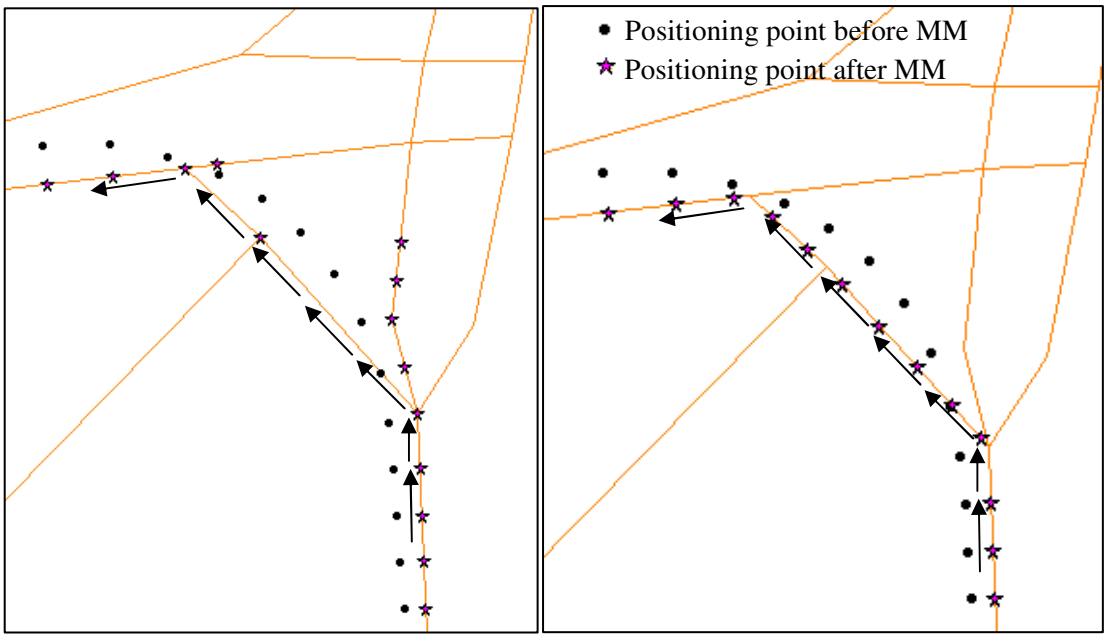


Figure 7.7: Algorithm performance in a dense urban area

7.6 Summary

An improvement process of a weight-based topological map-matching algorithm was presented in this chapter. The enhancement process included: mis-matching detection, improvement strategies identification, algorithm modification and performance re-evaluation. After map-matching using extensive positioning data sets, all mismatches due to positioning data errors, digital road map and map-matching process were identified. In positioning data sets 1 and 3 (for which the digital maps are good) about 50% of the wrong road link identification was due to the map-matching process. After further examining the mismatches due to the map-matching process, three strategies were identified to enhance the tMM algorithm. They were:

1. Re-examining the relative importance of weight scores using a Genetic Algorithm optimisation technique;
2. Using the lookup table for operational environment identification; and
3. Re-estimating threshold values.

The performance of the algorithm was re-evaluated using an independent positioning data. The enhanced algorithm succeeded 97.8% of the time in correct link identification with an horizontal accuracy of 9.1 m $(\mu_h + 2\sigma_h)$. Before the enhancement the success rate was 96.5% with 10.0m $(\mu_h + 2\sigma_h)$ horizontal accuracy. This suggests that the proposed modifications were rational as they improved the performance by 1.3% in correct link identification and 0.9m horizontal accuracy. The introduction of lookup table contributed to the greatest improvement.

Chapter 8

Development of a Map-aided integrity method

8.1 Introduction

Existing studies of land vehicle integrity monitoring have concentrated on either the integrity of raw positioning information obtained from GNSS/DR or the integrity of the map-matching process (see Chapter 4). Considering these three sources of errors together may lead to a better outcome. Moreover, taking the complexity of the road network (i.e., operational environment) into account can further improve the integrity process. Because, if a vehicle is in a simple road network (e.g. rural operational environment) a simple MM algorithm can identify the correct road segment even with an erroneous raw positioning point.

In this chapter an attempt has been made to develop an integrity method for land vehicle navigation, which considers all the sources of error with GPS raw positioning data, spatial road network map and the map-matching process simultaneously. Integrity of raw position points is measured using a weighted least squares method[10]. The complexity of road network is also considered in the integrity monitoring process.

The following section provides the description of the input data in the integrity method. Then the developed integrity method is outlined. This includes the

[10] See section 3.2.3

monitoring of GPS integrity, identification of operational environment, map-matching integrity and the derivation of an integrity scale using two knowledge-based sugeno fuzzy inference systems (FIS).

8.2 Input Data

The data required for the integrity method 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; turn restriction data for junctions and 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 metres per second, and error variance for heading and speed. Satellite data include satellite number and satellite X, Y, and Z coordinates in Earth-Centred Earth-Fixed (ECEF) coordinate system, satellite clock bias, satellite elevation and azimuth angles, measured ranges; and user X, Y, and Z position in ECEF coordinate system. Additional information on map scale, Horizontal Alert Limit (HAL), false alarm probability (P_{FA}) and missed detection probability (P_{MD}) are also inputs to the system.

A GIS map of 1:2500 scale was used in this study. The selection of HAL is subjective as this depends on the required horizontal positioning accuracy (DOT, 2004). For most location based ITS services (except a few safety critical services such as collision avoidance system), the horizontal accuracy ranges from 5m to 50m (DOT, 2004; Sheridan, 2001; Feng and Ochieng, 2007; and Quddus, 2006) and the HAL is 2.5 times the horizontal accuracy (DOT, 2004; Feng and Ochieng, 2007). In this study, the HAL is chosen to be 15m and this equals to the required horizontal accuracy of 6m, with which most ITS services can be supported (DOT, 2004; Sheridan, 2001; Feng and Ochieng, 2007; Quddus, 2006). The P_{FA} and the P_{MD} are defined to be 0.001 and 0.00001 as suggested by Feng and Ochieng (2007). That means the likelihoods of a false alarm and a

missed detection occurrence are 1 in 1,000 and 1 in 100,000 positioning points respectively.

The range residuals data for each positioning point is one of the inputs to measure the integrity of the raw positioning points. The range residual is the difference between the measured pseudorange and the predicted pseudorange. As the data on the range residuals are not directly available from the GPS receiver used, range residuals are calculated using a least squares method. In the following section, a brief description of the procedure to calculate the range residuals is provided.

8.2.1. Range residual calculation using a least squares method

The range residual information is used to in the estimation of integrity of raw positioning data (see section 8.3.1 for details). The range residual for satellite *i* can be defined as the difference between the measured pseudorange and the predicted pseudorange of the satellite *i*. The diagrammatic representation of range residuals is shown in Figure 8.1.

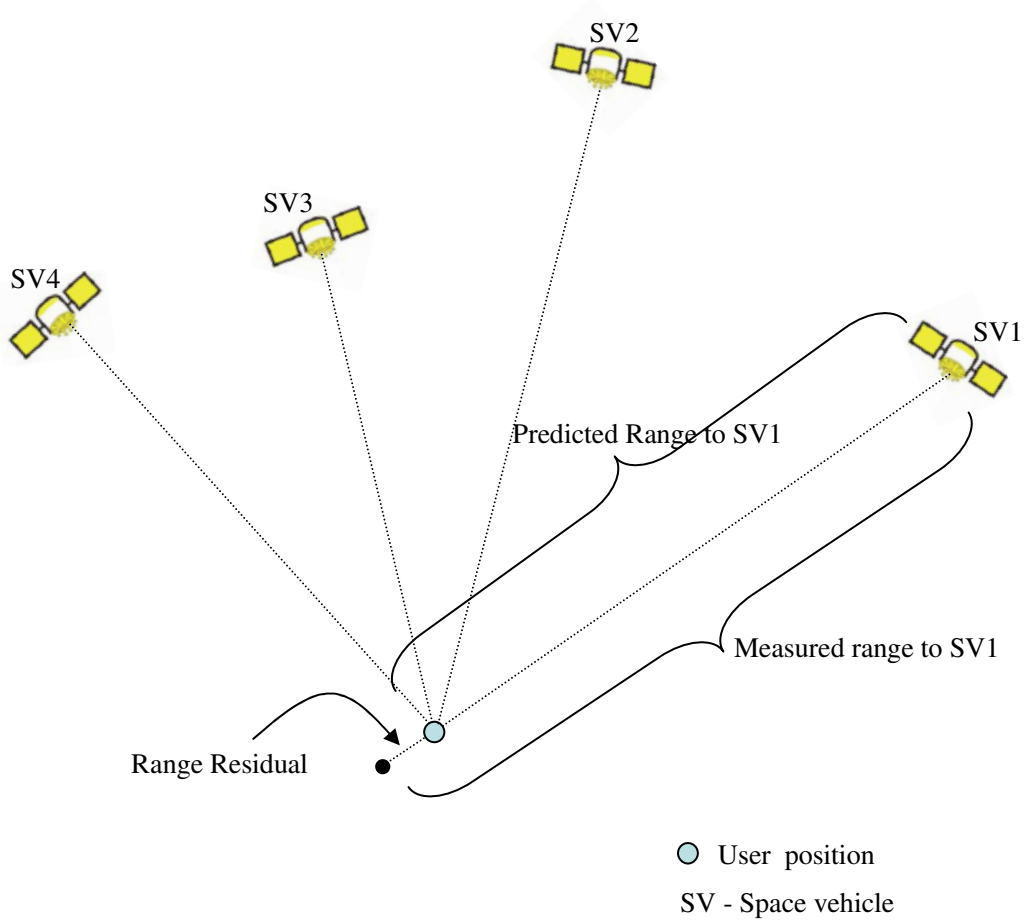


Figure 8.1: Range residuals (Source: modified from Lee, 1992)

The step-by-step procedure of least square method range residual estimation is provided here. This is a traditional method of estimating range residual matrix.

A least squares method of range residual estimation step-by-step procedure:

In the least square method the input data are:

(1) Satellite number

(2) Elevation angle of each satellite

(3) Measured ranges from each of the satellites (this is obtained from the GPS receiver)

(4) Satellite X_s, Y_s, Z_s coordinates and satellite clock offset.

Step 1: Initial user vehicle estimation:

The measured pseudorange calculation is as follows (Parkinson and Spilker 1996; Leeuwen, 2002):

$$\rho_m^i = \rho^i + C(ion^i + tro^i + Sc^i) \tag{8.1}$$

Where

ρ_m^i is the measured pseudorange from satellite *i*

ρ^i is the range from a satellite *i* (this is the input data 3 from step-1) in metres

C is the speed of GPS signal (i.e., speed of light)= 299792458 m/sec

ion^i is the ionospheric delay in seconds for satellite *i*

tro^i is the tropospheric delay in seconds for satellite *i*

Sc^i is the satellite clock offset for satellite *i*

The propagation of a GPS signal in the ionosphere and the troposphere differs with the speed of satellite signal (the speed of light). So the delay terms ion^i and tro^i are considered in the measured pseudorange calculation. In equation (8.1), both ion^i and tro^i need to be estimated and Sc^i is input 4 .

Calculation of ionospheric delay:

Ionospheric delay is calculated using the Klobuchar model (Klobuchar, 1987; Klobuchar, 1991). The delay calculation for a single frequency receiver is as follows:

$$ion^i = [1 + 16(0.53 - E^i)^3]5 * 10^{-9} \tag{8.2}$$

Where ion^i is the ionospheric delay in sec

E^i is the elevation angle of satellite *i*

Calculation of tropospheric delay:

The Calculation of tropospheric delay is followed by Tsui (2000); and Parkinson and Spilker (1996)

$$tro^i = \left[\frac{2.47}{\sin(E^i) + 0.0121}\right](1/2.99*10\text{^}8) \qquad (8.3)$$

Where tro^i is the tropospheric delay in sec

E^i is the elevation angle to satellite *i*

<u>Step 2</u>: calculation of user/GPS receiver position (X_u, Y_u, and Z_u coordinates):

This is an iterative process using a least squares method. The procedure followed is as given in Parkinson and Spilker (1996) and Jwo (2005) an shown below:

(1) In the first iteration, assume a user position (X_u^1, Y_u^1 and Z_u^1 coordinates) and receiver clock offset C^1. Here, the initial user position and the clock offset are assumed as zero.

(2) Calculate the predicted/estimated range to each satellite using equation (8.4)

$$\rho_e^i = \sqrt{(X_s^i - X_u^1)^2 + (Y_s^i - Y_u^1)^2 + (Z_s^i - Z_u^1)^2} + C^1 \qquad (8.4)$$

Where

ρ_e^i is the predicted or estimated pseudorange to satellite *i*

X_s^1, Y_s^1 and Z_s^1 are the satellite position

X_u^1, Y_u^1 and Z_u^1 are the user position in the first iteration

C^1 is the receiver clock bias in the first iteration

(3) calculate range residual (R_i^1) for iteration 1 to satellite *i*. This is accomplished by using equation (8.1) end equation (8.4)

$$R_i^1 = \rho_m^i - \rho_e^i \tag{8.5}$$

(4) Calculate the observation matrix (G)

(5) Calculate the change in user position and the receiver clock offset matrix by minimising the error. This is done using the following matrix computation.

$$\begin{bmatrix} \Delta X_u \\ \Delta Y_u \\ \Delta Z_u \\ \Delta C \end{bmatrix} = \left[G^{'}G\right]^{-1} G^{'} R_i^1 \tag{8.6}$$

(6) Calculate the new user position and receiver clock offset

$$\begin{bmatrix} X_u^2 \\ Y_u^2 \\ Z_u^2 \\ C^2 \end{bmatrix} = \begin{bmatrix} X_u^1 \\ Y_u^1 \\ Z_u^1 \\ C^1 \end{bmatrix} + \begin{bmatrix} \Delta X_u \\ \Delta Y_u \\ \Delta Z_u \\ \Delta C \end{bmatrix} \tag{8.7}$$

For the next iteration, the new user position and the receiver clock offset are X_u^2, Y_u^2, Z_u^2 and C^2

(7) Repeat the above procedure (1 to 6) and use X_u^2, Y_u^2, Z_u^2 and C^2 instead of X_u^1, Y_u^1, Z_u^1 and C^1

(8) Stop the procedure (iterations) when the change in user position and the receiver clock offset matrix is very small or negligible.

After few iterations (here approximately five iterations) the change in user position and the receiver clock offset matrix is very small. After the final iteration, say the final user position and the receiver clock offset are X_u, Y_u, Z_u and C.

Step 3: Range residual calculation

Range residuals to satellite *i* is given as

$$R_i = \rho_m^i - \rho_e^i \tag{8.8}$$

ρ_m^i is calculated using equation 3 and

$$\rho_e^i = \sqrt{(X_s^i - X_u)^2 + (Y_s^i - Y_u)^2 + (Z_s^i - Z_u)^2} + C \tag{8.9}$$

Where

R_i range residual to satellite *i*

X_s^i, Y_s^i and Z_s^i are the satellite (*i*) position

X_u, Y_u, Z_u and C are the user position and receiver clock offset calculated in step 2.

8.3 Integrity Method

A flowchart of the proposed integrity method is shown in Figure 8.2. The procedure starts with the calculation of Horizontal Protection Level (HPL) from GPS raw positioning data (see equations 9.10 to 9.14 below). HPL is the upper boundary of the confidence region of a GPS receiver, in which positioning error outside that region can be detected. The availability of Receiver Autonomous Integrity Monitoring (RAIM) is examined by comparing the calculated HPL with the Horizontal Alert Limit (HAL). HAL is the error tolerance not to be exceeded in horizontal direction (i.e., X and Y) without issuing an alert to users. If HAL is greater than HPL then it is assumed that the RAIM is available. If the RAIM is

not available an appropriate warning should be given to the user. On the other hand, if RAIM is available, the next step is to detect any fault in the positioning output. A weighted least squares residual method is used to identify the fault. If any fault is detected, rather than immediately giving a warning to the user, the integrity method identifies the operational environment in which the vehicle is travelling. Although the raw positioning fixes from the GPS contains some errors, a good map-matching algorithm can identify the correct road segment if the road network on which the vehicle is travelling is simple. If the operational environment is complex, an alert is raised. The procedure for the identification of the complexity of a road network is explained in the following section. If the operational environment is not complex (i.e., rural), then the integrity process is followed by the map-matching process and its integrity measurement. In this case, as shown in Figure 8.2, a fuzzy inference system (FIS-1) is used to derive an integrity scale. The integrity scale value determines whether an alert is issued or not.

If no fault is detected in the raw positioning fix, then the map-matching process and its integrity measurement are carried out. This is because, although there is no error with the GPS fixes, the map-matching process may fail to identify the correct link (from the set of candidate links) due to the complexity of road network, errors in GIS digital map, or errors in the map-matching process. Therefore another fuzzy inference system (FIS-2) is used so as to derive the integrity scale for the purpose of raising an alert or not.

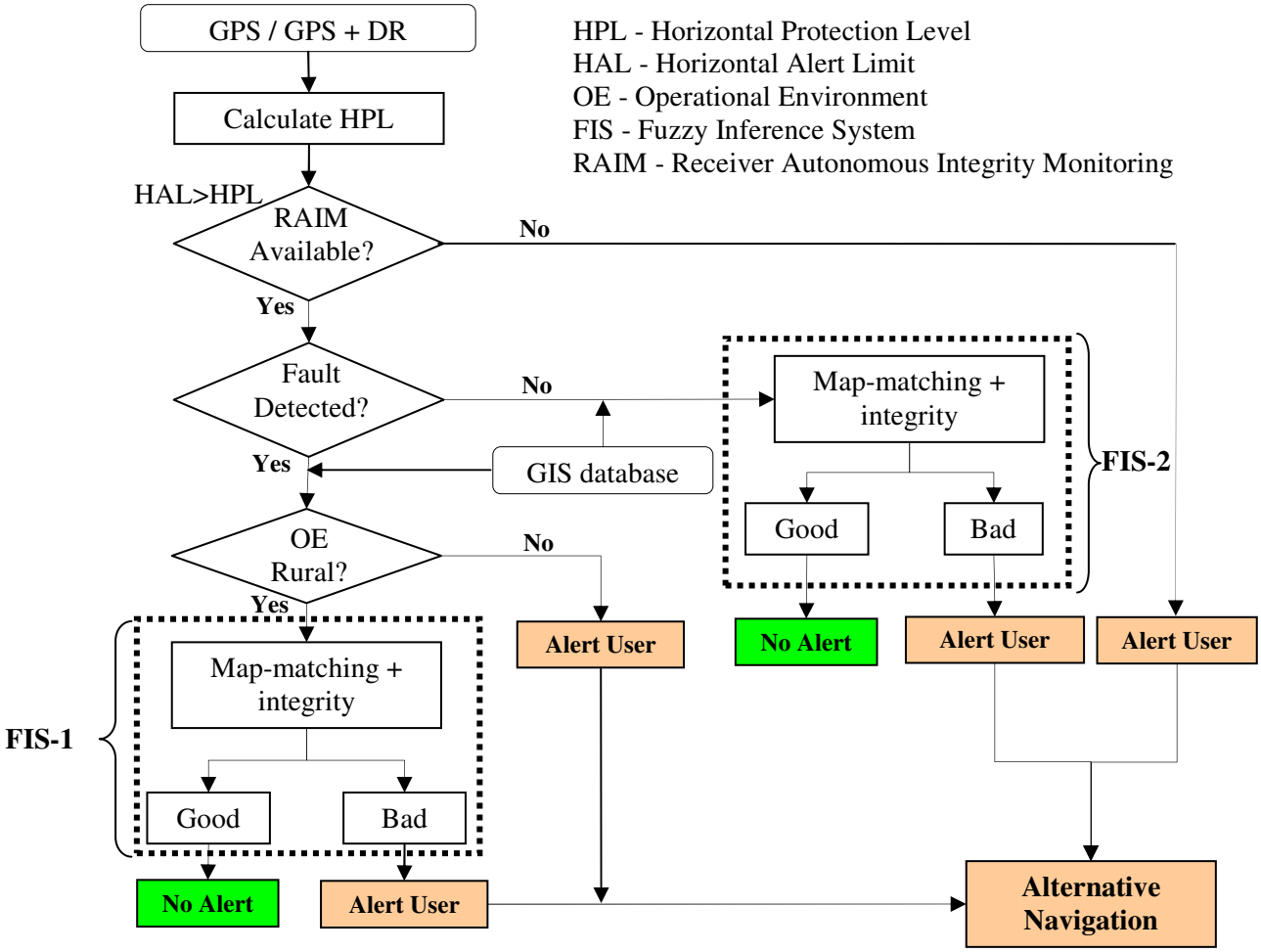


Figure 8.2 A flowchart representing the integrity method

8.3.1 Integrity of Raw GPS Positioning Fixes

RAIM is a powerful technique in providing final layer of integrity of the space (GPS) data processing (Walter and Enge, 1995). In this research, a measurement domain RAIM method is used to check the consistency of the raw positioning fixes. The process of integrity monitoring of raw positioning fixes can be divided into two stages: (1) computation of protection level to check the RAIM availability; and (2) detection of failures (Feng and Ochieng, 2007).

Here, a HPL method is used to check the RAIM availability. The horizontal protection level (HPL) is compared with the horizontal alert limit (HAL) to

check the availability of integrity function. The calculation of HPL is as given in Walter and Enge (1995).

$$HPL = \max[Hslope]T(N, P_{FA}) + k(P_{MD})HDOP \tag{8.10}$$

$$\max[Hslope] = Maximum(Slope(i)) \tag{8.11}$$

$$Slope(i) = \frac{\sqrt{(A_{1i}^2 + A_{2i}^2)}}{\sqrt{S_{ii}}} \tag{8.12}$$

$$A = (GWG)^{-1} G^T W \tag{8.13}$$

$$S = I - G(GWG)^{-1} G^T W \tag{8.14}$$

Where,

G is an observation matrix,

W is the weight matrix,

I is the identity matrix,

T is a threshold value, which is function of number of satellites (N) and probability of false alarm (P_{FA}), follows a chi-square distribution with N-4 degree of freedom,

$k(P_{MD})$ is the number of standard deviations corresponding to specified probability of missed detection (P_{MD}) and

$HDOP$ is the horizontal dilution of precision.

The failure detection process identifies the potential threats in the GPS based on the vehicle position calculation. A weight based least squares method is used to detect faults. The presence of failure is identified by comparing the test statistic (TS) with a decision threshold. If the TS is less than the decision threshold there is no fault detected, and vice versa.

Weighted Sum of Squares Errors (WSSE) can be defined as (Walter and Enge, 1995):

$$WSSE = Y^T W\left[(I - P)Y\right] \tag{8.15}$$

$$P = G\left(G^T WG\right)^{-1} G^T W \tag{8.16}$$

Where,

G is the observation matrix,

Y is the range residuals,

W is the weight matrix,

I is the identity matrix.

The test statistic is defined as the square root of WSSE (Walter and Enge, 1995). The corresponding threshold is determined based on the probability of false alarm and the number of satellites (Walter and Enge, 1995).

If the threshold exceeds the test statistic then the GPS positioning fix is assumed to be safe and usable and vice versa. Even if the raw GPS positioning fixes contain error, if the operational environment is not dense (i.e., rural) a good map-matching process has the potential to identify the correct road segment. Here, the system needs to check the operational environment in which the user is travelling. The following section describes the method of identifying an operational environment.

8.3.2 Identification of an Operational Environment

Though the raw GPS positioning fix has an error, if the vehicle is travelling in a rural environment, a map-matching process should be able to identify the correct road segment on which a vehicle is travelling. In the proposed integrity method, if the raw GPS fix is not trustable then the operational environment is identified; if the operational environment is rural then the process continues with map-matching rather giving an alert to the user.

Identification of operational environment is explained in Chapter 7. It is based on a threshold which is a function of the total length of the road network and the number of junctions per unit area (see section 7.4.2 for further details):

$$T_{OE} = f(L, N) \tag{8.17}$$

Where,

T_{OE} is the threshold for identifying an operational environment,
L is the total length of road network (in km) per given area,
N is the number of junctions in that area.

Here, the objective is to know whether a vehicle is in a simple road network (i.e., rural operational environment) is not. Assuming that *N/L* for different operational environments follow a normal distribution, the value obtained by adding mean and two times standard deviation ($\mu_R + 2\sigma_R$) for the rural environment (i.e., 4.19) is considered as the threshold value to decide whether the vehicle is in a rural area. It is also examined that the threshold value is less than the value obtained by subtracting the standard deviation of the suburban area from its mean (i.e., 4.29) and also the value obtained by subtracting the factor mean in urban area and three times the corresponding standard deviation (i.e., 5.82). Therefore, if the factor (i.e., *N/L* calculated from the same radius of 200m) is less than the threshold T_{OE} (which is 4.19) then it is determined that the user is in a rural area. Otherwise, the user is thought to be travelling in a suburban or an urban area.

8.3.3 Map-matching Integrity

If the quality of raw positioning point is good or a vehicle is in a simple road network; and if the quality of raw positioning point is bad, the process continues with MM the raw positioning point. Then the map-matching integrity process is carried out by considering the distance residual and the heading residual. These

two residuals check the ability of the algorithm to correctly identify the road segment and determine the vehicle location on the selected segment. Here, the uncertainty associated with the digital road map is also taken into account. The heading residual and distance residual are explained in this section.

8.3.3.1 Heading Residual

The correct road segment identified by the map-matching algorithm needs to be in-line with the vehicle's direction of movement. If the vehicle movement direction (heading) with respect to the North is θ, and the selected road link direction with respect to the North is β, then the absolute difference between these two directions (i.e., $|\theta - \beta|$) should be close to zero. However, the heading from GPS is not accurate and there is also error in the estimation of link heading due to errors in the digital map. These two errors need to be considered while calculating the heading residual. The heading residual is derived as suggested by Quddus et al. (2006c).

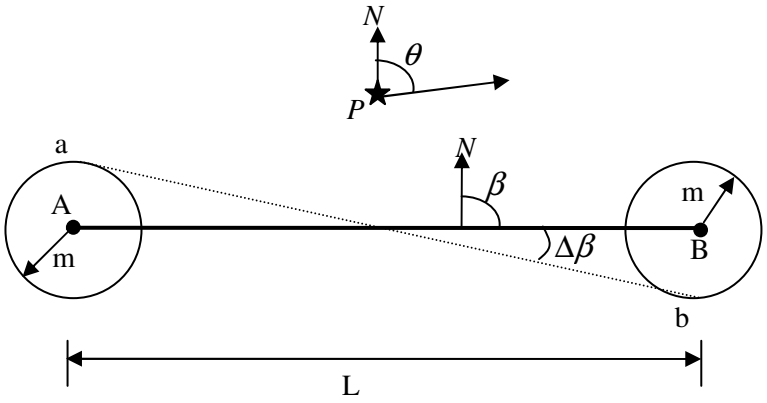


Figure 8.3 Heading residual (modified from Quddus et al., 2006c).

In Figure 8.3, P is the raw positioning point with heading angle θ from the North. The corresponding correct link for that positioning point is link 'AB' and β is the link direction with respect to the North. The scale of a GIS road map represents the minimum plottable error (Ochieng and Sauer, 2002). For example,

a digital map of scale 1:2500 has an error of 2.5m. Based on this plottable error, an error bubble is defined at the start and the end node. In Figure 8.3, for node 'A' and 'B' the error bubble of radius *m* (which depends on the map scale) is considered. In the worst case, the maximum error in link heading can be as large as $\Delta\beta$ with the position of node 'A' at 'a' and the node 'B' at 'b' in the error circle. The uncertainty in vehicle heading calculation is detected from the variance for heading.

Using an error propagation theorem the combined error can be given as (Quddus et al., 2006c):

$$HE = \sqrt{(3\sigma_h)^2 + (\Delta\beta)^2} \tag{8.18}$$

$$\Delta\beta = \tan^{-1}\left(\frac{m}{L/2}\right) \tag{8.19}$$

Where,

m is the digital map plottable error derived from the map scale,

L is the length of the road segment, and

$3\sigma_h$ is the error variance (99.8% confidence level) associated with the vehicle

heading.

If the difference between ($|\theta - \beta|$) and *HE* is less than or close to zero, then it can be said that the correct road segment identification by the map matching algorithm is reliable.

The heading residual can be represented as:

$$\Delta H = \left|\theta - \beta\right| - HE \tag{8.20}$$

8.3.3.2 Distance Residual

The distance between two consecutive map-matched positioning fixes should be comparable with the distance travelled by the user within the interval (i.e., the distance obtained by multiplying the vehicle speed and time). However, generally, there will be an error associated with the map-matched positioning points as well as the vehicle speed measurement. This is explained in Figure 8.4. P_1 and P_2 are the two consecutive positioning points from GPS and the corresponding map-matched positioning points on the identified map-matched segments are mp_1 and mp_2 respectively. Generally, there will be an error in the map-matched positions due the error in the digital map and the error in the map-matching process. Here, the map-matched positioning errors for positioning fixes P_1 and P_2 are Δd_1 and Δd_2 respectively.

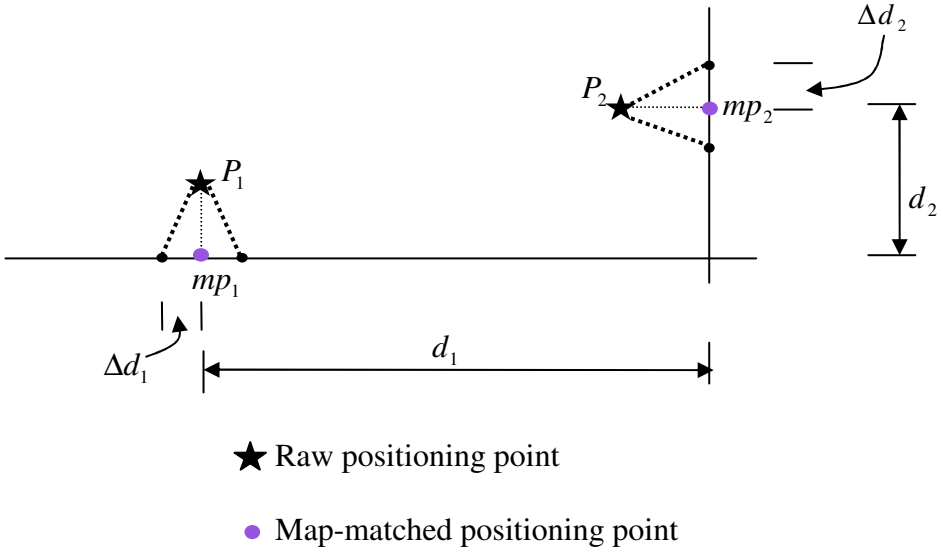


Figure 8.4: Distance residual

The uncertainty with the vehicle speed calculation is $3\sigma_v$ (i.e., error variance associated with the measurement of vehicle speed, with 99.8% confidence level). Based on the error propagation theorem the combined error (Δd_1, Δd_2 and $3\sigma_v$) is:

$$DE = \sqrt{(\Delta d_1)^2 + (\Delta d_2)^2 + (3\sigma_v)^2} \quad (8.21)$$

Here, the positioning error terms (Δd_1 and Δd_2) are identified from a series of map-matched positioning and their corresponding true positions. For the purpose of deriving these two errors a real-world dataset of 2,040 positioning fixes, (data set 5) from suburban areas of London, is used. The true positioning fixes are obtained from a carrier-phase GPS receiver. From a post-processing analysis of the above positioning data, the Δd_1 and Δd_2 are identified as 3.6m and 4.1m respectively.

The distance residual is given as follows:

$$\Delta D = \left| D_1 - D_2 \right| - DE \quad (8.22)$$

where,

D_1 is the travelled distance between the two consecutive map matched positioning points (in figure 8.4 sum of d_1 and d_2)

D_2 is the distance calculated by multiplying vehicle speed (m/sec) and the time (sec).

If the difference between the absolute value of (D_1 - D_2) and the DE is less than or close to zero then it can be said that the identification of the road segment and vehicle location on that road, by the map-matching algorithm, is more reliable, and vice versa. The following section provides a description of the derivation of the integrity scale, which is used to decide whether to alert users.

8.4 Derivation of Integrity scale using an artificial intelligence technique

Artificial intelligence (AI) can be defined as the ability of a computer software and hardware to do things that would require intelligence behaviour as if it is

done by a human (Callan, 2003). In other words, AI is a system that thinks and acts rationally like humans (Russel and Norvig, 2002). AI is used in many fields. The application areas can be broadly classified into: problem solving by searching and learning, planning and acting in real world, learning from observations, recognizing patterns and making logical inferences (Russel and Norvig, 2002; Callan, 2003).

Generally, the following artificial intelligence (AI) techniques are used in the field of transportation engineering (Bielli, 1991; Teodorovi, 1999; Bell, 2000; Yin et al., 2002; Zhong, et al., 2004; Konar, 2005; Hawas, 2007):

Genetic Algorithms (GA)
Neural Networks (NN)
Fuzzy Logic (FL)
Game theory
Belief theory

And combinations of the above techniques: such as neuro-fuzzy synergism, neuro-GA synergism, fuzzy-GA synergism, neuro-belief theory synergism and neuro-fuzzy-GA synergism.

Genetic algorithm is a technique used to find exact or approximate solutions by searching methods (Konar, 2005). This is used for problem solving by searching and learning application. The neural network (NN) technique is based on cellular structure (also called neurons) of the human brain. A NN has the ability to approximate arbitrary nonlinear functions in a decision making process (Karray and De Silva, 2004). The NN technique is further categorised into artificial neural network and biological neural network. These techniques are generally used for recognizing patterns, planning and acting in real world, and making logical inferences (Karray and De Silva, 2004).

Fuzzy logic (FL) is an effective way to make conclusions based on qualitative terms and linguistic vagueness. Fuzzy logic systems are usually built with a set of rules (using 'if' and 'then') to solve problem, rather attempting to model a system mathematically (Konar, 2005). The FL system is generally used for making logical inferences (Konar, 2005). Game theory attempts to mathematically capture behavior in strategic situations to make decisions based on number of available conditions (Levinson, 2005). It is useful in planning and acting in real world and in decisions making process (Levinson, 2005). Belief theory makes it possible to model the uncertainty of a problem (Konar, 2005). It is based on probability theory (particularly, conditional and Bayesian probability). Belief theory is generally used for learning from observations, recognizing patterns. In some situations, the combination of AI techniques are considered to solve a problem (Konar, 2005). For example, in integration of GA with FL (i.e., fuzzy-GA synergism) optimises the parameters of a fuzzy system by using a genetic algorithm technique. Further details of these integrated systems, and how combined systems work can be found in Konar (2005).

In the proposed integrity method, the user alert should be given based on the quality of raw politicising fix, uncertainty in digital map and errors associated with the map matching process. In other terms, *IF* the final positioning point has error *THEN* an alert should be given to the user. The process involves the logical functions 'if' and 'then'. Moreover, the system need to find a single output (integrity scale) from multiple inputs (different errors). In this kind of scenario, from the brief review of above AI technologies, an artificial neural network system or fuzzy logic system are suitable to solve the problem.

Though the artificial neural network (ANN) system is useful to take decisions from multiple inputs, in ANN the logical functions and linguistic terms (such as *IF* and *THEN*) are not possible to model (Konar, 2005). Moreover, the ANN is more complex than a fuzzy logic system (Matreata, 2005). So, the derivation of

integrity scale is carried out using a fuzzy inference system. This is further explained in the following section.

8.4.1 Derivation of an Integrity Scale Using Two Knowledge-based FISs

In this integrity method, an alert is raised based on the quality of raw positioning fix, the uncertainty associated with the digital map and the error related to the map matching process. In other words, the integrity monitoring process needs to deal with a series of logical statements in order to address with such uncertainties. As shown in Figure 8.2, two fuzzy inference systems (FIS) are used: FIS-1 is used if a fault is detected in the GPS positioning output and FIS-2 if no fault is detected. For both cases, an integrity scale, which represents the confidence level of the final map-matched positioning fix, is defined (Quddus et al., 2006c). The integrity scale ranges from '0' to '100'. The value 0 represents the most un-trustable user position and 100 represents the most trustable positioning fix.

As shown in Figure 8.2, FIS-1 is designed for the case of a faulty GPS positioning fix. If the raw GPS fix is not trustworthy, and the operational environment is rural (i.e., a less-dense road network), then the algorithm continues with the map-matching process. Finally, the estimated error associated with the raw positioning fix and the errors associated with the digital map and the map matching process are considered in the derivation of the integrity scale. The FIS-1 is a Sugeno-type fuzzy inference system with three stated input variables. These are the difference in test statistic and decision threshold (ΔT), the distance residual (ΔD) and the heading residual (ΔH). Twelve knowledge based fuzzy rules and one output (integrity scale) are considered in this FIS. The fuzzy sub-sets associated with the ΔT are *low*, *average* and *high;* ΔD and ΔH are *positive* and *negative*. The fuzzy rules are formulated based on knowledge-based interpretation of the characteristics of the variables. For example, if the ΔT values is very low (i.e., close to zero) and ΔD and ΔH are negative, then the

integrity scale should be *high*. A zero-order Sugeno output model, that has five constant output levels (Z_i), is developed. These output levels are selected as *very low* (Z_1=0), *low* (Z_2=40), *average* (Z_3=65), *high* (Z_4=85) and *very high* (Z_5=100).

FIS-2 is used when there is no GPS positioning fault. FIS-2 uses two inputs: ΔD and ΔH in which four knowledge based fuzzy rules are used. The shapes and the parameters of membership functions (MF) can be derived in an optimal way using advanced techniques such as a genetic algorithm or Matlab inbuilt toolbox Adaptive-Neuro Fuzzy Inference System (ANFIS) (Wong and Hamouda, 2000). However, these automatic MF tuning methods need true input and output (i.e., the true integrity output for the corresponding ΔT, ΔD and ΔH) so as to adjust the membership functions shape and parameters (Chen and Tsai, 2008). But, in our case, although the true input data can be obtained from carrier-phase GPS points, the corresponding integrity scale is not known. Here, an empirical analysis is used to determine the membership functions shape and parameters. This is accomplished using an independent dataset of 221 carrier-phase GPS positioning points collected in Nottingham, UK (data set – 4). This dataset is independent from the positioning dataset used for performance evaluation. The construction of the membership functions is done by adjusting the internal parameter values and the shape, and by checking the performance (i.e., whether the integrity method can provide low values of the integrity scale when there are failures in the inputs). The membership functions for the two FIS systems are shown in Figure 8.6. Fuzzy rules corresponding to both the FIS systems are illustrated in Table 8.1. Equal rule weights (firing strength) are considered for both the FISs.

Table 8.1: Fuzzy Rules Used in Fuzzy Inference Systems

Fuzzy Inference Systems-1 (FIS-1)
R_1: if ΔH is *negative* and ΔD is *negative* and ΔT is *low* then integrity scale (Z) is *very high*
R_2: if ΔH is *negative* and ΔD is *negative* and ΔT is *average* then integrity scale (Z) is *high*
R3: if ΔH is *negative* and ΔD is *negative* and ΔT is *high* then integrity scale (Z) is *average*
R4: if ΔH is *negative* and ΔD is *positive* and ΔT is *low* then integrity scale (Z) is *high*
R5: if ΔH is *negative* and ΔD is *positive* and ΔT is *average* then integrity scale (Z) is *average*
R6: if ΔH is *negative* and ΔD is *positive* and ΔT is *high* then integrity scale (Z) is *average*
R7: if ΔH is *positive* and ΔD is *negative* and ΔT is *low* then integrity scale (Z) is *high*
R8: if ΔH is *positive* and ΔD is *negative* and ΔT is *average* then integrity scale (Z) is *high*
R9: if ΔH is *positive* and ΔD is *negative* and ΔT is *high* then integrity scale (Z) is *average*
R_{10}: if ΔH is *positive* and ΔD is *positive* and ΔT is *low* then integrity scale (Z) is *average*
$R1_1$: if ΔH is *positive* and ΔD is *positive* and ΔT is *average* then integrity scale (Z) is *low*
R_{12}: if ΔH is *positive* and ΔD is *positive* and ΔT is *high* then integrity scale (Z) is *very low*
Fuzzy Inference Systems-2 (FIS-2)
R_1: if ΔH is *negative* and ΔD is *negative* then integrity scale (Z) is *very high*
R_2: if ΔH is *positive* and ΔD is *negative* then integrity scale (Z) is *high*
R_3: if ΔH is *negative* and ΔD is *positive* then integrity scale (Z) is *average*
R_4: if ΔH is *positive* and ΔD is *positive* then integrity scale (Z) is *low*

The coding of the integrity method was completed in Matlab programming environment; and also the Matlab fuzzy tool box is used for the above two FISs (Konar, 2005).

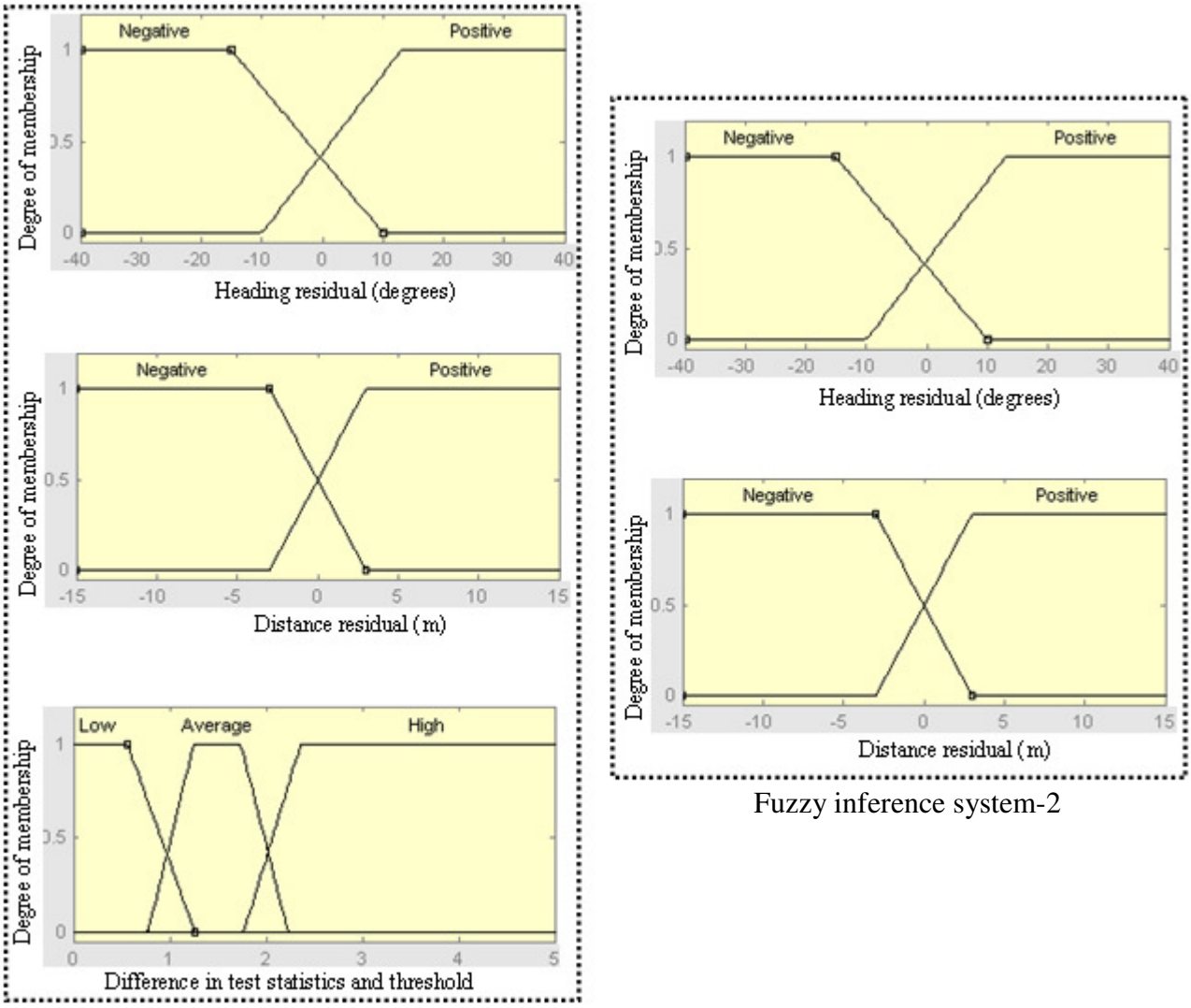


Figure 8.5 Fuzzy inference systems

As mentioned before the objective of the integrity method is to give alerts to users when the system is not usable. The criteria that is normally employed to evaluate the performance of an integrity method is Overall Correct Detection Rate (OCDR) (Jabbour et al., 2008; Quddus et al., 2006c). The OCDR refers to percentage of time the system can provide valid warnings to users. Further, invalid warnings can be either *missed detections* or *false alarms*. The missed detection (MD) suggests that although there is a mistake in the final positioning output of the navigation system, the integrity method could not identify it; and the false alarm (FA) indicates that although there is no error in the final

positioning output obtained from the vehicle navigation module the system gives an alert to users. So the performance of the developed integrity method could be measured with respect to Missed Detection Rate (MDR), False Alarm Rate (FAR) and Overall Correct Detection Rate (OCDR) (Jabbour et al., 2008; Quddus et al., 2006c). As suggested by Quddus et al. (2006c) and Jabbour et al. (2008) the overall correct detection rate (OCDR), which is derived with respect to false alarm rate (FAR) and missed detection rate (MDR), can be written as:

$$OCDR = 1 - (FAR + MDR) \tag{8.23}$$

$$FAR = \frac{f}{o} \tag{8.24}$$

$$MDR = \frac{m}{o} \tag{8.25}$$

Where,

f is the total false alarms,

m is the total missed detections and

o is the total observations.

To decide the total number of false alarms (f) and the number of missed detections (m), reference (true) vehicle positioning is required. Here, the positioning fixes obtained from GPS carrier-phase observations integrated with a high-grade Inertial Navigation System (INS) were used as reference positioning points.

The integrity scale value, which is derived in section 8.4.1, for each positioning point varies from 0 to 100. It is necessary to identify a threshold value of the integrity scale for the purpose of raising an alert or not. That means if the integrity value of a positioning point is less than the identified threshold, an alert should be given to users and vice versa. The integrity threshold is identified using an empirical analysis of an independent dataset consisting of 2,261

observations, collected in Nottingham, UK (part of data set 4). The result is shown in Figure 8.6. As can be seen, FA increases as the thresholds increases and MD decreases as the threshold increases. In order to achieve the optimal solution in terms of overall correct detection, the integrity value at which both FA and MD lines intersect is taken as the threshold.

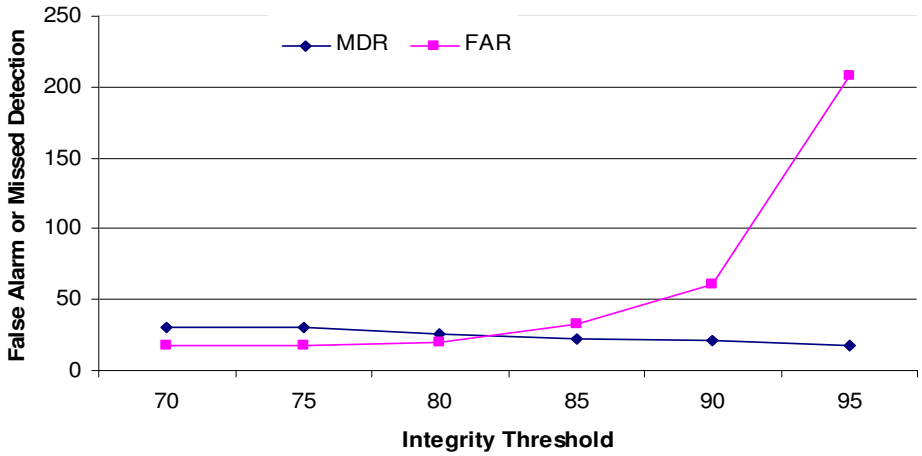


Figure 8.6: Variation in false alarms and missed detections

It can be seen that at an integrity scale of 82, the FA and MD lines intersect. So the integrity scale 82 is chosen as the threshold value to raise an alert to users. That means, if the integrity scale is less than or equal to 82 the users will be alerted and vice versa.

8.5 Performance

To evaluate the performance of the integrity method, positioning data (data set 4, in Table 5.1) collected in central Nottingham, UK, was used. The positioning fixes obtained from GPS carrier-phase observations integrated with a high-grade Inertial Navigation System (INS) were used as reference positioning points. For the collection of this dataset, a test vehicle equipped with a single frequency high sensitivity GPS receiver, a low-cost gyroscope and integrated carrier-phase GPS/high-grade INS was used. Accuracy of the integrated carrier-phase GPS and

INS was found to be better than 5 centimetres over 97.5% of the time in all three coordinate components (i.e., X, Y and Z directions) (Aponte et al., 2009). The positioning data was recorded every second. A total of 2,838 positioning fixes, along the test route of 20.6 km was used for integrity method performance evaluation. The test trajectory is shown in Figure 8.7. A spatial digital map of scale 1:2,500, obtained from Quddus (2006), was used in the analysis.

The analysis of the field data reveals that the total number of false alarms and missed detections were 24 and 28 respectively. The number of missed detection and false alarms in case of faulty raw positioning points, FIS 1, and fault free raw positioning points, FIS 2, are illustrated in Table 9.2. For five positioning points the RAIM is not available (HPL>HAL). The FAR, MDR and OCDR were found to be 0.0084, 0.0099 and 0.9817 respectively. That means, the integrity method gave correct warnings 98.2% of the time. The user needs to be alerted within the time-to-alarm limit. Therefore, the computational speed of the integrity method is also important. In terms of computational speed, the integrity method processed 4 positioning fixes per second (with a laptop of 2 GB RAM and 1.46 (1.83 GHz) processor speed).

Table 8.2: Integrity performance

	False Alarms	Missed detections
Faulty GPS points (FIS-1)	12	1
Fault free GPS points (FIS-2)	12	27

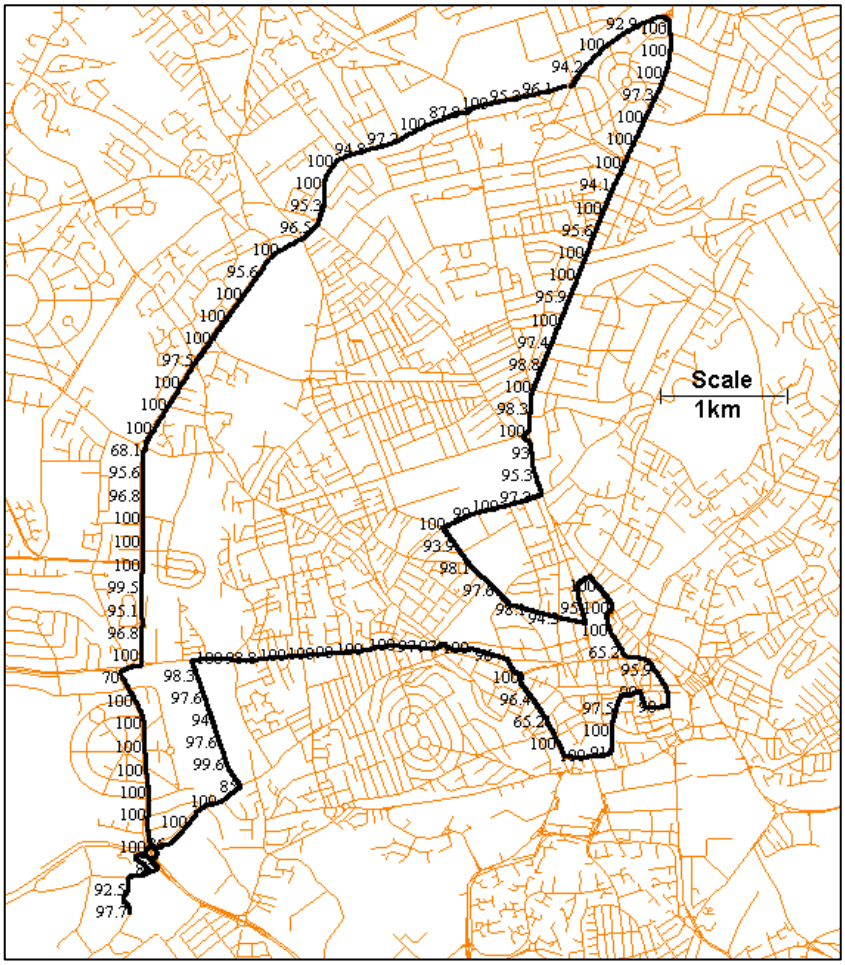


Figure 8.7: Test route in Nottingham, UK

8.6 Summary

An integrity method, that gives warnings to users when the final positioning output is not usable, was developed in this chapter. Here, the error sources associated with raw positioning points, digital map and map-matching process were considered simultaneously. Moreover, the operational environment in which a vehicle is travelling is also taken into account in the integrity process. The sequential process of checking outliers in the raw positioning fixes using a measurement domain RAIM method and followed by checking for an operational environment and then examining the integrity of map-matching process improved the performance. The integrity scale was derived using two

knowledge based fuzzy inference systems. The next chapter provides the discussion on research findings.

Chapter 9

Discussion

9.1 Introduction

This chapter firstly summarises the key features of the developed topological map-matching algorithm and the integrity method; secondly compares the performance of both the enhanced algorithm and the integrity method with the evidence available in the literature; then examines the extent to which the algorithm and integrity method can support navigation requirements of location-based ITS services; and finally describes the suitability of both the MM algorithm and the integrity method for practical implications by industry.

9.2 Key features of the MM algorithm and the Integrity method

This research has developed a weight-based topological MM algorithm and a map-aided integrity monitoring process for supporting the navigation module of location-based Intelligent Transport Systems (ITS) applications. Firstly, an in-depth literature review was carried out to identify constraints, limitations and the performances of existing MM algorithms and integrity methods. Different ways to improve tMM algorithm and integrity method were identified. Then, a weight-based topological MM algorithm and a map-aided integrity method were developed and tested.

The key features of the developed topological MM algorithm are:

(a) A robust method for selecting candidate links in the initial map-matching process and the map-matching at junctions;

(b) Introduction of two additional weight parameters: connectivity and turn restriction;

(c) Use of an optimisation process to derive the relative importance of weights using positioning data collected in three different operational environments;

(d) Identification of an operational environment in which a vehicle is travelling; and

(e) Implementation of two consistency checks to reduce mismatches.

The key features of the developed integrity method include:

(a) Considering errors associated with the positioning data, GIS map and map-matching process concurrently in identifying the goodness (trustability) of the final positioning point;

(b) Considering the operational environment; and

(c) Developing two fuzzy inference systems (FIS) to measure the integrity scale.

9.3 Performance of the enhanced MM algorithm and the integrity method compared with existing algorithms

These new features have improved the performance of the tMM algorithm and the integrity method. Both the tMM algorithm and the integrity method were tested using real-world field datasets. The enhanced algorithm succeeded 97.8% of the time in correct link identification with an horizontal accuracy of 9.1 m $(\mu + 2\sigma)$ and the integrity method was capable of providing correct warnings 98.2% of the time.

Chapter 3 reviewed existing topological map-matching algorithms and presented evidence on the performance of tMM algorithms from the literature. The summary of existing tMM algorithms performance along with the performance of the enhanced tMM algorithm developed in the research is illustrated in Table 9.1.

Table 9.1: Performance of topological MM algorithms

Author and year of publication	Correct link identification	Horizontal accuracy in metres (95% confidence level)
White et al. (2000)	85.80%	32
Greenfeld (2002)	85.60%	18.3
Srinivasan et al. (2003)	98.50%	21.2
Quddus et al. (2003)	88.60%	18.1
Yin and Wolfson (2004)	94%	--
Blazquez and Vonderohe (2005)	94.80%	--
tMM algorithm developed in this research	**97.8%**	**9.1**

The MM algorithm developed by Srinivasan et al. (2003) identified 98.5% of the segments correctly; this was based on a small positioning data of sample size 242 positioning points (about four minutes data) in a simple network (university roads). When tested on a larger, more representative, road network, the accuracy falls to 80.2% (see Quddus, 2006). The tMM algorithm developed in this study outperforms the existing topological MM algorithms (with respect to percentage of correct link identification and horizontal accuracy) reported in the literature except the algorithm by Srinivasan et al. (2003), which was evaluated using a very small positioning data in university roads.

The performance of existing integrity methods and the map-aided integrity method developed in this research are summarised in Table 9.2.

Table 9.2: Performance of Integrity methods

Author and Year of publication	Performance
Sun and Cannon (1998)	79.9% valid warnings
Quddus et al. (2006c)	91.1% valid warnings[1]
	97.5% valid warnings[2]
	98.2% valid warnings[3]
Yu et al. (2006)	MDR is 1.41%
Kuusniemi et al., (2007)	89.5% valid warnings
Jabbour et al. (2008)	88.8 % valid integrity warnings
Integrity method developed in this research	**98.2% valid integrity warnings**

[1]Using a weight based topological MM algorithm.
[2]Using a probabilistic MM algorithm.
[3]Using an advanced MM algorithm.
MDR: missed detection rate

The integrity method developed by Yu et al. (2006) provided the performance only with respect to the missed detection rate (MDR); therefore, the results are not directly comparable. The performance of the integrity method developed in this research is better than most existing integrity methods and equals that of an integrity method, supported by an advanced MM algorithm, developed by Quddus et al. (2006c). However, the advanced algorithms uses more refined approaches and may not be easy to implement for real-time ITS applications.

9.4 Location-based ITS services that may be supported by the algorithms developed in this thesis

Positioning requirements of ITS services are represented with Required Navigation Performance (RNP) parameters. The RNP parameters for land vehicle navigation are still under development. The evidence of RNP values in the literature, for various ITS user groups and services, are illustrated in Table 2.3. After the critical review of RNP for location-based ITS services, it is noticed that the identification of RNP parameters for all ITS services or applications are yet to fully develop. Ideally one needs to carry out a series of field tests to

determine the RNP parameters (that depend on various driving factors: target level of safety, economic and operational efficiency) of an intelligent transport system. This is however not within the scope of this research.

ITS services that require horizontal accuracy of 10 metres (95% of the times) can be supported by the developed algorithms. These include the following (identified from Table 2.3):

(a) Liability critical applications: electronic toll collection, electronic parking payment and GPS based variable road user charging;
(b) System performance critical applications: navigation and route guidance, public transport management, automatic bus arrival announcements;
(c) Commercial applications: fleet management, commercial vehicle administrative processes and electronic clearance; and
(d) Safety critical applications: emergency vehicle management, incident and accident management.

The developed tMM algorithm and the integrity method may support most ITS services the exceptions being safety-critical ITS services such as lateral and longitudinal collision avoidance systems and vehicle based collision notification system. The 'safety-of-life' (SOL) critical applications like vehicle collision avoidance systems require 0.1 m to 1 m horizontal accuracy (FRP, 1999; DOT, 2004; Quddus, 2006; and Feng and Ochieng, 2007). To achieve this high accuracy a stand-alone GPS, digital map and MM algorithm are not sufficient. The vehicle navigation module needs to be integrated with emerging positioning technologies such as laser scanners, radars, network-based real-time kinematic (N-RTK) carrier phase observables and video cameras. This is however not with in the scope of this thesis.

9.5 Suitability of the developed tMM algorithm and integrity method for practical implications

This research has developed a simple, fast and generic map-matching algorithm and an integrity method to support real-time ITS applications. To provide continuous vehicle location information, here the positioning data were recorded every second (i.e. frequency 1Hz). The positioning data were collected from three different countries (UK, India and USA) using a low-cost GPS receiver. The validation of the developed algorithm and the integrity method were carried out using a higher accuracy reference (truth) of the vehicle trajectory obtained from a carrier phase GPS observables integrated with high-grade INS. This provides evidence of the reliability and transferability of the enhanced algorithm and the integrity method. The developed tMM algorithm and integrity method have high potential to be implemented by industry for real- time applications as these both algorithms are easy to implement and have a high degree of transferability.

A significant improvement in the topological MM algorithm and the integrity method was achieved by introducing more sophisticated and logical techniques such as weight scores optimisation technique in the tMM algorithm and fuzzy inference system in the integrity method. The benefit of this research is mainly enhancement of real-time vehicle positioning systems to support a range of location-based ITS services.

Chapter 10

Conclusions and Recommendations

10.1 Map-matching algorithm and the integrity method

In this research, a weight-based topological map-matching algorithm was developed after identifying constraints of existing map-matching algorithms through a detailed literature review. Then, errors in the developed tMM algorithm were determined using an extensive positioning data collected in three different countries (UK, USA and India). Further enhancement of the tMM algorithm was carried out. Considering two new weights, optimisation technique to identify the relative importance of weight scores, considering operational environment were the main novelties of the developed tMM algorithm.

A user-level integrity method, which takes into account all error sources associated with a navigation system simultaneously, was developed. The sequential process of checking outliers in the raw positioning fixes using a measurement domain Receiver Autonomous Integrity Monitoring method, followed by checking for an operational environment then examining the integrity of map-matching process and finally deriving the integrity scale using two fuzzy inference systems improved the performance.

The developed tMM algorithm identified links correctly 97.8% of the time; the horizontal accuracy of the algorithm was measured as 9.1 m $(\mu + 2\sigma)$. The integrity method provided valid warnings 98.2% of the time. It was revealed that

the enhanced algorithm and the integrity method can support a range of ITS services that require positioning accuracy of 10 m. The main research contribution in the area of vehicle positioning systems for location-based ITS services is described in the following section.

10.2 Research Contribution

The primary aim of the proposed research was to enhance navigation modules of location-based Intelligent Transport Systems (ITS) by developing a weight-based tMM algorithm and a map-aided integrity monitoring process. In order to achieve this aim a set of objectives were formulated (see Section 1.3). Table 10.1 gives a brief description of each of the objectives and the corresponding Chapters in which they were achieved/addressed.

Table 10.1: Research objectives

Objective	Method/Description	Chapter
Identify positioning requirements for location-based ITS services	Literature review: RNP parameters for ITS	2
Critically assess existing tMM algorithms and integrity methods	Literature review: (1) map-matching algorithms and (2) integrity methods	3 and 4
Develop a weight-based topological map-matching algorithm	(1) A well-structured tMM algorithm (2) an optimisation technique (3) the performance evaluation	6
Further improvement of the tMM algorithm	(1) Explore transferability of the algorithm (2) Identify the sources of errors (3) Identify the enhancement strategies (4) Performance re-evaluation	7
Develop an improved integrity method	(1) Consider all error sources and operational environments (2) Develop a fuzzy inference system (3) Performance evaluation	8
Performance of the tMM algorithm and integrity method	(1) Performance with respect to existing algorithms and integrity methods (2) Location-based ITS services that can be supported	9

Chapter 5, which was not listed in Table 10.1, described different positioning data sets used in this research. The main research contribution of this thesis is described in the following sections.

10.2.1 Optimisation technique to identify the relative importance of weights

The weight-based topological MM algorithm assigns weights to all candidate links based on different criteria such as the similarity in vehicle movement direction and link direction, the nearness of the positioning point to a link, and the connectivity of a candidate road link to the previously travelled road link and selects the correct link based on total weight scores (TWS). In previous research, the relative importance of these weights were considered to be equal or derived empirically. In this research an optimisation technique was introduced to identify the relative importance of the weight scores for different operational environments: urban, suburban and rural.

10.2.2 Distinguishing among different operational environments

The relative importance of weight scores varied with operational environments (urban, suburban and rural). It was important to identify the operational environment in which a vehicle is travelling. The operational environment identification was achieved based on the complexity of road network, the number of junctions, and the length of road network per given area. This process was explained in Section 7.4.2. The algorithm identified the operational environment in which the vehicle was travelling, followed by total weight score calculation based on the corresponding weight scores.

In the map-aided integrity monitoring process, though a raw GPS positioning point has an error, if the vehicle is travelling in a simple operational environment a map-matching process should be able to identify the correct road segment on which a vehicle is travelling. It is more logical that if the raw positioning point quality is not good, before giving alert to an user the integrity method checks

whether the vehicle is in a simple operational environment; if so, the process continues with MM and checking the integrity of map-matching process. This enhanced the performance of the developed integrity method.

10.2.3 Transferable algorithm

In the existing research, MM algorithm's performance with respect to the percentage of the correct link identification is measured using a positioning dataset collected from a geographical place/a city. Whilst the horizontal accuracy (2D) has not been identified due to a lack of reference (true) vehicle positioning data. To explore the transferability of the developed tMM to different contexts, a total of six positioning datasets collected from the three different countries (UK, USA and India) and three different digital road network maps were used. As part of this study, positioning data were collected using a low-cost GPS receiver and a high accurate carrier-phase GPS (CGPS) receiver integrated with high-grade Inertial Navigation System (INS). The CGPS/INS system provides the reference (true) vehicle positioning data with centimetre level accuracy. This made it possible to measure the developed algorithms' horizontal accuracy (2D) and the performance of the map-aided integrity method.

10.2.4 Computational speed

This research aimed to develop a generic map-matching algorithm and an integrity method that can be useful for any location-based ITS service. The weight-based topological MM algorithm and the map-aided integrity method developed in this research were logical, simple, fast, efficient and easy to implement. In terms of computational speed, the MM algorithm and the integrity method achieved 180 positioning points per second and 4 positioning points per second respectively (with a laptop of 1 GB RAM and 1.46 processor speed). This implies that the developed algorithm has high potential to be implemented by industry for the purpose of supporting the navigation modules of location-based intelligent transport systems.

10.3 Limitations and future research

Optimisation test for more operational environments:

The relative importance of each weight score used in the tMM algorithm was optimally derived for urban, suburban and rural operational environments. This optimisation test can be further extended to more operational environments: such as dense urban, suburban, motorways, university roads, hilly roads, rural roads and others..

Integrating with a routing algorithm:

For many ITS services, origin and destination points are known before starting a trip. Examples include: navigation and route guidance, automatic announcement of bus stops and emergency vehicle management. For a given origin and destination, a routing algorithm suggests the preferred route based on the shortest distance path or the lowest travel time path or the lowest travel cost path. If a map-matching algorithm uses the recommended route information obtained from the routing algorithm when identifying the correct road segment from candidate segments, then the algorithm performance would be improved. Therefore, the integration of the MM algorithm with a routing algorithm may improve the MM algorithm's performance.

Derivation of threshold values:

In this thesis, many threshold values, derived empirically using a small positioning data set, were used. These include the distance threshold and bearing threshold in the tMM algorithm and an integrity threshold in the integrity method. These threshold values need to be further checked for different operational environments with large data sets to identify optimal values.

Identification of vehicle location on a selected road link:

The developed MM algorithm first identified the correct road segment from a set of candidate road segments. Then the perpendicular projection of the raw positioning point on the selected link gave vehicle position on that link. Further

improvement can be done in vehicle position identification on the selected link. Here, considering the road width parameter may drastically improve the algorithm's horizontal positioning accuracy. However, most of the GIS maps represent a road segment with road central line, ignoring road width.

Development of RNP for ITS:

It is noticeable from Table 2.3 that only two Required Navigation Performance (RNP) Parameters (*accuracy* and *availability*) were fully reported in the literature. There was partial information on the system *integrity* and *continuity* parameter for some services. This suggests that the impact of *integrity* (i.e. reliability of final vehicle positing data after map-matching) and *continuity* (i.e. the consequences due to a loss of either system accuracy or integrity) on the performance of a system is under development.

Unlike aviation and marine transport, for land navigation vehicles' position is always referred on a spatial road network map. In aviation and maritime, the positioning accuracy is measured with respect to the distance between true position and the measured position. In case of land vehicle navigation, firstly it is necessary to identify a road segment (from a set of candidate road segments) on which vehicle is travelling and then the vehicle position on that road segment. Here, the RNP parameter *accuracy* is required to categorise into percentage of correct link identification and horizontal positioning accuracy.

References

Aponte, J., Meng, X., Moore, T., Hill, C., and Burbidge, M., (2009), Assessing Network RTK Wireless Delivery, *GPS World*, 20(2), pp.14-27.

Bell, M. G. H., (2000), A game theory approach to measuring the performance reliability of transport networks, Transportation Research Part B, vol- 34, pp 533-545.

Basnayake, C., Mezentsev, O., Lachapelle, G., and Cannon, M.E., (2005), An HSGPS, inertial and map-matching integrated portable vehicular navigation system for uninterrupted real-time vehicular navigation. *International Journal of Vehicle Information and Communication Systems*, 1, pp. 131-151.

Bernstein, D., and Kornhauser, A., (1998), Map matching for personal navigation assistants. *In proceedings of the 77th annual meeting of the Transportation Research Board,* 11-15 January, Washington D.C.

Bhatti, U., Ochieng, W.Y. and Feng S., (2006), Integrity of Integrated GPS/Low cost Inertial System and Failure Modes: Analysis and Results – Part 1. *GPS Solutions Journal*, 11 (3), pp.173-182.

Bhatti, U., Ochieng, W.Y. and Feng S., (2007), Integrity of Integrated GPS/Low cost Inertial System and Failure Modes: Analysis and Results – Part 2. *GPS Solutions Journal*, 11 (3), pp.183-192.

Bielli, M., (1991), Artificial Intelligence Techniques for Urban Traffic Control, Transportation Research Part A, Vol 25A, pp319-325.

Blazquez, C.A., and Vonderohe, A.P., (2005), Simple map-matching algorithm applied to intelligent winter maintenance vehicle data. *Transportation Research Record,* 1935, pp. 68–76.

Bouju, A., Stockus, A., Bertrand, F., and Boursier, P., (2002), Location-based spatial data management in navigation systems. *IEEE Symposium on Intelligent Vehicle*, pp. 172-177.

Britting K.R., (1971), Inertial navigation system analysis. *Wiley*, New York.

Broughtom, R., (2003), Evaluation of a new satellite navigation integrity monitoring algorithm. *M.Sc dissertation, Astronautics and space engineering*, Cranfield University, UK.

Brunner, F.K., and Welsch, W.M., (1993), Effect of the troposphere on GPS measurements. *GPS World*, 4(1), 42-51.

Callan, R., (2003), Artificial intelligence, Ashford colour press, Gosport.

Cassell, R., and Smith, A., (1995), Development of Required Navigation Performance (RNP) for airport surface movement Guidance and Control. *Proceeding of Digital Avionics System Conference*, 14 th DASC, 5-9 November, pp. 57-64.

Chadwick, D. (1994), Projected navigation system requirements for intelligent vehicle highway systems. *In: Proceedings of Institute of Navigation GPS-94*, 485–490.

Chen, W., YU, M., LI, Z.-L., and Chen, Y.Q., (2003), Integrated vehicle navigation system for urban applications. *In Proceedings of the 7th International*

Conference on Global Navigation Satellite Systems (GNSS), Graz, Austria, April 22-24, pp. 15-22.

Chen, W., Li, Z., Yu, M., and Chen, Y., (2005), Effects of sensor errors on the performance of map matching. *Journal of Navigation*, 58, pp. 273-282.

Chen, S., and Tsai, F, (2008), Generating fuzzy rules from training instances for fuzzy classification systems, *Expert Systems with Applications*, Vol 35, pp. 611-621.

Chowdhury, M.A., and Sadek, A., (2003), Fundamentals of Intelligent Transportation System planning. *First Edition, Artech House*, London.

Christos, N.E., and Anagnostopoulos, I.E., (2006), A license plate – recognition algorithm for Intelligent Transportation System applications. *IEEE Transactions on Intelligent Transportation Systems*, 7 (3), pp. 377-392.

Ding, L., Chi, L., Chen, J. B., and Song, C., (2007), Improved Neural Network Information Fusion in Integrated Navigation System. *In Proceedings of 1st IEEE International Conference on Mechatronics and Automation*. August 5 - 8, 2007, Harbin, China.

Dixion, C. S., (2003), GNSS Local Component Integrity Concept. *Journal of Global Positioning Systems*, vol 2 (2), pp. 126-134.

DOT (Department of Transportation), (2004), Radionavigation systems: A capabilities investment strategy. *A Report to the Secretary of Transportation*, USA. Available at: http://ostpxweb.dot.gov/policy/Data/DOT%20Radionavigation%20Task%20Force%20Report%20-%206%20Feb%202004_FINAL.pdf (Accessed August 2009)

Drene, C., and Rizos, C., (1997), Positioning Systems in Intelligent Transportation Systems. *Artech House*, London.

Farrell, J.A., and Barth, M., (1999), The Global Positioning System & Inertial Navigation. *McGraw-Hill*, New York.

Feng, S., Ochieng W.Y., Hide, C., Moore, T., and Hill, C., (2007), Carrier Phase Based Integrity Monitoring Algorithms for Integrated GPS/INS Systems. *Proceedings of the European Navigation Conference*, Geneva, Switzerland.

Feng, S., and Ochieng, W.Y., (2007), Integrity of Navigation System for Road Transport. *Proceedings of the 14th ITS World Congress*, 9-13, October, 2007, Beijing, China.

FHA (Federal Highway Administration), (2003), Intelligent Transportation Systems Benefits and Costs. A *report by Federal Highway Administration*, *US Department of Transportation,* Available at: http://www.its.dot.gov/jpodocs/repts_te/13772.html (Accessed August 2009).

FRP, (1999), Federal radionavigation plan.
Available at: http://avnwww.jccbi.gov/icasc/PDF/frp1999.pdf. (Accessed August 2009)

Fu, M., Li, Jie, Wang, M., (2004), A hybrid map matching algorithm based on fuzzy comprehensive Judgment, IEEE Proceedings on Intelligent Transportation Systems, pp. 613-617.

Godha, S., and Cannon, M.E., (2005), Integration of DGPS with a MEMS-based inertial measurement unit (IMU) for land vehicle navigation application. *In: Proceedings of ION GPS- 05, Institute of navigation*, Long Beach, pp. 333–345.

Gomes, P and Pereira, R, B., (2007), The Value of Positioning Integrity for GNSS-Based Electronic Fee Collection Systems, Data Systems. *In proceedings of Aerospace – Conference*, 29 May to 1 June, Naples, Italy.

Goodwin, C., and Lau, J., (1993), Vehicle Navigation and Map Quality. *Proc., of the IEEE-IEE Vehicle Navigation & Information Systems Conference*, Ottawa, pp. 17-20.

Greenfeld, J.S., (2002), Matching GPS observations to locations on a digital map. *Proceedings of the 81st Annual Meeting of the Transportation Research Board*, January, Washington D.C.

Gujarati, D. N., (2004), Basic econometrics, *McGraw−Hill publisher*, ISBN: 978-0072478525.

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P., (2002), Particle Filters for Positioning, Navigation and Tracking, *IEEE Transactions on Signal Processing*, Vol 50 (2), pp. 425-437.

Haibin, S., Jiansheng, T., Chaozhen, H., (2006), A Integrated Map Matching Algorithm Based on Fuzzy Theory for Vehicle Navigation System, IEEE, 6, pp. 916-919.

Hawas, Y. E., (2007), A fuzzy-based system for incident detection in urban street networks, Transportation Research Part C, Vol-15, pp 69-95.

Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J., (2001), GPS theory and practice. *SpringerWien,* Nework.

Hotchkiss, N.J., (1999), A comprehensive guide to land navigation with GPS. *Alexis Publication*, *3rd Edition*.

ICAO, (1997), All Weather operations panel, advanced surface movement guidance and control systems. *Draft 4*, January-1997.

Jabbour, M., Bonnifait, P., and Cherfaoui, V., (2008), Map-Matching Integrity using Multihypothesis road-tracking. *Journal of Intelligent Transport systems*, Vol 12 (4), PP. 189 – 201.

Jagadeesh, G.R., Srikanth, T., Zhang, X.D., (2004), A map-matching method for GPS based real-time vehicle location, *Journal of Navigation*, 57(3), pp. 429-440.

Jo, T., Haseyamai, M., Kitajima, H., (1996), A map matching method with the innovation of the Kalman filtering. *IEICE Transactions on Fundamentals of. Electronics Communication and Computer Science*, E79-A, pp 1853-1855.

Jwo, D. J., (2001), Efficient DOP Calculation for GPS with and without Altimeter Aiding, *The Journal of Navigation*, volume 54, issue 2, pp 269-281.

Jwo, D., (2005), GPS navigation solution by Analogue Neural network Least-squares process, The journal of navigation, Vol 58, PP 105 to 118.

Kaplan, E.D., and Hegarty, C.J., (2006), Understanding GPS – Principles and Applications. *second edition, Artech House*, Boston/London.

Karray, F, O., and De Silva, C., (2004), Soft Computing and Intelligent Systems Design: Theory, Tools and Applications, *Addison-Wesley*, Essex, England.

Kennedy, P., (2008), A guide to econometrics, *Wiley-Blackwell publisher*, ISBN: 978-1405182584.

Krakiwsky, E.J, (1993), The diversity among IVHS navigation systems worldwide. IEEE-IEE Vehicle Navigation and Information Systems Conference, Ottawa, 433-436.

Kealy, A., Young, S., Leahy, F., and Cross, P., (2001), Improving the performance of satellite navigation systems for land mobile applications through the integration of MEMS inertial sensors. In: *Proceedings of ION GPS-01*, Institute of navigation, Salt Lake City, pp. 1394–1402.

Kibe, S.V., (2003), Indian plan for satellite-based navigation system for civil aviation. *Current Science,* 84 (11), pp. 1405-1411.

Kim, J.S., Lee, J.H., Kang, T.H., Lee, W.Y., and Kim, Y.G., (1996), Node based map matching algorithm for car navigation system. *Proceedings of the 29th ISATA Symposium*, Florence, 10, pp. 121–126.

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

Kim, S., and Kim, J., (2001), Adaptive fuzzy-network based C-measure map matching algorithm for car navigation system. *IEEE Transactions on industrial electronics*, 48 (2), pp. 432-440.

Klein, L.A., (2001), Sensor technologies and data requirements for ITS. *Artech House*, London.

Klobuchar, J.A., (1987), Ionospheric time-delay algorithm for single-frequency GPS users*, IEEE Transactions on Aerospace and Electronic Systems*, AES-23(3), 325-331.

Klobuchar, J.A., 1991, *Ionospheric effects on GPS*, GPS World, 2(4): 48-51.

Klobuchar, J.A., (1996), Ionospheric effects on GPS, *Parkinson, Spilker (eds), Vol. 1*.

Konar, A., (2005), Computational intelligence: Principles, Techniques and Applications, Springer Berlin, New York, 2005.

Krakiwsky, E.J., Harris, C.B., and Wong, R.V.C., (1988), A Kalman Flter for integrating dead reckoning, map matching and GPS positioning. *In: Proceedings of IEEE Position Location and Navigation Symposium*, pp. 39-46.

Kuusniemi, H., Wieser, A, Lachapelle, G and Takala, J, (2007), User-level Reliability Monitoring in Urban Personal Satellite-Navigation. *IEEE Transactions on Aerospace and Electronic Systems,* Vol 43 (4), pp.1305-1318.

Lachapelle, G., (1990), Kinematic system in geodesy, surveying, and remote sensing, *Springer-Verlag,* New York.

Leick, A., (2004), GPS Satellite Surveying. *Third Edition, John Wiley & Sons*.

Lee, Y, C., (1992), Receiver Autonomous Integrity Monitoring (RAIM) Capability for Sole-Means GPS Navigation in the Oceanic Phase of Flight. *IEEE Aerospace andElectronic Systems Magazine*, Vol 7 (5), pp. 29 -36.

Lee, J., Won, D. H., Sung, S., Kang, T. S., and Lee, Y. J., (2007), High Assurance GPS Integrity Monitoring System Using Particle Filtering Approach. *In proceedings of 10th IEEE High Assurance Systems Engineering Symposium*, PP 437-438.

Leeuwen, S. S. V., (2002), GPS point position calculation, GPS solutions, vol 6, PP 115-117.

Levinson, D., (2005), Micro-foundations of congestion and pricing: A game theory perspective, Transportation Research Part A, Vol 39, pp. 691–704

Li, J., and Fu, M., (2003), Research on route planning and map-matching in vehicle GPS/deadreckoning/electronic map integrated navigation system. *IEEE Proceedings on Intelligent Transportation Systems*, 2, pp. 1639-1643.

Li, X., Lin, H., and Zhao, Y., (2005), A connectivity based Mapmatching algorithm. *Asian Journal of Geoinformatics*, 5, pp. 69-76.

Lu, G. and Lachapelle, G., (1991), Reliability Analysis for Kinematic GPS Position and Velocity Determination. *Manuscripta Geodaetica*, Vol. 18 (13), pp. 124-130.

Maddala, G. S. and Lahiri, K., (2009), Introduction to econometrics, *Wiley publishers*, ISBN: 978-0470015124.

Marchal, F., Hackney, J., and Axhausen, W., (2005), Efficient map matching of large global positioning system data sets tests on speed-monitoring experiment in Zurich. *Transportation Research Record,* 1935, pp. 93-100.

Mashrur, A.C., and Sadek, A., (2003), Fundamentals of Intelligent Transportation System planning. *First Edition, Artech House, London*.

MathWorks, (2008), *Genetic Algorithm and Direct Search toolbox user's guide*. The Mathworks inc.,

Matreata, M., (2005), Overview of the artificial neural networks and fuzzy logic applications in operational hydrological forecasting systems, Available at: http://www.balwois.com/balwois/administration/full_paper/ffp-641.pdf

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

Mintsis, G., Basbas, S., Papaioannou, P., Taxiltaris, C., and Tziavos, I. N., (2004), Applications of GPS technologies in land transportation system. *European Journal of Operational Research*, 152, pp. 399 – 409.

Montenbruck, O., and Gill, E., (2000), Satellite, orbits, models methods applications. *Springler, Berlin, Heidelberg*, New York.

Najjar, M.E., and Bonnifait, P., (2005), A roadmap matching method for precise vehicle Localization using belief theory and Kalman filtering. *Autonomous Robots*, Vol. 19, pp 173-191.

Nassreddine, G., Abdallah, F., and Denreux, T., (2008), Map matching algorithm using belief function Theory. *In proceedings of the 11th international conference on information fusion*.

Nijkamp, P., Pepping, G., and Banister, D., (1996), Telematics and transport behaviour. *Springer-Verlag Berlin and Heidelberg GmbH*.

Ochieng, W.Y., and Sauer, K., (2002), Urban road transport navigation: Performance of GPS after selective availability. *Transportation Research C*, 10 (3), pp. 171-187.

Ochieng, W.Y., Sauer, K., Walsh, D., Brodin, G., Griffin, S., and Denney, M., (2003), GPS integrity and potential impact on aviation safety. *The Journal of Navigation*, 56, pp. 51-65.

Ochieng, W.Y., Quddus, M.A., and Noland, R.B., (2004a), Integrated positioning algorithms for transport telematics applications.' *In proceedings of the Institute of Navigation (ION) annual conference*, 20-24 September, California, USA.

Ochieng, W.Y., Quddus, M.A., and Noland, R.B., (2004b), Map matching in complex urban road networks. *Brazilian Journal of Cartography (Revista Brasileira de Cartografia)*, 55(2), pp.1-18.

Ochieng, W. Y., Feng, S., Moore, T., Hill, C., and Hide, C., (2007), User Level Integrity Monitoring and Quality Control for Seamless Positioning in All Conditions and Environments. *In Proceedings of ION GNSS 20th International Technical Meeting of the Satellite Division*, September 25-28, Fort Worth, USA, pp.2573-2583.

Parkinson, B.W., Spilker, J. R., and Axelrad, P., (1996), Global positioning system: theory and Applications. *Volume 1-2, AIAA, Washington*, D.C.

Pecchioni, C., Ciollaro, M., and Calamia, M., (2007), Combined Galileo and EGNOS Integrity Signal: a multisystem integrity algorithm. *In proceedings of 2nd Workshop on GNSS Signals & Signal Processing - GNSS SIGNALS*.

Pink, O., Hummel, B., (2008), A statistical approach to map matching using road network geometry, topology and vehicular motion constraints. *In proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems*, Beijing, China.

Philipp, A. B., and Zunker, H., (2005), Integrity Hits the Road. *GPS World* pp. 30-36. July 2005.

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.

Price, D.C., and Hancock, G.M., (2005), Modernizing Accident Investigation using GPS. *ACSM CLSA NALS WFPS Conference and Technology Exhibition*, March 18-23, Las Vegas, Nevada, USA.

Pyo, J., Shin, D., and Sung, T., (2001), Development of a map matching method using the multiple hypothesis technique. *IEEE Proceedings on Intelligent Transportation Systems*, pp. 23-27.

Quddus, M.A., Ochieng, W.Y., Lin, Z., and Noland, R.B., (2003), A general map matching algorithm for transport telematics applications. *GPS Solutions*, 73, pp. 157-167,

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

Quddus, M.A., Noland, R.B. and Ochieng, W.Y., (2006b), The effects of navigation sensors and digital map quality on the performance of map matching

algorithms. *Presented at the Transportation Research Board (TRB) Annual Meeting*, Washington, D.C., USA.

Quddus, M.A., Ochieng, W., Y., and Noland, R, B., (2006c). Integrity of map-matching algorithms, *Transportation Research C: Emerging Technologies*, 144, pp.283 – 302.

Quddus, M.A., Ochieng, W.Y., and Noland R.B., (2007a), Current Map Matching Algorithm for Transport Applications: A state-of-the art and Future Research Direction. *Transportation Research Part-C,* 15, pp. 312 -328.

Quddus, M.A., North, R.J., Ochieng, W.Y., and Noland R. B., (2007b), Technologies to measure indicators for variable road user charging. *Universities Transport Study Group (UTSG) conference*, UK.

Russel, S., and Norvig, P., (2002), Artificial Intelligence: A Modern Approach, Prentice Hall, New Jersey.

Smaili, C., El Najjar, M. E., and Charpillet, F. O., (2008) HARPILLET A Road *Matching Method for Precise Vehicle Localization Using Hybrid Bayesian Network. Journal of Intelligent Transportation Systems*, 12(4), pp.176–188.

Sandhoo, K., Turner, D., and Shaw, M., (2000), Modernization of the Global Positioning System, *In Proceedings of the 13th International Technical Meeting of the Satellite Division of the Institution of Navigation, 2175-2183, Institute of Navigation*, Salt Lake City, Utah.

Sang, J., and Kubic, K., (1998), GNSS navigation system accuracy requirements for vehicle driving automation. *IEEE Position Location and Navigation Symposium*, pp. 426 -431.

Santa, J., Ubeda, B., Toledo, R., and Skarmeta, A., (2006), Monitoring the position integrity in road transport localization based services. *Proceedings of the 64th IEEE Vehicular Technology Conference-VTC2006*, Montreal, QC, Canada, 1–5.

Santa, J., Úbeda, B., and Skarmeta, A. F. G., (2007), Monitoring the Position Integrity in Road Transport: Using SBAS/EGNOS and Communication Issues. Proceedings of Location international conference, Bangalore, India.
Available at: http://www.location.net.in/magazine/2007/jan-feb/38_1.htm

Seeber, G., (2003), Satellite geodesy. *De Gruyter*, New York.

Sheridan, K., (2001), Service requirements document (SRD) for vehicle performance and emissions monitoring system (VPEMS). *Project Development Document (PDD) by Centre for Transport Studies*, Department of Civil and Environmental Engineering, Imperial College London.

Smaili, C., Najjar, M. E., and Charpillet, F., (2008), A Road Matching Method for Precise Vehicle Localization Using Hybrid Bayesian Network. *Journal of Intelligent Transportation Systems*, Vol 12 (4), pp.176 -188.

Sohn, H. J., (2009), Vector pattern map-matching algorithm for efficient multi-hypothesis tracking. IEICE Electronics Express. Vol. 6 (13), pp. 910-915.

Srinivasan, D., Chue, R. L., and Tan, C. W., (2003), Development of an improved ERP System Using GPS and AI Technique. *IEEE Proceedings on Intelligent Transportation Systems*, pp. 554-559.

Su, H., Chen, J., and Xu, J., (2008), A Adaptive Map Matching Algorithm Based on Fuzzy- Neural-Network for Vehicle Navigation System. *In proceedings of the*

7th World Congress on Intelligent Control and Automation. June 25 - 27, 2008, Chongqing, China

Sun, H, and Cannon, M. E., (1998), Reliability analysis of an ITS navigation system. *IEEE*, 1998, pp.1040-1046.
Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00660617 (Accessed June, 2009).

Sussman, J.M., (2005), Prospectives on Intelligent Transportation Systems. *Springer-Verlag Berlin and Heidelberg GmbH*

Syed, S., and Cannon, M.E., (2004), Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons. *In proceedings of the Institute of Navigation (ION) national technical meeting*, 26-28 January, California, USA.

Syed, S., (2005), Development of Map Aided GPS Algorithms for Vehicle Navigation in Urban Canyons, *M.Sc dissertation*, Department of Geomatics Engineering, University of Calgary.

Taghipour, S., Meybodi, M. R., and Taghipour, A., (2008), An Algorithm for Map Matching For Car Navigation System. *In Proceedings of 3rd IEEE International conference on information and communication technologies: From theory to application*. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4529959&isnumber=4529902

Tanaka, J., Hirano, K., Itoh, T., Nobuta, H., Tsunoda, S., (1990), Navigation system with map-matching method. *Proceeding of the SAE International Congress and Exposition*, pp 40-50.

Taylor, G., Blewitt, G., Steup, D., Corbett, S., and Car, A., (2001), Road Reduction Filtering for GPS-GIS Navigation. *Transactions in GIS*, 5(3), pp. 193-207.

Taylor, G and Blewitt, G, (2006), Intelligent Positioning: GIS-GPS Unification. *Wiley Blackwell publisher*, ISBN: 978-0470850039.

Taylor, G., Brunsdon, C., Li, J., Olden, A., Steup, D., and Winter, M., (2006), GPS accuracy estimation using map matching techniques: Applied to vehicle positioning and odometer calibration. *Computers, Environment and Urban Systems*, 30, pp. 757-772.

Teodorovi, D., (1999), Fuzzy logic systems for transportation engineering: the state of the art, Transportation Research Part A, Vol 33, pp 337-364.

Torriti, M. T., and Guesalaga, A., (2008), Scan-to-Map Matching Using the Hausdorff Distance for Robust Mobile Robot Localization. *In proceedings of 2008 IEEE international Conference on Robotics and Automation*, Pasadena, CA, USA

Tsui, J.B., (2000), Fundamentals of global positioning system receivers: a software approach, *John Wiley and Sons*.

Velaga, N. R., Quddus, M. A., and Bristow, A. L., (2009), Developing an Enhanced Weight Based Topological Map-Matching Algorithm for Intelligent Transport Systems. *Transportation Research Part-C,* 17, pp. 672 -683.

Walter, T. and Enge, P., (1995), Weighted RAIM for Precision Approach. *Proceedings of the Institute of Navigation (ION) GPS*.
Available at: http://waas.stanford.edu/~wwu/papers/gps/PDF/wraim_tfw95.pdf

White, C.E., Bernstein, D., and Kornhauser, A.L., (2000). Some map-matching algorithms for personal navigation assistants. *Transportation Research Part C,* 8, pp. 91–108.

Wong, S. V., and Hamouda, A. M. S., (2000), Optimization of fuzzy rules design using genetic algorithm, Ad*vances in Engineering Software*, Vol 31, pp. 251-262.

Wu, D., Zhu, T., Lv, W., and Gao, X., (2007), A Heuristic Map-Matching Algorithm by Using Vector-Based Recognition. *In Proceedings of the International Multi- on Computing in the Global Information Technology (ICCGI'07).*

Wullems, C., Pozzobon, O., and Kubik, K., (2004), Trust Your Receiver?. GPS World, October, 2004.

Xu, H., Liu, H., Norville, H. S., and Bao, Y., (2007), A Virtual Differential Map-Matching Algorithm. *In Proceedings of the IEEE Intelligent Transportation Systems Conference Seattle*, WA, USA.

Yang, D., Cai, B., and Yuan, Y., (2003), An Improved Map-Matching Algorithm Used in Vehicle Navigation System. *Proc., IEEE Conference on Intelligent Transportation Systems*, pp. 1246-1250.

Yin, H., Wong, S. C., Xu, J., and Wong, C. K., (2002), Urban traffic flow prediction using fuzzy-neural approach, Transportation Research Part C, Vol 10, pp 85-98.

Yin, H., and Wolfson, O., (2004), A weight-based map-matching method in moving objects databases. *Scientific and Statistical Database Management - Proceedings of the International Working Conference* , 16, pp. 437-438.

Yu, M., Li, Z., Chen, Y., and Chen, W., (2006), Improving Integrity and Reliability of Map Matching Techniques. *Journal of Global Positioning Systems,* 5, pp. 40-46.

Yun, Y., Yun, H., Kim, D., and Kee., C., (2008), A Gaussian Sum Filter Approach for DGNSS Integrity Monitoring. *The journal of Navigation*, 61, pp. 687–703

Zhang. Y., and Gao, Y., (2008a), A fuzzy logic Map matching Algorithm. *In Proceedings of 5th International conference on Fuzzy Systems and Knowledge discovery*, pp 132-136.

Zhang, Y., and Gao, Y., (2008b), An Improved Map Matching Algorithm for Intelligent Transportation System. *In proceedings of IEEE International conference on industrial technology*, pp 1-5.

Zhao, Y., (1997), Vehicle Location and Navigation Systems. *Artech House*, Inc., Norwood.

Zhao, L., Ochieng, W.Y., Quddus, M.A, and 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. *The Journal of Navigation*, 56, pp. 257-275.

Zhong, M., Lingras, P., Sharma, S., (2004), Estimation of missing traffic counts using factor, genetic, neural, and regression techniques, Transportation Research Part C, Vol 12, pp139-166.

Frequently used Acronyms

A&E:	Accident & Emergency
AL:	Alarm Limit
aMM:	advanced Map Matching
ANFIS:	Adaptive Neuro Fuzzy Inference System
ANN:	Artificial Neural Network
CI:	Commercial Issues
CR:	Continuity Risk
DoD:	Department of Defence
DGPS:	Differential Global Positioning System
DR:	Deduced Reckoning
ECEF:	Earth Centred Earth Fixed
EGNOS:	European Geostationary Navigation Overlay Service
EKF:	Extended Kalman Filter
ERP:	Electronic Road Pricing system
FAR:	False Alarm Rate
FIS:	Fuzzy Inference System
FL:	Fuzzy Logic
FM:	Filtering Method
GA:	Genetic Algorithm
GBAS:	Ground Based Augmentation System
gMM:	geometric Map Matching
GIS:	Geographic Information Systems
GLONASS:	GLObal NAvigation Satellite System
GPS:	Global Positioning Systems
GNSS:	Global Navigation Satellite Systems
GPRS:	General Package for Radio Service
GSM:	General System for Mobile

HDOP:	Horizontal Dilution of Precision
HTL:	Horizontal Tolerance Limit
ICAO:	International Civil Aviation Organization
INS:	Inertial Navigation Systems
IR:	Integrity risk
ITS:	Intelligent Transportation Systems
LAAS:	Local Area Augmentation System
MDR:	Missed Detection Rate
MEMS:	Micro-Electro Mechanical Systems
MF:	Membership Function
MM:	Map Matching
NN:	Neural Networks
OCDR:	Overall Correct Detection Rate
OE:	Operational Environments
pMM:	probabilistic Map Matching
RCM:	Range Comparison Method
RAIM:	Receiver Autonomous Integrity Monitoring
RMS:	Root Mean Square
RNP:	Required Navigation Performance
SBAS:	Satellite Based Augmentation System
SI:	Safety Issues
SM:	Snapshot Methods
SOL:	Safety of Life
SPR:	System Performance Requirements
ToO:	Type of Operation
tMM:	topological Map Matching
TTA:	Time to alarm
TWS:	Total Weighting Score
UERE:	User Equivalent Range Error
ULIM:	User Level Integrity Monitoring
VRUC:	Variable Road User Charging

WLAN:	Wireless Local Area Network
WSSE:	Weighted Sum of Squares Errors